チュートリアル

機械学習の概要

鈴木 大慈

1 はじめに

現在，機械学習は学術界だけでなく産業界においても幅広く用いられ，人工知能技術のコア技術として重要な役割をはたしている．機械学習はもともと人と同様の知能的機能を実現させるために研究開発が進められてきた分野であるが，「データから学ぶ」という過程とデータ解析・統計学の方法論がうまくマッチし，現在では狭い意味での人工知能としての使い方にとどまらず幅広いデータ科学の方法論として発展している．本チュートリアルではそのような機械学習の方法論について四回にわたり入門的な解説を行う．初回の今回は機械学習の概要として機械学習のおおまかな歴史とその代表的な問題設定および機械学習を実行するためのツールについて述べる．

2 機械学習の歴史

本節では機械学習の歴史を述べる．上述のように機械学習は人工知能の一分野であり，そのなかでも特に「学習」に焦点を合わせた方法論である．機械学習の目標は「汎化」にある．汎化とは過去の知識やデータから未来の状況について正しく予測し，最適な判断を下すことである．機械学習の歴史は，いわばこの汎化能力を獲得するための研究の歴史と考えてよい．機械学習の方法論として，大きく分けて，人間が大規模な知識データベースを構築しそこから様々な法則を学習する「知識ベースの学習」(エキスパートシステムなど)と，大量のデータからパターンを自動で抽出し学習する「統計的学習」がある．知識ベースの学習も盛んに研究されてきたが，計算機の発達によって大量のデータが手に入るようになり，またそれを処理することが可能になったことから，現在はより人の手によるチューニングが少なくて済む統計的学習が主流になっている．ここでは，統計的学習を軸にその歴史を振り返る．

最初期の発明として，Minsky と Edmonds が 1951 年に開発した Hebb 則で学習するニューラルネットワーク SNARC(Stochastic Neural Analog Reinforcement Calculator) や，Samuel が 1952 年に作成した世界初のチェッカーズのプログラムがある[12]．これらは人工知能研究の先駆けといえる研究である．より，現在の統計的学習に近い研究として 1957 年には Rosenblatt により Perceptron が提案された[17][18]．Perceptron は現在でも使われており，線形識別機の基本的な手法である．Perceptron は当時大いに話題になり，第一次ニューラルネットワークブームの火付け役になった．同時期に 1963 年に Vapnik らが線形の Support Vector Machine(SVM) を提案している[23]．しかし，同時に線形学習機の限界が指摘されニューラルネットワークの冬の時代を迎える[13]．

しかし，1980 年の福島邦彦によるネオコグニトロン[8][26]や誤差逆伝搬法の再発見[19]により，いわゆる深層学習が実現されていった．これによ

すずき たいじ．東京大学，JST さきがけ，理研 AIP.

り XOR 関数の表現など非線形な学習が可能になり，第二次ニューラルネットワークブームを引き起こした．さらに，多層パーセプトロンの万能近似能力[6]が証明され深層学習の理論的裏付けも進んだ．

しかし，深層学習は凸最適化問題として定式化されず，局所最適解しか得られないという問題がある．その問題を解決する方法として，カーネル法による SVM が提案された[5]．カーネル法は，数理最適化手法による高速な学習が可能である．

1990 年代から 2000 年にかけて，データから学習する統計的学習が発展し，統計学との融合が進んだ．それによってビッグデータ解析に機械学習が用いられるようになった．その間に機械学習において盛んに研究された統計的手法として，代表的なものは Lasso[22]に代表される高次元スパース推定や，トピックモデル[3]に代表されるベイズモデリングがある．また，大規模データにおいて高速に学習を実現させるための確率的最適化手法も発展した．確率的最適化は元をただせば 1951 年に統計学の分野で提案された Robbins と Monro による stochastic approximation に端を発している[16]．

一方で，深層学習の研究も 2006 年の deep belief net[9]といった形で継続はされていたが，しばらく研究のメインストリームからは外れていた．しかし，2012 年の ImageNet Large Scale Visual Recognition Competition（ILSVRC）において畳み込みネットワークを用いた AlexNet[11]が前年度から大幅に精度を改善させて優勝したことから深層学習研究が再興し，第三次ニューラルネットワークブームが起き現在に至っている．

