機械学習基礎

赤穂 昭太郎

2025 年 4 月 4 日

本講義シリーズの主旨

本講義シリーズでは、将来統計教育に携わることを志す方々に向けて、機械学習の広範な領域を紹介することを目的とする。

今日、データサイエンスの重要性が高まる中で、いわゆる「狭義の統計学」だけでなく、機械学習をはじめとする多様な分野に触れる必要性が増している。こうした分野は、従来の統計学の枠組みとは異なる発想やアプローチを取ることも多く、混乱を招く場面もあるかもしれない。たとえば、同じ用語が統計学とは異なる意味で使われていたり、理論的な整合性よりも実務的な有用性が重視されたりすることも少なくない。

その意味で、機械学習は統計学と比べてまだ学術的に成熟しきった体系ではなく、実務主導で発展してきた背景を持つ。だからこそ、統計学の確かな基盤を持つ人にとっては、ノイズの多い分野と映ることもあるだろう。

しかし、こうした現状をネガティブに捉えるのではなく、機械学習が統計学とは異なる視点から問題にアプローチしてきたことを前向きに受け止めてほしい。統計学が築いてきた理論的な堅牢さと、機械学習がもたらした柔軟かつスケーラブルな技術とを架橋することで、より深みのあるデータサイエンスの教育と実践が可能になる。

本講義では、受講生の知識レベルに幅があることを考慮し、初歩的な内容から発展的な話題まで幅広く取り上げる。重要なのは、すべての内容を即座に理解・習得することではなく、現代のデータサイエンスにおいて求められる技術や発想がどこにあるのかを把握し、それらを伝統的な統計学の文脈の中でどのように位置づけ、活用していくかの素地をつくることである。

統計学という強固な学術的基盤を大切にしつつ、機械学習という隣接分野と誠実に向き合い、教育者としての視野を広げる第一歩になれば幸いである。

第 1 章

導入：MNIST データの分類

1.1　MNIST データ

まず事例を通じて機械学習の手順・概念の基礎を理解する.

図 1.1　MNIST data の一例

ここでは MNIST data という手書き数字データを使用する．深層学習研究などで有名な Yann LeCun らが作成した自由に使うことができるデータセット (`http://yann.lecun.com/exdb/mnist/`)．28×28 pixel の 70,000 枚の画像からなる (もともとは 60,000 枚学習用，10,000 枚テスト用として用意されている)．OpenML (`https://www.openml.org/`) からダウンロード可能.

Program 1.1　MNIST data 読み込み

```
from sklearn.datasets import fetch_openml
# Step 1: Load MNIST dataset
mnist = fetch_openml('mnist_784', version=1)
X, y = mnist.data, mnist.target.astype(int)
```

1.2　ロジスティック回帰の適用

とりあえずロジスティック回帰による識別を行い，精度を算出するところまでやってみる．20% のテストデータに対する精度（Accuracy）は 0.9216 であった．

Program 1.2　MNIST 識別

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score

# Step 2: Split into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y
    , test_size=0.2, random_state=42)

# Step 3: Normalize features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Step 4: Train Logistic Regression model
model = LogisticRegression(penalty='l2', solver='lbfgs',
     multi_class='multinomial', max_iter=1000,
    random_state=42)
```

```
model.fit(X_train, y_train)

