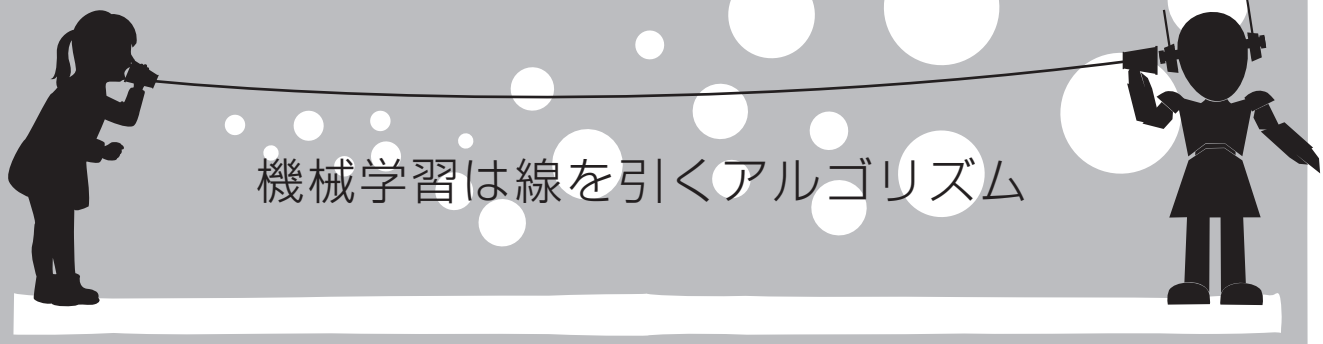

機械学習は線を引くアルゴリズム

正会員 間下以大†

1．まえがき

筆者は機械学習のアルゴリズムを一言で説明するために，“線を引く”アルゴリズムと言っている．もちろんこれは極端な解釈ではあるが，この“線を引く”ということを理解していれば．多くのアルゴリズムの理解に役立つはずである．わかりやすくするため”線を引く”と述べたが，線を引くアルゴリズムは識別モデル（discriminative model）と呼ばれる．そして”線を引く”とは異なるアルゴリズムもあり，それらは生成モデル（generative model）と呼ばれる．今回の講座ではこの線を引くアルゴリズム（識別モデル）について解説する．生成モデルについては第3回で解説予定である．

2．識別モデルと生成モデル

識別モデルと生成モデルの違いを簡単に述べるために，図1のような●と○のどちらかのラベルに分類する問題を考える．識別モデルの場合，図2のように線を引くことで分類を実現する．識別モデルはまさに図2のように線を引くことで線の右側と左側（実際には正負で変わる）に空間を分割し，そのどちらに属するかで未知の入力を識別する．生成モデルの場合，図3のように各ラベルの分布の形状やデータが生成される過程をモデルとして表現する．図3の破線は各データが生成される確率の高さを表す等高線である．そうすることで，新しい入力に対してそのデータがそれぞれのラベルに属する度合いを計算し，分類結果を得る．通常，ある入力に対して各ラベルに属する度合い（確率）を出力する（確率）モデルとして表現する．

機械学習で入力となるのは特徴量ベクトルと呼ばれる元のデータに何らかの処理を行い，識別を行うのに有用と考えられる情報である．前回の例では林檎の識別に色のヒストグラムを特徴量ベクトルとした．この特徴量ベクトルはパターン認識において非常に重要であり，識別の性能に関してはどのような機械学習のアルゴリズムを用いるかよりも目的に適した特徴量をどのように得るかが重要であることが多い．本稿の図では紙面上で図示するために1次元や2次元で特徴量ベクトルを表現しているが，通常はもっと高次元のデータになる．そしてその特徴量空間は1次元低い部分空間によって二つに分けることができる．例えば，2次元空間は1次元の線によって二つに分けられ，3次元空間は

