Generative Adversarial Networks

小泉 孝弥

1 Intoroduction

近年, 機械学習

2 Fundermetal concepts of Machine Learning

この節では機械学習の数学的な定式化を行う. $\mathcal{X}$ と $\mathcal{Y}$ を有限次元線型空間とし, $(\Omega, \mathcal{F}, \mathbb{P})$ を完備な確率空間とする.

Definition 2.1. (仮説空間, 仮説)
$\mathcal{X}$ から $\mathcal{Y}$ へのなんらかの条件を満たす写像の集まりのことを仮説空間といい $\mathcal{H}$ と表記する. すなわち,

$$\mathcal{H} = \{f : \mathcal{X} \to \mathcal{Y} \mid f \text{ が満たす条件 }\}$$

である. (条件の具体例は後に述べる). 仮設空間 $\mathcal{H}$ の元のことを仮説 (またはモデル) と呼ぶ. また, この時の $\mathcal{X}$ を入力空間, $\mathcal{Y}$ を出力空間と呼ぶ.

これ以降, 入力空間を $\mathbb{R}^d$ 出力空間を $\mathbb{R}^m$ とする.

Definition 2.2. (データ)
$\mathcal{X}$ と $\mathcal{Y}$ の直積集合 $\mathcal{X} \times \mathcal{Y}$ の有限部分集合のことを教師ありデータという. また, $\mathcal{X}$ の有限部分集合のことを教師なしデータと呼ぶ.

Definition 2.3. (予測損失)
$\mathcal{H}$ を仮説空間とし, (X, Y) をデータの分布 $\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$ に従う確率変数とする. 以下で定義される $\ell : \mathcal{H} \to \mathbb{R}$ を予測損失と呼ぶ.

$$\ell(f) = \mathbb{E}[\|f(X) - Y\|].$$

予測損失が最小となるような $f \in \mathcal{H}$ を求めることが機械学習の目標である. しかし, 一般にデータの分布は未知であるためこの式を解くことができない. そこで、損失関数というものを定義し、それを最適化することを考える.

Definition 2.4. (損失関数)
$\mathcal{H}$ を仮説空間とする. $\mathcal{H}$ から $\mathbb{R}$ への写像 $\mathcal{L}_D : \mathcal{H} \to \mathbb{R}$ を損失関数 (Loss function) と呼ぶ.

Definition 2.5. (機械学習空間)
D をデータ, $\mathcal{H}$ を仮説空間, $\mathcal{L}_D$ を損失関数とする. この時 5 つ組 $(\mathcal{X}, \mathcal{Y}, D, \mathcal{H}, \mathcal{L}_D)$ を機械学習空間 (Machine Learning space) という.

Definition 2.6. (学習, 最適仮説)
$\mathcal{H}$ を仮設空間, $\mathcal{L}_D : \mathcal{H} \to \mathbb{R}$ を損失関数とする. 損失関数が最大または, 最小となるような $f^* \in \mathcal{H}$ を求めること[*1]を学習といい, $f^* \in \mathcal{H}$ を最適仮説と呼ぶ.

Definition 2.7. (機械学習)
機械学習空間 $(\mathcal{X}, \mathcal{Y}, D, \mathcal{H}, \mathcal{L}_D)$ 上での学習を機械学習という. 特に, D が教師ありデータの時, 教師あり機械学習と呼び, 教師なしデータの時, 教師なし機械学習と呼ぶ.

Remark 2.8. (訓練データとテストデータ)
機械学習で学習を行う際は, データ D を全て使って学習させるのではなく, データを訓練データ D_{Train} とテストデータ D_{Test} に分ける. 訓練データはモデルの学習に用いて, テストデータはモデルの制度の確認のために用いる.

3 教師あり機械学習の具体例

前節では, 機械学習の抽象的な枠組みを紹介したが, この説では機械学習空間の具体例を述べる. $D = \{(x_i, y_i)\}_{i=1}^N \subset \mathcal{X} \times \mathcal{Y}$ を教師ありデータとする.

3.1 単回帰分析・重回帰分析

最初に機械学習の最も基本的なモデルである単回帰分析を紹介する. その後その多変数拡張である重回帰分析を紹介する.