# Step 5: Evaluate the model
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)

print(f"Test accuracy: {accuracy:.4f}")
```

1.3 各要素の機械学習的位置づけ

この事例には機械学習全般に共通する要素が含まれている．まず，事例に即してそれぞれの要素について簡単に説明する．

枠組み ほぼすべての機械学習では，与えられた入力 X に対する出力 Y への関数 $Y = f(X)$ を学習する．統計における推定と学習はほぼ同義である (主成分分析のようなものもパラメータの推定という意味で学習に含める)．ここでやったように，X と Y のペアを学習データ（訓練データ）として f を学習する枠組みを教師あり学習 (supervised learning) という．

入出力 この問題では画像データを実数値ベクトルとして入力し，対応するクラスラベルを出力とした．統計では入力を説明変数，出力を目的変数と呼ぶのが普通．また，実数値のようなものを連続変数，クラスラベルを離散変数と呼ぶが，それぞれ量的変数，質的変数と同じ．

モデル 多項ロジスティック回帰モデルは，

$$P(Y = k) = \exp(-f_k(x))/\sum_{k'} \exp(-f_{k'}(x)),$$

$$f_k(X) = \beta_{k,0} + \sum_i \beta_{k,i} X_i)), \quad k = 1, \ldots, K-1$$

という確率モデルをあてはめる（一般化線形モデルの一種）．Python の scikit-learn パッケージに LogisticRegression 関数として含まれている．

目的関数 教師あり学習では，モデル出力 $f(X)$ と Y の誤差を表す目的関数を定義

して，それを最小化する．ロジスティック回帰では通常，上記確率モデルの負の対数尤度を目的関数として最小化する．

正則化　パラメータ数に比べてデータ数が十分でない場合は，過学習という現象が起きて，学習データに対する目的関数は小さくなっても，テストデータに対する目的関数は小さくならないことがある．そのための汎用的な方法が正則化で，以下のようにもともとの目的関数に正則化項を足して最小化する．

$$\text{目的関数} + \lambda \times \text{正則化項} \to \min_f$$

ここでやった実例では定数項を除くパラメータの 2 乗和を正則化項とし（L_2 正則化），正則化パラメータはデフォルト値 ($C = 1/\lambda = 1.0$) となっている．正則化パラメータは，ハイパーパラメータの一種で，モデル選択の対象となる．

学習アルゴリズム　機械学習では複雑なモデルをあてはめることが多いので，パラメータの最適化にどのようなアルゴリズムを使うかも重要な要素となる．ここで用いられているのは L-BFGS (limited-memory Broyden-Fletcher-Goldfarb-Shanno) アルゴリズムで，準ニュートン法の一種である．

その他　上記が機械学習を構成する要素であるが，それ以外についても少し補足する．事例ではまず変数の値の規格化を行っている．規格化は正則化によって解に影響を与えるとともに，学習アルゴリズムの収束性にも関係する．また，データを学習用データとテスト用データに分けるというのもテスト誤差を評価するための自然な評価法.

課題 1　データ分析を行う際に，とりあえず既存の機械学習の枠組みに無理やり帰着させて問題を解くことが多い．自分が興味があるデータ分析の問題を上のような形で整理してみよう（特に問題設定，入出力，目的関数あたりが重要）．ただ，それだと必ずしも適切な問題になっているとは限らず，新たな問題設定を考えることで機械学習の世界が広がっていく．もし可能であれば，最初に整理したものの問題点を議論して，通常の機械学習では考えられていない（と自分が考える）定式化を考えてみよう．

以下では，上で簡単に述べたそれぞれの要素について，より一般的な観点から説明する．

1.4　枠組み

入力 X から出力 Y への関数 f を学習するのに，入力 X だけをデータとして与える枠組みを教師なし学習 (unsupervised learning) という．教師なし学習では主に X から重要な情報を抽出する関数を学習することが目的となる．何が「重要か」によって，目的関数を適切に設定する必要がある．

一方，出力 Y が連続変数か離散変数かでアプローチがおおまかに分かれる．「教師あり・なし」と「連続・離散」の 4 通りに機械学習の枠組みは大きく分けることができる．一般に，自然な線形モデルがあてはめやすい連続出力の場合に比べて，離散化処理が入る離散出力の場合の方が非線形性を含むため，最適化や理論解析は難しくなる傾向にある．表 1.1 にまとめる[*1]．

	連続出力	離散出力
教師あり	回帰 (regression)	識別 (classification)
教師なし	次元縮約 (dimension reduction)	クラスタリング (clustering)

表 1.1　機械学習のおおまかな分類

ただ，現実にはこれらの中間的なものを含め，さまざまな問題設定がある．そのいくつかについて概要を述べる．

半教師あり学習　入力データセット X の一部のデータにだけ出力 Y が与えられているような状況での学習を半教師あり学習 (semi-supervised learning) という．クラスラベルを人手で与える必要があるような場合は，ラベリング（アノテーションという言葉を使うこともある）のコストが高いため，大規模データを扱う上で実用上重要な役割がある．音声認識はラベリングコストの高い代表的な応用例．

転移学習　過去に収集したデータなどを現在の課題に活用してデータ不足を補うため