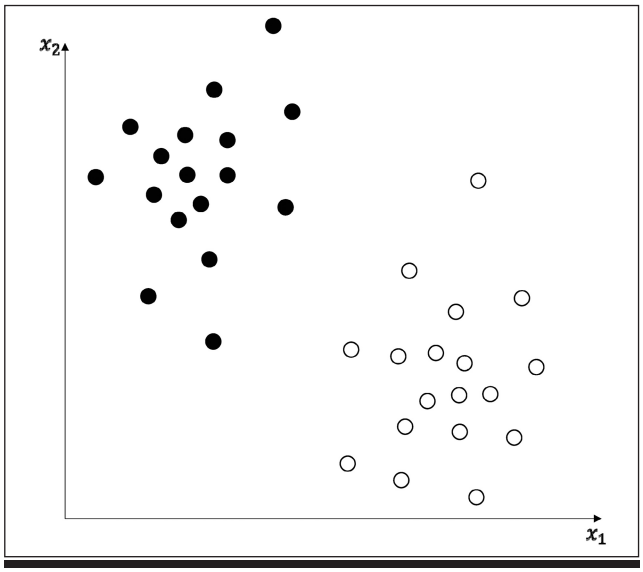

図1　2クラス分類の問題例

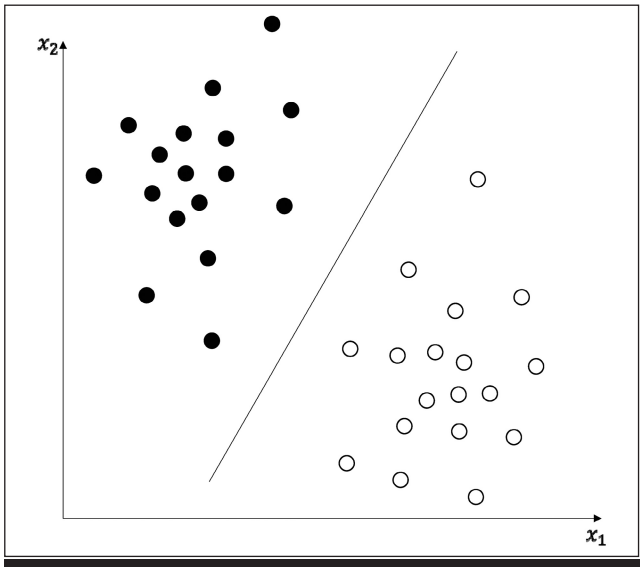

図2　識別モデルによる分類

† 大阪大学
"A Machine Learning Primer (2): Machine Learning is Line Drawing Algorithm" by Tomohiro Mashita (Osaka University, Osaka)