3 機械学習のコミュニティ

さて，「機械学習」が研究されている主なコミュニティはどこにあるであろうか．機械学習は国際会議が重視される分野であり，その主要な国際会議として NIPS（Annual Conference on Neural Information Processing Systems），ICML（International Conference on Machine Learning），COLT（Conference On Learning Theory），AISTATS（International Conference on Artificial Intelligence and Statistics），UAI（The Conference on Uncertainty in Artificial Intelligence）がある．これらの会議に出される論文を読むことでいわゆる「機械学習業界」の研究動向は調べられる．関連分野として，統計学や数理最適化の分野は非常に機械学習との結びつきが強い．他にも関係の深い分野で国際会議が重要視される分野としてコンピュータビジョン（CVPR, ICCV, ECCV）やデータマイニング（KDD, WWW, WISDM, SIGIR），自然言語処理（ACL, NAACL, EMNLP, COLING），人工知能（AAAI, IJCAI），理論計算機科学（STOC, FOCS, SODA）などがある．

4 機械学習の問題設定

さて，ここからより技術的な話題に入る．まずは，機械学習の問題設定を紹介する．基本的に多くの機械学習手法は何らかの損失関数を設定し，その期待値を最小化することを目標とする．データからある構造を学習する際に，まずはその構造を数式で記述した数理モデルである仮説集合を設定する必要がある．仮説集合は統計モデルと呼んでもよい．$\mathcal{H}$ を仮説集合として，$\mathcal{H}$ に含まれる各元 $h \in \mathcal{H}$ に対しその「良さ」を評価したい．そこでデータの集合を $\mathcal{Z}$ とし，具体的な観測値 $z \in \mathcal{Z}$ を h がどれだけ良く記述しているかを表す損失関数 $l: \mathcal{Z} \times \mathcal{H} \to \mathbb{R}$ を用意する．損失関数の典型例として，ある統計モデルの負の対数尤度が用いられることが多い．損失関数をデータの出方について期待値をとった汎化誤差

$$L(h) = \mathrm{E}_Z[l(Z, h)]$$

を考える．汎化誤差は特に期待損失とも呼ばれる．機械学習の目標は，基本的には汎化誤差最小化

$$\min_{h \in \mathcal{H}} L(h)$$

として記述される．しかし，$L(h)$ の定義には未知の確率分布による期待値が入っているため実際には計算できない．そこで，何らかの方法でデー

タ $(z_i)_{i=1}^n$ を集めて損失関数の期待値の代わりに経験的平均を用いた訓練誤差(もしくは経験損失という)

$$\widehat{L}(h) = \frac{1}{n}\sum_{i=1}^{n} l(z_i, h)$$

を考える．しかし，訓練誤差最小化と汎化誤差最小化にはギャップがあり，単純に訓練誤差最小化を行ってしまうと「過学習」と呼ばれる現象を起こしてしまう．ここで，過学習とは観測データに混入している誤差に強く引っ張られ結果的に汎化誤差が悪くなる現象である．過学習の様子を**図 1** に示す．

過学習の存在により，統計モデリングの重要さが現れてくる．すなわち，あらゆる現象を記述できる非常に広い仮説集合を用意すれば，確かに汎用性は高いが微小な誤差に過敏に反応し過学習を起こしやすくなる．その代わり，対象の性質をうまく取り込んだ数理モデルを作り込み，現在興味のある現象を的確にとらえ無駄な情報を捨象したシンプルなモデルを用いることで，過学習を避けることができる．機械学習におけるモデリングの歴史は統計学と同様に，無駄な複雑さを入れないシンプルなモデルを構築する試みの歴史ともいえる．この問題は，学習理論の立場からは Vapnik-Chervonenkis 理論[24]によって数理的に説明され，機械学習の指導原理になっている．同様の議論は no free lunch theorem としても知られており[25]，またオッカムの剃刀としても表現される．この原理は実際に，AIC[1]や BIC[20]に代表されるモデル選択規準および高次元データ解析およびカーネル法などに現れる正則化学習などにも見て取れる．

