機械学習の数理

正則化と一様大数の法則

楠岡成雄（東京大学名誉教授）

機械学習とは何か

目的、手法により多くの種類のものが存在

教師あり学習

教師なし学習

強化学習

第 1 次ブーム (1950 年代～1970 年頃)　人工知能の理論研究の始まり

第 2 次ブーム (1980 年代)　エキスパートシステム

第 3 次ブーム (2000 年代～現在)

2015 年～2017 年　アルファ碁　 (強化学習によるもの)

人間より強いプログラム　必勝法が見つかったわけではない

強化学習

近年、ファイナンスの世界でも注目されている

理論研究

(1)　ポートフォリオ最適化

(2)　取引費用のある場合のデリバティブ価格

確率制御最適化問題

(1) はファンドの運用：うまくいけば理屈は問われない

機械学習（ブラックボックス）になじんでいる?

(2) は金融機関のリスク管理

機械学習の導いたものに対する説明責任

慎重に検討されており　まだ実用段階ではない　（らしい？）

教師あり学習

以下では主に、教師あり学習を例に考えていく

単純な数学的な定式化は難しく、多様な定式化が必要

この講演では次の例を念頭に考えていく

例「犬猫判定問題」

カラー写真データから写っているものが犬か猫かを判定する

画素数を $\tilde{d}$ とする　各画素の　赤、緑、青、3 色

それぞれの明度を b $(0 \leqq b \leqq 1)$ とすると

すべての画像データは $[0,1]^d$ の元として表現される （ $d = 3\tilde{d}$)

犬は 0 猫は 1 を返す　真の判定関数を f_0 とする

$f_0 : [0,1]^d \to \{0,1\}$ を見つけることが目的

それを「良く」模倣する関数を見つけたい

教師あり学習

過去（既知）のデータ（訓練データ）　$(x_1, y_1), \ldots, (x_n, y_n)$, が所与

$(x_k \in [0,1]^d,\ y_k \in \{0,1\},\ k = 1, \ldots, n\ )$

$y_k = f_0(x_k),\ k = 1, \ldots, n.$

f $(f : [0,1]^d \to \{0,1\}\ )$ を模倣する関数の候補とする

模倣の精度を測るもの

損失関数 （誤差関数）　$L(y, y')\ y, y' \in \{0,1\}$ を与え

$$L(f(x), f_0(x))$$

を小さくしたい

L の満たすべき性質

$L(y,y) = 0,\ y \in \{0,1\}, \quad L(y',y) > 0,\ y \neq y'\ y, y' \in \{0,1\}$

話の一般化

d 次元ベクトル x に対して実数値 y を返す真の関数 $y = f_0(x)$ がある

さらに n 個の $x_k \in \mathbf{R}^d$ に対しては

$y_k = f_0(x_k)$, $k = 1, \ldots, n$, がわかっている

$(x_1, y_1), \ldots, (x_n, y_n)$ が訓練データ

f_0 を模倣する関数 f をどのように探すか

ただし、模倣の精度を測る損失関数 $L(y, y')$, $y, y' \in \mathbf{R}$,

が与えられている

$$L(y, y) = 0,\ y \in \mathbf{R},$$

$$L(y', y) > 0,\ y \neq y' \quad y, y' \in \mathbf{R}$$

なぜ模倣関数を見つけようとするのか？

将来新たなデータ $x' \in \mathbf{R}^d$ が与えられたとき $f_0(x')$ を予測するため

もし過去データと将来データに何の関係もない

$\Rightarrow$　学習は無意味

仮設　過去データと将来データに何らかの関係がある

統計学的な関連づけ

過去データと将来データに何らかの統計的な関係がある

［仮設］　データはすべて独立同分布を持つ確率変数の実現値

非統計学的な関連づけもあり得る（現在まで考察されていない？）

［仮設］の下では統計的推定の手法を用いることが多い

以下 [仮設] の下で考える

[モデル]