Example 3.1. (単回帰分析)
機械学習空間 $(\mathcal{X}, \mathcal{Y}, D, \mathcal{H}, \mathcal{L}_D)$ を以下のように定義する.
$\mathcal{X} = \mathbb{R}$, $\mathcal{Y} = \mathbb{R}$,

$$\mathcal{H} = \{f : \mathcal{X} \to \mathcal{Y} \mid f(x) = wx, w \in \mathbb{R}\},$$
$$\mathcal{L}_D(f) = \sum_{i=1}^{N} |f(x_i) - y_i|^2.$$

[*1] 機械学習においては厳密に損失関数が最大・最小となる関数が学習されるとは限らないが, そのような関数を求めることも学習の定義に含めるものとする.

この機械学習空間 $(\mathcal{X}, \mathcal{Y}, D, \mathcal{H}, \mathcal{L}_D)$ 上で,

$$\arg\min_{f \in \mathcal{H}} \mathcal{L}_D(f)$$

を求める問題を単回帰分析という. 単回帰分析の最適仮説 $f^* \in \mathcal{H}$ は

$$f^*(w) = \frac{\sum_{i=1}^N x_i y_i}{\sum_{i=1}^N x_i^2} w$$

となる. また, $\mathcal{H} \simeq \mathbb{R}$ である.

Example 3.2. (重回帰分析)
機械学習空間 $(\mathcal{X}, \mathcal{Y}, D, \mathcal{H}, \mathcal{L}_D)$ を以下のように定義する.
$\mathcal{X} = \mathbb{R}^d (N \geq d)$, $\mathcal{Y} = \mathbb{R}$,

$$\mathcal{H} = \{f : \mathcal{X} \to \mathcal{Y} \mid f(x) = W^T x + b, W \in \mathbb{R}^d, b \in \mathbb{R}\},$$
$$\mathcal{L}_D(f) = \sum_{i=1}^N |f(x_i) - y_i|^2.$$

この機械学習空間 $(\mathcal{X}, \mathcal{Y}, D, \mathcal{H}, \mathcal{L}_D)$ 上で

$$\arg\min_{f \in \mathcal{H}} \mathcal{L}_D(f)$$

を求める問題を重回帰分析という. $X \in \mathbb{R}^{N \times (d+1)}$ を

$$X = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{Nd} \end{pmatrix}$$

と定義し, $\mathbf{y} = (y_1, y_2, \cdots, y_N)^T \in \mathbb{R}^N$ とする. この時, $\mathbf{rank}(X) = d + 1$ ならば, 最適仮説 $f^* \in \mathcal{H}$ は $[b\ W] = (X^T X)^{-1} X^T \mathbf{y}$ とすれば

$$f^*(x) = [b\ W]^T x$$

である.

3.2 線形回帰

前小節で定義した重回帰分析は線形的なデータにしか適応できなかった. それを基底関数と呼ばれる関数列を定義する事で, 非線形データにも対応することができる.

Definition 3.3. (基底関数)
$\mathcal{X}$ を入力空間とし, 任意に $d \in \mathbb{N}$ をとる. $\mathcal{X}$ 上の連続関数列 $\{\phi_n\}_{n=1}^d \subset C(\mathcal{X}, \mathcal{X})$ が一次独立であるとき, $\Phi = (\phi_1, \phi_2, \cdots, \phi_d)^T : \mathcal{X} \to \mathcal{X}^d$ を基底関数 (basis function) と呼ぶ.

Example 3.4. (多項式回帰)
機械学習空間 $(\mathcal{X}, \mathcal{Y}, D, \mathcal{H}, \mathcal{L}_D)$ を以下のように定義する.
$\mathcal{X} = \mathbb{R}, \mathcal{Y} = \mathbb{R}$,

$$\mathcal{H} = \{f : \mathcal{X} \to \mathcal{Y} \mid f(x) = W^T \Phi(x), W \in \mathbb{R}^{d+1}\},$$
$$\mathcal{L}_D(f) = \sum_{i=1}^{N} |f(x_i) - y_i|^2.$$

ここで, $\phi_n(x) = x^n$, $n \in \{0, 1, \cdots, d\}$ とする.
この機械学習空間 $(\mathcal{X}, \mathcal{Y}, D, \mathcal{H}, \mathcal{L}_D)$ 上で

$$\arg\min_{f \in \mathcal{H}} \mathcal{L}_D(f)$$

を求める問題を多項式回帰という. $X = [\phi_1(x_1), \phi_2(x_2), \cdots, \phi_N(x_N)]^T \in \mathbb{R}^{N \times d}$ とし, $\mathbf{y} = (y_1, y_2, \cdots, y_N)^T \in \mathbb{R}^N$ とする. この時, 最適仮説 f^* は重回帰分析と同様に $W = (X^T X)^{-1} X^T \mathbf{y}$ とすれば

$$f^*(x) = W^T x$$

である.