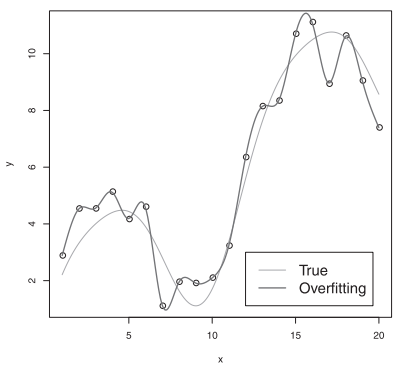

図 1　過学習の様子

4.1　教師あり学習

教師あり学習は機械学習の中でも特に基本的な問題である．教師あり学習において目指すのはある入力 $x\in\mathcal{X}$ (画像など)に対して，そのラベル $y\in\mathcal{Y}$ (その画像に写っている物体など)を当てることにある．訓練データとして n 個の入出力データの組 $(z_i)_{i=1}^n=(x_i, y_i)_{i=1}^n$ が得られているとして，仮説集合は $\mathcal{X}$ から $\mathcal{Y}$ への関数 $h(x)$ を考える．損失関数は正解ラベル y と $h(x)$ の差を評価する関数 $l(y, h(x))$ を用いる:

$$\widehat{L}(h) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, h(x_i)).$$

教師あり学習の基本的な問題として回帰がある．回帰問題においては $\mathcal{Y}=\mathbb{R}$ で，損失関数として二乗損失 $l(y, h(x))=(y-h(x))^2$ を用いることが多い．また，$\mathcal{X}=\mathbb{R}^p$ の時に $h(x)$ として線形関数 $h(x)=\beta^\top x$(だたし $\beta\in\mathbb{R}^p$)を用いれば線形回帰になる．特に，このときの訓練誤差最小化

$$\min_{\beta\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n}(y_i-x_i^\top\beta)^2$$

は最小二乗法と呼ばれる．しかし，データが高次元であったりデータ量がそもそも少ない場合には最小二乗法は過学習を引き起こす．過学習を防ぐためには，正則化法と呼ばれる手法が有用である．正則化法はパラメータ β に罰則(正則化項という)をかけ，関数 $x\mapsto x^\top\beta$ があまり複雑にならないようにする方法である．正則化項を $\phi(\beta)$ とすると正則化最小二乗法は

$$\min_{\beta\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n}(y_i-x_i^\top\beta)^2+\lambda\phi(\beta)$$

で書ける．ここで $\lambda>0$ は正則化パラメータと呼ばれ，正則化の強さをコントロールするパラメータである．正則化項としては，リッジ正則化 $\phi(\beta)=\sum_{j=1}^p\beta_j^2$ や L_1-正則化 $\phi(\beta)=\sum_{j=1}^p|\beta_j|$ が代表的である．なお，正則化学習は二乗損失だけで

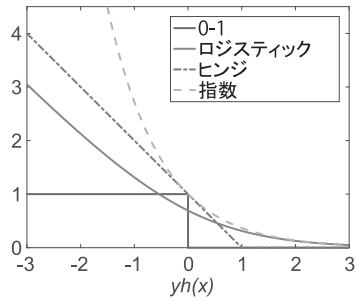

図 2　各種代理損失関数と 0-1 損失関数

なく，より一般の損失関数に対しても適用でき，また教師あり学習以外の様々な場面で用いられる．

一方，判別も頻繁に現れる重要な問題である．$\mathcal{Y}=\{\pm 1\}$ である二値判別を考えると，自然な損失関数として 0–1 損失関数が考えられる:

$$l(y, h(x)) = \begin{cases} 0 & (y = h(x)), \\ 1 & (y \neq h(x)). \end{cases}$$

しかし，0–1 損失は非凸関数であり，連続関数でもないので最適化が難しいという難点がある．そこで，0–1 損失の代わりにより扱いやすい代理損失(surrogate loss)を用いることが多い．ある実数値関数 $h:\mathcal{X}\to\mathbb{R}$ に対して，$h(x)>0$ なら 1 と判別し，$h(x)\leq 0$ なら -1 と判別するルールを考える．この時，h に対する代理損失として，ヒンジ損失 $l(y, h(x))=\max\{0, 1-yh(x)\}$ やロジスティック損失 $l(y, h(x))=\log(1+\exp(-yh(x)))$，指数損失 $l(y, h(x))=\exp(-yh(x))$ が有名である(**図 2** にその概形を図示する)．これら代理損失を用いても最適な判別機が学習できることが知られている[2]．また，損失関数としてロジスティック損失を用いた時の訓練誤差最小化をロジスティック回帰と呼び，損失関数としてヒンジ損失を用いてリッジ正則化を適用した場合は線形 SVM と呼ばれる．

4.2　教師なし学習

教師なし学習は教師あり学習と違い，入力に対するラベルが付いていない．このような問題は，トピックモデルによる文章分類に代表されるクラスタリング[3]，データの低次元圧縮や音源分離の ICA[10]などに現れる．クラスタリングの場合，観測データからその裏にある真の分布を推定することで実現されることが多い．例えば，混合ガウス分布の密度関数

$$p(x|\{\pi_k, \mu_k, \textstyle\sum_k\}_{k=1}^K) = \sum_{k=1}^K \pi_k \frac{1}{\sqrt{(2\pi)^d|\sum|}} \exp\left(-\frac{1}{2}\|x-\mu_k\|^2_{\sum_k^{-1}}\right)$$

をデータにあてはめることでソフトクラスタリングが実現できる．ここで，$\{\pi_k\}_{k=1}^K$ は混合比を表した変数で $\pi_k\geq 0 (k=1, ..., K)$ かつ $\sum_{k=1}^K \pi_k=1$ を満たす．$\{\mu_k\}_{k=1}^K$ は各クラスタの平均を表し，$\{\sum_k\}_{k=1}^K$ は各クラスタの分散共分散行列を表す．データ $(x_i)_{i=1}^n$ へのあてはめは基本的に負の対数尤度最小化

$$\min_{\{\pi_k, \mu_k, \sum_k\}_{k=1}^K} \sum_{i=1}^n -\log(p(x_i|\{\pi_k, \mu_k, \textstyle\sum_k\}_{k=1}^K))$$

で実現できる．これは EM アルゴリズムによって近似的に解くことができる[7]．もっとも，上記の最適化問題を本当に解いてしまうと過学習を起こすことが知られており(たとえばある x_i に対して，$\mu_k=x_i$ かつ $\sum_k=O$ とすれば訓練誤差を $-\infty$ にできる)，実際は適当な局所最適解を用いるか，ベイズ推定を用いて過学習を防ぐ．このようにして得られた混合ガウス分布 $(\{\hat{\pi}_k, \hat{\mu}_k, \widehat{\sum}_k\}_{k=1}^K)$ に対して，点 x が各クラスタ $\{1, ..., K\}$ に入っている確率(寄与率)は

$$\tilde{g}_k(x) = \frac{1}{\sqrt{(2\pi)^d|\widehat{\sum}_k|}} \exp\left(-\frac{1}{2}\|x-\hat{\mu}_k\|^2_{\widehat{\sum}_k^{-1}}\right)$$

とすると，$\dfrac{\hat{\pi}_k\tilde{g}_k(x)}{\sum_{k'=1}^K \hat{\pi}_{k'}\tilde{g}_{k'}(x)}$ で求められる．

4.3　半教師あり学習

半教師あり学習は教師あり学習と教師なし学習の中間に位置する問題である[4]．すなわち，一部のデータには教師ラベルが与えられ，残りのデータには教師ラベルが得られていない状況で，判別機などを構成する問題である．教師ラベルの与えられていないデータも用いることでデータの概