パラメータ空間 $\Theta \subset \mathbf{R}^m$

パラメータ付き関数 $F : \mathbf{R}^d \times \Theta \to \mathbf{R}$ を適当に定める

(i)　θ について線形なモデル

m 個の関数 g_i, $i = 1, \ldots, m$, ($g_i : \mathbf{R}^d \to \mathbf{R}$) を用意し

$$F(x, \theta) = \sum_{i=1}^{m} \theta_i g_i(x)$$

機械学習での例:　カーネル法

(ii)　θ について非線形なモデル

代表例：ニューラルネットワーク（後に述べる）

モデルの選択：　機械学習において最も重要

　ここが人間の腕の見せ所！？

（統計的推定との類似）

良いパラメータ $\theta_0 \in \Theta$ を見つけて $\tilde{f} = F(\cdot, \theta_0)$ とする

問題点

(I) モデル $F : \mathbf{R}^d \times \Theta \to \mathbf{R}$ は適切か

モデル $F : \mathbf{R}^d \times \Theta \to \mathbf{R}$ を定めた時

(II) どのパラメータ $\hat{\theta} \in \Theta$ が「良い」のか

(III) 「良い」パラメータ $\hat{\theta} \in \Theta$ を学習データからどう見つけるのか

問題点 (I), (II), (III) は相互に関連している

損失関数 $L : \mathcal{Y} \times \mathcal{Y} \to [0, \infty]$ の選び方にも関連する

今日は (I) の問題は扱わない

X : データとなるベクトル値（$\mathbf{R}^d$-値）確率変数

$$L_0(\theta) = E[L(F(X,\theta), f_0(X))]$$

を最小にする $\theta_0 \in \Theta$ を求めたい。

しかし、$L_0(\theta)$ は未知

実際に行われる（行える）こと

過去データ $(x_1, y_1), \ldots, (x_n, y_n)$ が与えられた時

$$\frac{1}{n}\sum_{k=1}^{n} L(F(x_k,\theta), y_k)$$

を最小にする $\hat{\theta} \in \Theta$ を求める

問題あり！

線型モデル

$$F(x,\theta)=\sum_{i=1}^{m}\theta_i g_i(x)$$

$$\sum_{i=1}^{m}\theta_i g_i(x_k)=F(x_k,\theta)=y_k,\qquad k=1,\ldots,n$$

θ の一次方程式

$m>n$ ならば解は存在しないか　無数に存在するかのどちらか

従来の統計学では $m<<n$ とせよ　（AIC)

機械学習では $m>>n$ にとることが多い

過剰適合 (Overfitting) 過学習 (Overlearning)

過剰適合を回避するためのアイデア　**正則化** (regularization)

「正則化関数」$\phi : \Theta \to [0, \infty)$ を一つ定める

$\lambda > 0$ に対して

$$\frac{1}{n}\sum_{k=1}^{n} L(F(x_k, \theta), y_k) + \lambda\phi(\theta)$$

を最小にする $\hat{\theta}_\lambda \in \Theta$ を求める

正則化関数の例

(L^1 正則化)　　$\phi(\theta) = \sum_{i=1}^{m} |\theta_i|$

(L^2 正則化)　　$\phi(\theta) = (\sum_{i=1}^{m} |\theta_i|^2)^{1/2}$, $\phi(\theta) = \sum_{i=1}^{m} |\theta_i|^2$

正則化の意味は過剰適合を避けるだけか？

$$L_0(\theta) = E[L(F(X,\theta), f_0(X))]$$

を最小にする $\theta = \theta_0$ が良いパラメータ

X_k, $k = 1, \ldots,$ が独立で分布が ν であるならば

$$\hat{L}_n(\theta) = \frac{1}{n}\sum_{k=1}^{n} L(F(X_k,\theta), f_0(X_k))$$

は n が大きければ $L_0(\theta)$ に近いはず（大数の法則）

$\hat{L}_n(\theta)$ の最小を求めるのは何故だめなのか

θ を動かすので、$|\hat{L}_n(\theta) - L_0(\theta)|$ が「一様」に小さい必要がある

「一様大数の法則」

一様大数の法則のための重要な道具：Radmacher 複雑度

$n \geqq 1$

$\mathbf{R}^d$ 上の分布 ν

$\mathbf{R}^d$ 上の関数の族 $\mathcal{K}$

($g \in \mathcal{K}$ は $\mathbf{R}^d$ 上の関数 $g : \mathbf{R}^d \to \mathbf{R}$)

Radmacher 複雑度

$$\mathcal{R}_n(\mathcal{K}; \nu)$$

が定まる

Radmacher 複雑度の定義

$$\mathcal{R}_n(\mathcal{K};\nu)$$

$$= E[\sup_{g\in\mathcal{K}} \sum_{k=1}^{n} g(X_k)\sigma_k]$$

ただし、$\sigma_1,\ldots,\sigma_n, X_1,\ldots,X_n$ は独立確率変数、

σ_k, $k=1,\ldots,n$, は $\{-1,1\}$-値

$$P(\sigma_k = 1) = P(\sigma_k = -1) = \frac{1}{2}$$

X_k, $k=1,\ldots,n$, は $\mathbf{R}^d$-値確率変数でその分布は ν

Radmacher 複雑度の性質

金森敬文「統計的学習理論」　講談社

MLP　機械学習プロフェッショナルシリーズ

$\mathbf{R}^d$ 上の分布 ν, $\mathbf{R}^d$ 上の関数の族 $\mathcal{K}$

$\rho(s,x)$ $\mathbf{R}\times\mathbf{R}$ 上の関数

Lipschitz 連続性 $|\rho(s,x)-\rho(t,x)| \leqq |s-t|$, $s,t\in\mathbf{R}$, $x\in\mathbf{R}^d$, とする

