確率・統計（真貝）第 11 回　参考資料

11_PS_MachineLearning.pdf

機械学習・深層学習の概観

人間の知能と機械が同レベルに達するか超えるものを「人工知能」と呼ぶが，ビッグデータが取り扱える時代になり機械学習の手法が身近になって，その実現に近づいている．

1　機械学習の要素

大量のデータから必要な情報を抽出する「機械学習」は，分類・回帰・クラスタリングなどに応用される．類似度を明示的に扱うか（いわゆる機械学習），暗黙的に評価するか（ニューラルネットワーク neural network; NN，多層にした NN を深層学習 deep learning とも呼ぶ）によって大きく２種類に分かれる．

1.1　3 つの応用

- **分類（Classification）**
  データの類似性に基づいて，データをカテゴリー（あるいはクラス）に分ける機能である．郵便番号に記載された文字の判読，画像中の猫の判定などが代表例である．
  データの中でどこに境界線（境界面）を引くか，という数学になる．
  $\Longrightarrow$ 数学的手法の例：教科書 §4.6 判別分析
  $\Longrightarrow$ 明示的方法の例：k 近傍法，サポートベクターマシン
- **クラスタリング（Clustering）**
  類似しているデータどうしを集めて集合（クラスター）をつくり，データを仕分けする機能である．類似性分類の前処理でもある．正解か不正解かが明確ならば，「教師あり学習」と言える．
  データ間の距離を定義する，という数学になる．
  $\Longrightarrow$ 数学的手法の例：教科書 §4.7 クラスター分析
  $\Longrightarrow$ 明示的方法の例：階層クラスタリング法，多次元尺度構成法，カーネル k 平均法
- **回帰（Regression）**
  データの類似性に基づいて，データの数値を予測する機能である．株価予測や製造業における工程管理などに応用される．
  データ $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$ から，全体の傾向を知るための関数 $y = f(x)$ を導く数学になる．
  $\Longrightarrow$ 数学的手法の例：教科書 §4.3 回帰分析
  $\Longrightarrow$ 明示的方法の例：カーネル線形回帰

1.2　類似度関数

データが類似していることを示す方法の 1 つは，データ間の**距離関数**を定義することである．距離が小さいほど類似している，という考えである．データ $(x_1, \cdots, x_n)$ と $(y_1, \cdots, y_n)$ のユークリッド距離（あるいは**ノルム**）は，

$$||\boldsymbol{x} - \boldsymbol{y}|| = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2} \qquad (1)$$

である．一般に，p ノルムを

$$||\boldsymbol{x} - \boldsymbol{y}||_p = \sqrt[p]{(x_1 - y_1)^p + \cdots + (x_n - y_n)^p} \qquad (2)$$

とする．また，重みをつけるさまざまな距離が提案されている．

別の方法として, データの内積をとる（**カーネル関数** $k(x, y)$ を定義する）考えがある．値が大きいほど類似していることになる．もっとも簡単なものは

$$k(x, y) = \sum_i x_i y_i \qquad (3)$$

とするものだ．より，一般には，カーネル関数は，グラム行列

$$G(x_1, \cdots, x_n; k) = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{bmatrix} \qquad (4)$$

が正定値となればよい．（G が正定値とは，$G = V^T V$ (V は行列) と分解できること，あるいは G が対称行列かつ固有値が非負であること）．

1.3　目的関数 objective func.

データ $(x_1, \cdots, x_m)$ と $(y_1, \cdots, y_n)$ があり，両者の関係が線形であるとして，その関係式（目的関数）

$$y = ax + b \qquad (5)$$

を導きたい．データからは

$$\begin{aligned} y_1 &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1m}x_m + b_1 \\ y_2 &= a_{21}x_1 + a_{22}x_2 + \cdots + a_{2m}x_m + b_2 \\ \vdots &= \vdots \\ y_n &= a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nm}x_m + b_n \end{aligned} \qquad (6)$$

と書ける．これをベクトルと行列を用いて，

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \qquad (7)$$

と表す．以下のように定義した．

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix},$$

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}.$$

2　目的関数最適化の例：最小二乗法

2.1　目的関数

例えば，最小二乗法で，データ $\boldsymbol{x}$ と $\boldsymbol{y}$ の関係式 $y = ax+b$ を求めたいとき，誤差

$$e = \sum_i \{y_i - (ax_i + b)\}^2$$

を最小にするような (a, b) を求める問題になる．ここでは誤差（距離）最小を原理としている．

これを一般化して，ある関数 S を最小化（あるいは最大化でもよい）するような問題を考えよう．例えばノルム

$$S = ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}||_2 \tag{8}$$

を最小化する問題を考える．$\boldsymbol{A}$ を既知と考えて S を最小とする解 $\boldsymbol{x}_0$ を求めたい．式 (8) は

$$\begin{aligned} S &= (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x})^T(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}) \\ &= \boldsymbol{y}^T(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}) - \boldsymbol{x}^T\boldsymbol{A}^T(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}) \\ &= \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} \\ &= \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} \end{aligned}$$

S を $\boldsymbol{x}$ で微分して，（ベクトルの微分は成分ごとの微分をまとめたもの）

$$\frac{\partial S}{\partial \boldsymbol{x}} = -2\boldsymbol{A}^T\boldsymbol{y} + 2\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}$$