形が推定でき，教師ラベル付きデータのみを用いるよりも精度を上げることができる．

4.4 強化学習

強化学習は環境に合わせて自ら最適な行動を学ぶ学習問題である．例えばゲームを解く AI やロボットの動作学習などに用いられる．実際，AlphaGo は強化学習を要素技術として用いている[21]．強化学習は能動的にデータを取得しながら学習するという点で，上記の学習方法とは異なる．その応用範囲の広さから深層学習による応用研究が急速に進められている．

強化学習の基本的な手法である価値反復法(Value iteration)について説明する．強化学習では状態 $s\in S$(ゲームの局面など)と行動 $a\in\mathcal{A}$(その局面に対して行う行動)が基本要素になる．$P(s'|s,a)$ を状態 s で行動 a を取った時に，次の時刻に状態 s' に移る遷移確率(これは既知であるとして話を進める)を表し，$R(s)$ で状態 s に到達したときに得られる報酬を表す．例えば迷路を解く問題なら，$R(s)$ としてゴールすることで高い報酬が得られる関数に設定する．ある方策 π(各状態で次に起こす行動を決める関数)に従って行動する時の報酬の総和の期待値を

$$U^{\pi}(s)=\mathrm{E}\left[\sum_{t=0}^{\infty}\gamma^{t}R(s_t)\,|\,\pi,\,s_0=s\right] \quad (1)$$

と定義する．これは，状態 s から始めて未来に得られる報酬の総和であり，$\gamma<1$ は時間割引を表す係数である．強化学習が目標とするのは報酬 $U^{\pi}(s)$ を最大にする方策 π を求めることにある．もし，$U^{\pi}(s)$ が分かっているのなら，最適な方策は期待報酬を最大化する行動を取るので，

$$\pi^{*}(s)=\underset{a\in\mathcal{A}}{\mathrm{argmax}}\sum_{s'\in S}P(s'|s,a)U^{\pi^{*}}(s)$$

を満たすはずである．この時，U^{π} の定義(1)から

$$U^{\pi^{*}}(s)=R(s)+\gamma\max_{a\in\mathcal{A}}\sum_{s'\in S}P(s'|s,a)U^{\pi^{*}}(s')$$

が満たされる．これを Bellman 方程式という．しかし，実際は π^{*} も $U^{\pi^{*}}(s)$ も分からないので，状態空間を遷移しながら Bellman 方程式に従って各遷移ごとに $U^{\pi}(s)$ を更新し収束するまで続ける．このような方法を価値反復法と呼ぶ．価値反復法は $P(s'|s,a)$ を知っている必要があるが，実際はこれも推定する必要がある．これも考慮した学習方法が Q-学習である．深層学習を用いた Q-学習の方式として Deep Q-network(DQN)が[14][15]によって提案されている．DQN は Atari のゲームを高い精度で解き話題になった．

5 機械学習のツール

現在は機械学習のツールが発達しており，各種手法が簡単に利用できるようになっている．機械学習を始めるにあたり，プログラミング言語はライブラリの充実ぶりとオープンソースであることから Python が最も適していると考えられる．他には Matlab や R がよく用いられている．Python では scikit-learn と呼ばれる機械学習ライブラリが標準的なライブラリとしてよく用いられている．また，深層学習のライブラリとして代表的なものは Tensorflow, Keras, Caffe, Torch, Chainer, Mxnet がある．どれも Python で利用できる．さらに，Amazon Web Services, Google cloud platform, Microsoft Azure といったクラウドサービスも充実しており，いつでも機械学習を始められる環境が整っている．

6 ま と め

本稿では機械学習の概要を紹介した．現在，様々な場面で機械学習を用いる機会が増えている．その要因として計測機器やインターネットの発達によりデータが手に入りやすくなったことから，それらデータに基づいた意思決定を行う方法論として機械学習が重要な役割を果たしていることが挙げられる．歴史的には人工知能の一分野として発展してきた機械学習がデータ量の増加によって統計学の方法論を取り込み統計的学習という枠組みを発展させ，ひいてはデータ科学の重要な部分を占めるに至った．現在では様々な機械学習ツー

ルが揃っており，誰でも機械学習を始められる環境が整っている．しかし，機械学習の諸手法をその数理や原理まで理解して正しく用いるとなるとやや敷居が高いのも事実である．本チュートリアル連載によって機械学習導入の一助になれば幸いである．

参考文献

[1] Akaike, H., Information theory and an extension of the maximum likelihood principle; 1973. Tsahkadsor, Armenian SSR, 1971, 267–281.