$\mathbf{R}^d$ 上の関数の族 $\mathcal{K}'$ を

$$\mathcal{K}' = \{\rho(g(x),x);\ g\in\mathcal{K}\}$$

で定めると

$$\mathcal{R}_n(\mathcal{K}';\nu) \leqq \mathcal{R}_n(\mathcal{K};\nu) \qquad \text{Talagrand の定理}$$

一様大数の法則との関係

X, X_k, $k = 1, \ldots,$ が独立で分布が ν であるならば

$$E[\sup_{g \in \mathcal{K}}(\frac{1}{n}\sum_{k=1}^{n} g(X_k) - E[g(X)])] \leqq 2\mathcal{R}_n(\mathcal{K}; \nu)$$

$$E[\inf_{g \in \mathcal{K}}(E[g(X)] - \frac{1}{n}\sum_{k=1}^{n} g(X_k))] \leqq 2\mathcal{R}_n(\mathcal{K}; \nu)$$

が成立する

これが一様大数の法則というわけではない

(仮定 1)　ある $\ell_0 > 0$ が存在して」$0 \leqq L(y, y') \leqq \ell_0,\ y, y' \in \mathbf{R}$

(仮定 1) の下では、個々の θ に対しては $1 - 2\exp(-\dfrac{d_0^2}{2\ell_0^2})$ の確率で

$$|\hat{L}_n(\theta) - L_0(\theta)| \leqq \frac{d_0}{\sqrt{n}} \qquad \text{(小偏差原理)}$$

一様大数の法則の応用

$\phi_0(\theta)$ 正則化関数の種 $\phi_0 : \Theta \to [0, \infty)$

$r \geqq 1$ に対して

$$\mathcal{K}_r = \{L(F(\cdot, \theta) f_0(\cdot));\ \phi_0(\theta) \leqq r\}$$

とおく

$\mathcal{K}_r$ は関数 $x \to L(F(x, \theta), f_0(x)),\ \phi_0(\theta) \leqq r,$ の族

以下を仮定する

(仮定 2)　$C_0 > 0,\ \alpha \geqq 1$ が存在して

$$\mathcal{R}_n(\mathcal{K}_r; \nu) \leqq \frac{C_0 r^\alpha}{\sqrt{n}}, \qquad r \geqq 1,\ n = 1, 2, \ldots$$

(仮定 1)、(仮定 2) の下で以下が成立する

$d_0 \geqq \ell_0$ ならば $1 - 2\exp(-\frac{d_0^2}{2\ell_0})$ の確率ですべての $\theta \in \Theta$ に対して

$$|\hat{L}_n(\theta) - L_0(\theta)| \leqq \frac{d_0}{\sqrt{n}}(1 + \phi_0(\theta))^{1/2} + \frac{C_0}{\sqrt{n}}(1 + \phi_0(\theta))^\alpha$$

正規化関数 ϕ $(\phi : \Theta \to [0, \infty))$

$\theta_0 \in \Theta$: $L_0(\theta)$ の最小点（その中で $\phi(\theta)$ を最小にするもの）

$\hat{\theta}_\lambda \in \Theta,\ \lambda > 0,\ \hat{L}_n(\theta) + \lambda\phi(\theta)$ の最小点

$$L_0(\theta_0) \leqq L_0(\hat{\theta}_\lambda),$$

$$\hat{L}_n(\hat{\theta}_\lambda) + \lambda\phi(\hat{\theta}_\lambda) \leqq \hat{L}_n(\theta_0) + \lambda\phi(\theta_0)$$

$1 - 4\exp(-\frac{d_0^2}{2\ell_0})$ の確率で

$$L_0(\hat{\theta}_\lambda) - L_0(\theta_0)$$

$$\leqq \frac{d_0}{\sqrt{n}} + \lambda\phi(\theta_0)$$

$$+\frac{d_0}{\sqrt{n}}(1+\phi_0(\hat{\theta}_\lambda))^{1/2} + \frac{C_0}{\sqrt{n}}(1+\phi_0(\hat{\theta}_\lambda))^{\alpha} - \lambda\phi(\hat{\theta}_\lambda)$$

$\hat{\theta}_\lambda$ が θ_0 に近いことは諦める

$L_0(\hat{\theta}_\lambda) - L_0(\theta_0)$ が小さければ良い

正則化関数 ϕ, $\lambda > 0$ をどうとるべきか。

$\phi(\theta) = \phi_0(\theta)^\beta$, $\beta > 0$ とおく。

$\phi_0(\theta_0)$ がきわめて大きい場合：$\beta > 0$ をどうとってもだめ

$\beta = \alpha$ ととると

$$\lambda > \frac{C_0}{\sqrt{n}}$$

であれば良い？

$$\lambda \to \phi_0(\hat{\theta}_\lambda)$$

$$\lambda \to \phi(\hat{\theta}_\lambda)$$

を見る必要がある

深層学習

順伝搬型ニューラルネットワーク　（FNN：Forward nueral network)