[*1] 識別の呼び方には議論がある．ほかの呼び方の例：判別：これは判別分析で先に使われているので若干微妙だが意味は近い．分類：統計学ではクラスタリングの和名として分類を使うことがある．そこで，紛らわしい場合はクラス分類という言葉を使う．

の枠組み．さまざまなシナリオが考えられるが，過去に収集したものと現在必要なデータとは品質や属性が異なっていたり欠けていたりするため，それらを埋めるためのモデル化が必要となる．

意思決定問題　出力 Y が行動などの意思決定であるような枠組みで，機械学習でよく研究されているものに，バンディット問題・強化学習・能動学習・ベイズ最適化などがある．これらの特徴は，出力 Y そのものが与えらるのではなく，入力 X のよさを表す「報酬」のような間接的な情報が与えられて，それを大きくするような行動を出力する必要があることである．

（多腕）バンディット問題　((multi-armed) bandit problem) 複数の確率未知のスロットマシンで，できるだけコインの総和が大きくなるように現時点での最適なスロットマシンを選ぶという問題．もともと適応的 (逐次) 割り当て (adaptive(sequential) allocation) として 1950 年代から研究されてきており，治験における患者の治癒数最大化などを想定．一方，インターネット広告の最適な表示との関連で IT 系の大手企業を中心に盛んに研究されるようになり，機械学習における主要課題となった．

強化学習　(reinforcement learning) 状態がマルコフ連鎖に従って時間発展していき，各状態において報酬 R が与えられる．このとき，今後得られる報酬の総和が最大になるように，各状態での行動を出力する．確率モデルとしてマルコフ決定過程 (MDP: Markov Decision Process) というモデルとなる．ロボットの制御や将棋 AI などの学習をはじめ，その汎用性から，時系列をともなう学習課題に対して広く用いられている．

能動学習　(active learning) 入力データを得るのにコストがかかる場合に，入力を指定して学習データを獲得する枠組み．逐次実験計画の一種で，目的が，できるだけ少ない学習データで，現在仮定している学習モデルの精度が高くなることが目的となる．

ベイズ最適化　(Bayesian optimization) これも逐次実験計画の一種で，目的は，ある関数の値の最大となる入力をできるだけ少ない回数で探索することが目的で，こちらがむしろ通常の実験計画の枠組みに近い．

なお, バンディット問題を含め, 意思決定問題では, “Exploration-Exploitation

trade-off” (探索と活用のトレードオフ) というものが存在する．バンディットの例でいえば，当初確率が未知なのでいろいろなスロットを選ぶ必要がある（探索）のだが，当初の目的であるコインの総和を最大化するためには確率の高いスロットを選ぶ必要がある（活用）ため，それらのバランスをうまくとる必要がある．

1.5 入出力

入出力として最もよく用いられるのは連続値・離散値（のベクトル）である．MNIST 識別の例のような画像データ，音声データなどは高次元データであるというのも大きな特徴で，「次元の呪い」と呼ばれる，次元が高いことによる問題点がさまざまに起きる．

機械学習では，連続値・離散値以外にも入出力にはさまざまな形のものを扱う．

画像データは高次元のベクトル値データとして扱うこともできるが，もともとは 2 次元データとみなす方がより自然であるし，色情報も入れて考えればこれは 3 次元的な配列データとみなすこともでき，これは機械学習ではテンソルデータと呼ばれることもある．

このほかの重要な例として，ChatGPT や機械翻訳で扱う自然言語がある．これは文字や単語を離散的なアルファベットとした系列データである．これは時系列の仲間のようなものと思うこともできる．類似したものに，バイオインフォマティクスで用いられる遺伝子やたんぱく質の配列がある．

そのほかのデータとしては，グラフ構造やネットワーク構造のようなものが挙げられる．バイオインフォマティクスやマテリアルインフォマティクスで登場する分子構造や反応ネットワーク，SNS の人間関係のつながりなどが該当する．

これらのデータを処理するための最も基本的で多用されているアプローチは，対象に対する知識を用いて「特徴量」を設計して数値ベクトルに変換することでベクトル値データのための多変量解析手法を使えるようにするというものである．この特徴量エンジニアリングと呼ばれるアプローチは長年にわたって王道の手法として用いられてきた．

対象の知識が不明な場合などに数値でない情報を数値化する方法として，カーネル法があり，バイオインフォマティクスなどでサポートベクターマシンなどのカーネル法が流行したのにはこのような背景がある．

一方，機械学習で大量のデータが得られるようになると，特徴量の抽出もデータから学習させてしまおうという考え方が生まれてきた．多変量解析でも数量化，主成分分析やクラスタリングはその方向性の手法であると考えることもできるが，深層学習はさらにそれを進めたものと考えることができる．深層学習では，モデルアーキテクチャを工夫して学習によって有効な特徴量を抽出することを目指すため，アーキテクチャエンジニアリングと呼ばれることもある．

1.6　モデル

機械学習では，データに基づいて予測性能を上げたいという要請が通常の統計学よりも強いので，複雑な非線形モデルが数多く提案されている．また，前項の入出力のところでも述べたように，特徴量エンジニアリングのためのモデル設計という側面もある．

モデルについては多岐にわたるのでそのすべてを網羅することは不可能だが，大まかな分類については非線形性の強さ・パラメトリックかノンパラメトリックかといったようなモデルの性質で分けることができる．