[2] Bartlett, P. L., Jordan, M. I. and McAuliffe, J. D., Convexity, classication, and risk bounds, Journal of the American Statistical Association, 101[473] (2006), 138–156.

[3] Blei, D. M., Probabilistic topic models, Communications of the ACM, 55[4] (2012), 77–84.

[4] Chapelle, O., Schölkopf, B. and Zien, A., Semi-supervised Learning, Adaptive computation and machine learning, MIT Press, 2010.

[5] Cortes, C. and Vapnik, V., Support-vector networks, Machine Learning, 20[3] (1995), 273–297.

[6] Cybenko, G., Approximation by superpositions of a sigmoidal function, Mathematics of Control, Signals, and Systems (MCSS), 2[4] (1989), 303–314.

[7] Dempster, A. P., Laird, N. M. and Rubin, D. B., Maximum likelihood from incomplete data via the em algorithm, Journal of the royal statistical society. Series B (methodological), 1977, 1–38.

[8] Fukushima, K., Neocognitron: A self-organizing neural network model for a mechanism of pattern the recognitron unaffected by shift in position, Biological Cybernetics, 36 (1980), 193–202.

[9] Hinton, G. E., Osindero, S. and Teh, Y.-W., A fast learning algorithm for deep belief nets, Neural computation, 18[7](2006), 1527–1554.

[10] Hyvarinen, A., Karhunen, J. and Oja, E., Independent Component Analysis, John Wiley & Sons, 2001.

[11] Krizhevsky, A., Sutskever, I. and Hinton, G. E., Imagenet classication with deep convolutional neural networks, Advances in neural information processing systems, 2012, 1097–1105.

[12] McCarthy, J. and Feigenbaum, E., In memory. arthur samuel: Pioneer in machine learning, AI Magazine, 11[3] (1990), 10–11.

[13] Minsky, M. and Papert, S., Perceptrons: An Introduction to Computational Geometry, The MIT Press, Cambridge MA, 1969.

[14] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. and Riedmiller, M., Playing atari with deep reinforcement learning, arXiv preprint arXiv:1312.5602, 2013.

[15] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G. et al., Human-level control through deep reinforcement learning, Nature, 518[7540] (2015), 529–533.

[16] Robbins, H. and Monro, S., A stochastic approximation method, The annals of mathematical statistics, (1951), 400–407.

[17] Rosenblatt, F., The perceptron-a perceiving and recognizing automaton., Technical Report Report 85–460–1, Cornell Aeronautical Laboratory, 1957.

[18] Rosenblatt, F., The perceptron: A probabilistic model for information storage and organization in the brain, Psychological review, 65[6] (1958), 386.

[19] Rumelhart, D., Hinton, G. and Williams, R., Learning representations by back-propagating errors, Nature, 323 (1986), 533–536.

[20] Schwarz, G., et al., Estimating the dimension of a model, The annals of statistics, 6[2] (1978), 461–464.

[21] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al., Mastering the game of go with deep neural networks and tree search, Nature, 529[7587] (2016), 484–489.

[22] Tibshirani, R., Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society, Series B, 58[1] (1996), 267–288.

[23] Vapnik, V. and Lerner, A., Pattern recognition using generalized portrait method, Automation and Remote Control, 24 (1963), 774–780.

[24] Vapnik, V. N., Statistical Learning Theory, Wiley, New York, 1998.

[25] Wolpert, D. H., The lack of a priori distinctions between learning algorithms, Neural computation, 8[7] (1996), 1341–1390.

[26] 福島邦彦，位相ずれに影響されないパターン認識機構の神経回路モデル―ネオコグニトロン―，電子通信学会論文誌 A, J62-A[10] (1979), 658–665.