$\rho : \mathbf{R} \to \mathbf{R}$：活性化関数 (activation function)

活性化関数の例

(1) $\rho(u) = \max\{0, u\}$ (非有界、一次同次、 Lipshitz 連続)

(2) $\rho(u) = \dfrac{1}{1+e^{-u}}$ (有界、Lipshitz 連続)

(3) $\rho(u) = \tanh(u)$ (有界、Lipshitz 連続)

(4) (　$\rho(u) = 1_{[0,\infty)}(u)$　) (有界、連続性無し)

もともとの NN は (4) を用いていたが、現在の機械学習では使われない

FNN のパラメータ

$m \geqq 1$ を定める　（層の数：色々な流儀あり）

$I_0 = d$,

$I_1 \geqq 1$, $I_2 \geqq 1$, $\ldots$, $I_m \geqq 1$, を定め、

$I_{m+1} = 1$ とする

パラメータ

$\vec{a} = (\vec{a}^{(0)}, \vec{a}^{(1)}, \ldots, \vec{a}^{(m)})$

$$\vec{a}^{(k)} = \{a_{i,j}^{(k)}\}_{i=1,\ldots,I_{k+1}\ j=1,\ldots,I_k+1}, \qquad k = 0, \ldots, m+1,$$

$\mathbf{R}^d$ 上の関数 $g_i^{(k)}$, $k = 0, 1, \ldots, m$, $i = 1, \ldots, I_{k+1}$ $(g_i^{(k)} : \mathbf{R}^d \to \mathbf{R})$
を以下のように帰納的に定める

$$g_i^{(0)}(x;\vec{a}) = \sum_{j=1,\ldots,d} a_{i,j}^{(0)} x_j + a_{i,d+1}^{(0)} \qquad 1\text{ 次関数}$$

$$g_i^{(k+1)}(x;\vec{a}) = \sum_{j=1,\ldots,I_{k+1}} a_{i,j}^{(k+1)} \rho(g_j^{(k)}(x)) + a_{i,I_{k+1}+1}^{(k+1)}, \quad k = 0,1,2,\ldots,m-1.$$

$g_1^{(m)}(x;\vec{a})$ が出力　: $F(x,\theta)$

$g_i^{(0)}$ は x_j の一次関数, $g_i^{(k+1)}$ は $\rho(g_j^{(k)}(x))$ の一次結合

パラメータの次元　$\displaystyle\sum_{k=0}^{m}(I_k+1)I_{k+1}$

実際のモデルではパラメータの次元はずっと小さい

多くの $a_{i,j}^{(\ell)} = 0$ とおくことが多い

また、$a_{i,j}^{(\ell)} = a_{i',j'}^{(\ell')}$ とおいたりする（CNN)、

正則化

$$w_k(\vec{a}) = \max_{i=1,\ldots,I_{k+1}} \sum_{j=1}^{I_k+1} |a_{i,j}^{(k)}|, \qquad k = 0, 1, \ldots, m,$$

$$\phi_0(\vec{a}) = 1 + |\rho(0)| + \sum_{k=0}^{m+1} w_k(\vec{a})$$

（仮定 3）$\ell_1 > 0$ が存在して $|L(y_1, y') - L(y_0, y')| \leqq \ell_1 |y_1 - y_0|$,

(ν に対する若干の仮定の下で)

$$\mathcal{R}_n(\mathcal{K}_r, \nu) \leqq \frac{\log(d+1)}{\sqrt{n}} 2^m \ell_1 r^{m+1}$$

$1-4\exp(-\frac{d_0^2}{2\ell_0})$ の確率で

$$L_0(\hat{a}_\lambda) - L_0(\vec{a}_0) \leqq \frac{d_0}{\sqrt{n}} + \lambda\phi(\vec{a}_0)$$

$$+\frac{d_0}{\sqrt{n}}(1+\phi_0(\hat{a}_\lambda))^{1/2} + \frac{\log(d+1)2^m\ell_0}{\sqrt{n}}(1+\phi_0(\hat{a}_\lambda))^{m+1}$$

$$-\lambda\phi(\hat{a}_\lambda)$$

$$||\vec{a}||_{L^1} = \sum_{k=0}^{m}\sum_{i=1}^{I_{k+1}}\sum_{j=1}^{I_k+1}|a_{i,j}^{(k)}|$$

$$\phi_0(\vec{a}) \leqq 1 + |\rho(0)| + ||\vec{a}||_{L^1}$$

$$\phi(\vec{a}) = ||\vec{a}||_{L^1}, \qquad \phi(\vec{a}) = ||\vec{a}||_{L^1}^{m+1}$$