Remark 3.5. (ハイパーパラメータと学習パラメータ)
多項式回帰の多項式の次数 $d \in$ や基底関数 $\{\phi_n\}_{n=1}^d$ のようにコンピュータに学習させるのではなく, 機械学習モデルの設計者が設定するパラメータのことをハイパーパラメータと呼ぶ. 一方, $W \in \mathbb{R}^{d+1}$ のように, データからコンピュータが自動で学習するパラメータのことを学習パラメータと呼ぶ.

3.3 過学習と正則化

前節の多項式回帰で学習を行うと以下のグラフのように, 訓練データに過剰適合してしまい, 未知のデータに関しての汎化性能が小さくなってしまうことを過学習 (overfitting) という.

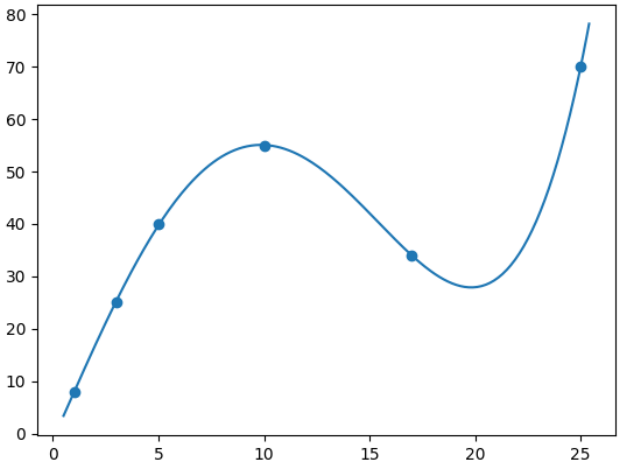

図 1　$d = 4$ の時の多項式回帰の過学習

過学習を防ぐための手法が正則化である.

Definition 3.6. (正則化)
$\mathcal{L}_D : \mathcal{H} \to \mathbb{R}$ を損失関数とする. $\mathcal{L}_D^R := \mathcal{L}_D + L^R$ とする. この時, $\mathcal{L}_D^R$ を $\mathcal{L}_D$ の正則化と呼び, L^R を正則化項と呼ぶ.

Example 3.7. (Lasso 正則化多項式回帰)
$\lambda \in \mathbb{R}^+$, $d \in \mathbb{N}$ を任意にとる. 機械学習空間 $(\mathcal{X}, \mathcal{Y}, D, \mathcal{H}, \mathcal{L}_D)$ を以下のように定義する.
$\mathcal{X} = \mathbb{R}, \mathcal{Y} = \mathbb{R},$

$$\mathcal{H} = \{f : \mathcal{X} \to \mathcal{Y} \mid f(x) = W^T \Phi(x), W \in \mathbb{R}^{d+1}\},$$
$$\mathcal{L}_D(f) = \sum_{i=1}^{N} |f(x_i) - y_i|^2 + \lambda W^T W. \qquad (W \text{ は } f \text{ のパラメータ})$$

ここで, $\phi_n(x) = x^n$, $n \in \{0, 1, \cdots, d\}$ とする.
この機械学習空間 $(\mathcal{X}, \mathcal{Y}, D, \mathcal{H}, \mathcal{L}_D)$ 上で

$$\arg\min_{f \in \mathcal{H}} \mathcal{L}_D(f)$$

を求める問題を Lasso 正則化多項式回帰という. 通常の多項式回帰の時と同様に, $X = [\phi_1(x_1), \phi_2(x_2), \cdots, \phi_N(x_N)]^T \in \mathbb{R}^{N \times d}$ とし, $\mathbf{y} = (y_1, y_2, \cdots, y_N)^T \in \mathbb{R}^N$ とする. この時, 最適仮説 f^* は $W = (X^T X + \lambda I)^{-1} X^T \mathbf{y}$ とすれば

$$f^*(x) = W^T x$$

である.