学習モデルの詳細については次回で詳しく説明するが，複雑なモデル化はモデルの解釈性とトレードオフの関係にある．機械学習では複雑なモデルを構成するため，必然的にモデルの解釈性は低かったが，AI が社会に浸透するにつれ，なぜその出力を出したかという説明を求められるようになり，XAI (説明可能 AI: explainable AI) という研究分野が生まれ，複雑な非線形モデルを解釈するための仕組みが数多く考え出されるようになってきた．

1.7　目的関数

予測を目的とした機械学習の場合は，予測誤差を最小にするように目的関数を設定する．

教師なし学習では，有用な情報をできるだけ多くするという観点で情報量のような目的関数を設定する．

統計モデルの場合は最尤推定が基本となる．二乗誤差は正規ノイズを仮定した場合の最尤推定であり，両側指数分布（ラプラス分布）であれば絶対誤差の和を最小にすることになる．

一方，リスクを最小にするという観点で，期待リスクを最小化するという観点でパラメータの最適化を行うという考え方があり，統計的決定理論（ベイズ決定理論）として知られている．

後で述べるように，機械学習では勾配法に基づいて繰り返し計算により最適化を行う際には目的関数が凸であると最適化が容易になるため，2 乗誤差や絶対誤差はその観点でも都合がよい．

1.8 正則化

学習データ以外のテストデータに対する性能を汎化能力 (generalization ability) と呼ぶ．機械学習の多くの場合，学習データとテストデータは未知の独立同分布 (i.i.d.) に従って生成されると仮定し，汎化能力を評価する．統計学ではこの未知の分布は母集団分布という言葉で説明される．

過学習を避けるための正則化技法は，深層学習を含め機械学習の多くのモデルで採用されている．

L_p 正則化では，パラメータ $\boldsymbol{a}$ の p 次ノルム

$$\|\boldsymbol{a}\|_p = \left(\sum_{i=1}^{d} a_i^p \right)^{1/p} \tag{1.1}$$

が最もよく用いられているもので，$0 < p < 1$ のときは $\boldsymbol{a}$ の成分が厳密に 0 になるものが現れ得るという意味で変数選択に用いることができる (スパースモデリングと呼ぶこともある)．AIC や BIC も正則化の一種とみなすことができる．これらについて詳しくはモデル選択のところで紹介する．

それ以外にも数多くの正則化がありえる．L_p 正則化以外によく用いられるものとして，隣接するパラメータの値を近くするような (例： $|a_i - a_{i+1}|$) は時系列や画像

のように変数間に連続性が仮定できるような場合に用いられる．

最大エントロピー原理（与えられた条件の下で最もエントロピーの高い解を選ぶべき）も一種の正則化と考えることができる．

なお，正則化には和の形で制約をかけるもの (Tikhonov 型) のほか，ノルム一定という条件のもとで損失関数を最適化するもの (Ivanov 型) があり，一般にこれらはパラメータを調整すれば等価な関係にある．Ivanov 型正則化：

$$\text{目的関数} \to \min_f \qquad \text{s.t.} \quad \text{正則化項} \leq C$$

深層学習における Early stopping や dropout も implicit regularization とみなすことができる．

機械学習の公平性を保つ研究が盛んにおこなわれるようになったが，できるだけ公平性を保ちつつ誤差を最小にするような枠組みも提案されている (Kamishima et al.)．

1.9　学習アルゴリズム

一旦目的関数（&正則化）の最小化問題として定式化されてしまえば，基本的には最適化分野の力を借りてそのアルゴリズムを利用することになる．

最も汎用的に用いられているものは勾配法あるいは最急降下法と呼ばれる手法で，パラメータ θ の関数 $L(\theta)$ を最小化するのに

$$\Delta\theta = -\epsilon \frac{\partial L(\theta)}{\partial \theta} \tag{1.2}$$

によって少しずつ解の改良を行う方法である．

統計モデルに特化した学習アルゴリズムとして，潜在変数モデルに対する EM (Expectation-Maximization) アルゴリズムがよく知られている（混合分布の項で解説予定）．

そのほかの汎用的なアルゴリズムとして動的計画法が知られている．潜在変数モデルの時系列モデルである隠れマルコフモデル・状態空間モデルの学習や，強化学習の最適化などにも用いられる．

ベイズモデリングでは，本章で述べたレシピとは異なる面もあるが，ベイズ推論の

ためのアルゴリズムとして MCMC, 変分ベイズ法（平均場近似），粒子フィルタといったアルゴリズムが研究されている．

正則化の形によっては劣モジュラー最適化など最適化分野の最新のアルゴリズムを用いることもある．

参考文献

[1] G. James, D. Witten, T. Hastie, R. Tibshirani “An Introduction to Statistical Learning, Second Edition”, Springer 2023

[2] C.M. Bishop, “Pattern recognition and machine learning”, Springer 2006 (ビショップ：パターン認識と機械学習（上下），丸善)

[3] 萩原，入門 統計的回帰とモデル選択，共立出版 2022

[4] 赤穂，カーネル多変量解析，岩波書店 2008

[5] 青嶋，矢田，高次元の統計学，共立出版 2019

[6] S. Watanabe, M. Opper, Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. Journal of machine learning research, 11(12) 2010