となるので，S を最小にする $\boldsymbol{x}$ は，この微分がゼロとなることから

$$\boldsymbol{x} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{y} \tag{9}$$

などと求められる．

2.2　最小二乗解が不偏推定量であること

式 (7) の $\boldsymbol{b}$ が誤差 $\boldsymbol{e}$ とする．$\boldsymbol{e}$ は，平均ゼロ，分散 σ_0 の正規分布にしたがうとする．最小二乗解を導いた式 (9) で，真の値 $\boldsymbol{x}$ を $\boldsymbol{x}_0$ とすると，

$$\begin{aligned} \hat{\boldsymbol{x}} &= (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{e}) \\ &= \boldsymbol{x}_0 + (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{e}. \end{aligned} \tag{10}$$

誤差 $\boldsymbol{e}$ は正規分布にしたがう確率変数とすれば，$\hat{\boldsymbol{x}}$ の各成分も正規分布にしたがう．期待値をとると，

$$E[\hat{\boldsymbol{x}}] = \boldsymbol{x}_0 + (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T E[\boldsymbol{e}] = \boldsymbol{x}_0 \tag{11}$$

最後の部分は，$E[\boldsymbol{e}] = 0$ を用いた．この式は，最小二乗解 $\hat{\boldsymbol{x}}$ の期待値が真の値に一致するという保証（不偏推定量であること）を与えてくれる．

2.3　最小二乗解の分散

最適化された解を得たとしても，その不定性を添えないと不毛な議論になる．最小二乗解 $\hat{\boldsymbol{x}}$ の分散を計算しよう．

$$\begin{aligned} & E[(\hat{\boldsymbol{x}} - \boldsymbol{x}_0)(\hat{\boldsymbol{x}} - \boldsymbol{x}_0)^T] \\ = \; & E[((\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T E[\boldsymbol{e}])((\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T E[\boldsymbol{e}])^T] \\ = \; & (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T E[\boldsymbol{e}\boldsymbol{e}^T]\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1} \\ \underbrace{=}_{*} \; & (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T(\sigma_0^2\boldsymbol{I})\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1} = \sigma_0^2(\boldsymbol{A}^T\boldsymbol{A})^{-1} \end{aligned}$$

ここで，*部分では，$V[\boldsymbol{e}] = \sigma_0^2$ を用いた．この式で，$(\boldsymbol{A}^T\boldsymbol{A})^{-1}$ の (i, i) 成分を a_{ii} とする．

σ_0^2 の不偏推定量は，データ数を N，モデルのパラメータ数を K とすれば，

$$\hat{\sigma}^2 = \frac{||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}||_2^2}{N - K} \tag{12}$$

となるので，パラメータ $\hat{x}_i$ の分散は，$\hat{\sigma}^2 a_{ii}$ となる．この平方根

$$\hat{s}(\hat{x}_i) = \hat{\sigma}\sqrt{a_{ii}} \tag{13}$$

は $\hat{x}_i$ の標準誤差と呼ばれる．

3　ニューラルネット (NN) の考え

NN は暗黙的に目的関数を探る方法である．正解・不正解の判定を促すために，神経の閾値に見立てた**活性化関数** (activation func.)

$$z = h(u), \quad \text{関数形は任意}, \quad \begin{cases} h(u) \to 1 & u \to +\infty \\ h(u) \to 0 & u \to -\infty \end{cases} \tag{14}$$

を定義する．そして式 (5) を得るために，式 (6) のように，n 本の連立（ユニット）を考え，

$$\boldsymbol{z}^{(1)} = h(\boldsymbol{A}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) \tag{15}$$

とする．これをさらに多段階の中間層を考え（ユニット数は各層で異なってよい），

$$\boldsymbol{z}^{(1)} = h(\boldsymbol{A}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) \tag{16}$$

$$\boldsymbol{z}^{(2)} = h(\boldsymbol{A}^{(2)}\boldsymbol{z}^{(1)} + \boldsymbol{b}^{(2)}) \tag{17}$$

$$\vdots \qquad \vdots$$

$$\boldsymbol{z}^{(N)} = h(\boldsymbol{A}^{(N)}\boldsymbol{z}^{(N-1)} + \boldsymbol{b}^{(N)}) \tag{18}$$

とする．これらをまとめて，入力 $\boldsymbol{x}$ から出力 $\boldsymbol{y} = \boldsymbol{z}^{(N)}$ への対応式とみなすのである．決めるべきパラメータが多いので，学習させる量も多く必要になるが，多層にすれば予測精度がよくなると期待される．

ユニット間のつなぎ方にもさまざまな提案がなされている．すべてのユニットを次のユニットにつなぐ**全結合** (fully-connedted) か，入力の一部だけを次の層のユニットにつなぐ**畳み込み NN**(convolutional NN; CNN) か．（CNN は画像判別問題でよく使われる）．多層を順にこなす**順伝播型** (feedforward) か，一度出力された層の結果をもう一度使う**再帰型** (recurrent NN; RNN) か．（RNN は時系列データでよく使われる）．

参考文献

[1] 植村誠『物理のためのデータサイエンス入門』（講談社，2023）

[2] 申吉浩 監修『機械学習アルゴリズム入門　類似性の科学』（工学社，2022）

[3] 真貝寿明『徹底攻略 確率統計』（共立出版，2012）