正則化の結果が以下のグラフである.

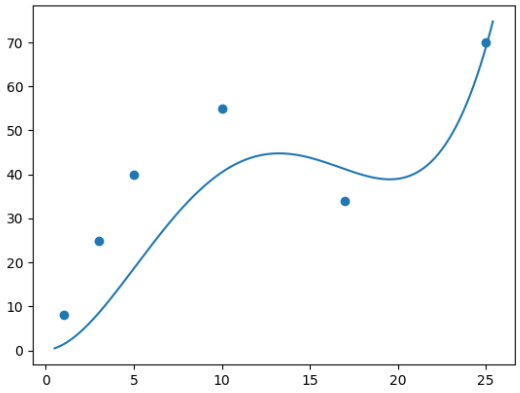

図 2　$\lambda = 700$, $d = 4$ の時の Lasso 正則化多項式回帰

このように正則化をすることで, パラメータが大きくなりすぎることを防ぐことで過学習を防ぐことができる.

Remark 3.8. (未学習について)
正則化を行う際にパラメータ $\lambda \in \mathbb{R}^{+}$[*2]は自分で決める必要がある. その際に λ の値を大きくしすぎると未学習 (underfitting) という問題が発生する. 未学習とは訓練データに対してモデルが十分に適合できていないことである.

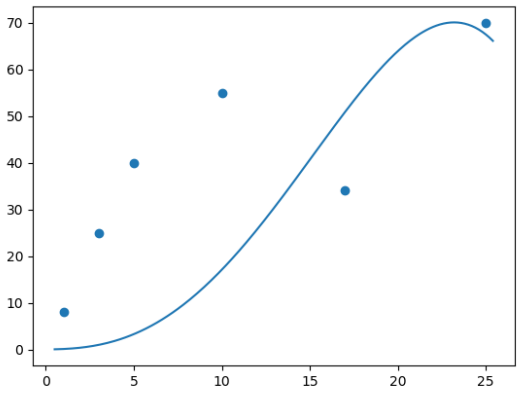

図 3　$\alpha = 30000$, $d = 4$ の時の Lasso 正則化多項式回帰の未学習

このように, 正則化パラメータは適切に選ぶことが大切である.

[*2] 正則化パラメータという.

3.4 ロジスティック回帰と勾配降下法

最後の具体例として, ロジスティック回帰を紹介する. ロジスティック回帰は分類問題を解くための手法の 1 つである.

Example 3.9. (ロジスティック回帰)
機械学習空間 $(\mathcal{X}, \mathcal{Y}, D, \mathcal{H}, \mathcal{L}_D)$ を以下のように定義する.
$\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \mathbb{R}^m$,

$$\mathcal{H} = \{f : \mathcal{X} \to \mathcal{Y} | f(x) = \sigma(W^\top x), W \in \mathbb{R}^d\},$$
$$\mathcal{L}_D(f) = \sum_{i=1}^{N} (-y_i \log f(x_i) - (1 - y_i) \log(1 - f(x))).$$

4 Deep Learning

4.1 Neural Network

4.2 The Universal Theorem of Neural Network

5 Generative Adversarial Networks

5.1 GAN の定式化

$(\Omega, \mathcal{F}, \mathbb{P})$ を完備な確率空間とする. $\mathcal{X}$, $\mathcal{Y}$ を線型空間とする.

$$\mathcal{H}_1 = \{G : \mathcal{Z} \to \mathcal{X} \ | G \text{ はニューラルネット } \},$$
$$\mathcal{H}_2 = \{D : \mathcal{X} \to [0, 1] \ | D \text{ はニューラルネット } \}$$

ここで、$\mathcal{Z}$ は潜在空間と呼ばれる $\mathbb{R}^{\dim \mathcal{X}}$ の線型部分空間である. また, 確率変数 $Z : \Omega \to \mathcal{Z}$ に対し, $g(Z)$ が従う確率分布を $\mathbb{P}$

6 Applications of GANs

参考文献

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville and Yoshua Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems, 2014.

[2] Takeru Miyato, Toshiki Kataoka, Masanori Koyama and Yuichi Yoshida, Spectral Normalization for Generative Adversarial Networks, International Conference on Learning Representations, 2018.