映像情報メディア学会誌 Vol. 72, No. 3, pp. 412～417（2018）

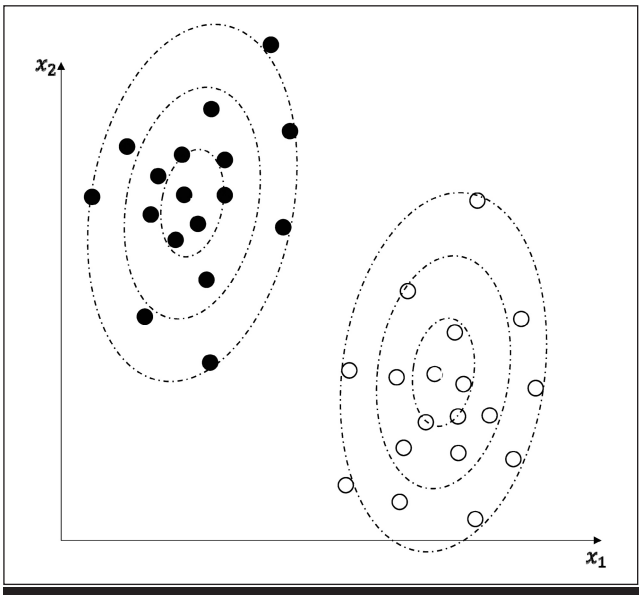

図3　生成モデルによる分類

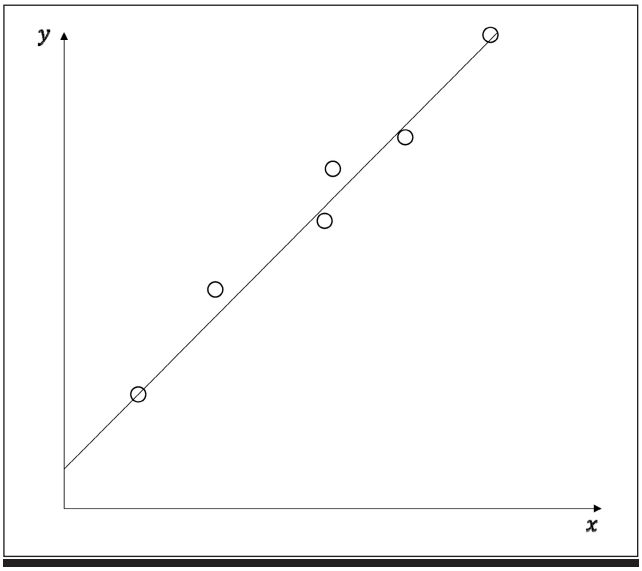

図4　回帰の例

2次元の平面によって二つに分けられる．この1次元低い部分空間は超平面（Hyper plane）と呼ばれ．“線を引く”の“線"とはこの超平面のことである．

識別モデルはさらに分類と回帰に分けられる．分類モデルでは入力に対してあらかじめ定義されている非連続な値のどれかが出力される．通常，その離散値をラベルとして扱い，ラベル付け問題を推定する．出力が0か1といった二つの値を持つ分類を2クラス分類と呼び，三つ以上の場合多クラス分類と呼ぶ．

一方，回帰モデルでは入力に対して出力される値が連続値を持つ．問題によっては出力も多次元となることもある．

単純な例を用いて図示すると，2次元（x_1, x_2）の2次元の入力に対する2クラス分類を図で示すと図2のようになり，1次元（x）から1次元（y）への回帰を図で示すと図4のようになる．どちらの場合においても，2次元の平面上に線が引かれていることがわかる．最初に述べたように，識別モデルのアルゴリズムはこの線を引くアルゴリズムであり，さまざまなアルゴリズムが発明されている．実際の問題に適用するアルゴリズムを決定する場合，識別性能は各アルゴリズムが描く線の複雑さや学習データの数，次元数，性質等によって変わってくる．問題によっては計算リソースや求められる処理時間等に制限があり，その制限によっても適切なアルゴリズムは変わる場合もある．

3．線はどのようにして決定されるのか

もう一度図1のように，二つのラベル（●と○）に分類する問題を考える．先ほど説明したように二つのラベルを分類するには図2のような直線を引くことができればよい．直線の方程式は

$$y - ax - b = 0$$

として表すことができる．つまり，ある入力$(x, y) = (x', y')$があるとすると，$y' - ax' - b$の値が正になる空間と負になる空間に分けられ，正の場合には●，負の場合には○と見なすことで分類結果がえられる．ここで，$y - ax - b = 0$としているが，実際に分類を行うためには具体的なa, bの値を決める必要があり，このa, bの値を既知のラベル付けされたサンプルデータから決定するのが機械学習である．この，既知のラベル付けされたサンプルデータは教師付データ（supervised data）や学習データ（training data）等と呼ばれる．実際の問題では高次元の入力になり，N次元空間を分割する超平面は

$$y = w_0 + \sum_{i=1}^{N} w_i x_i$$

という形で表現され，$w_i (i = 0, 1, ..., N)$を求める．

この例は最もシンプルな場合であり，1本の直線（線形な超平面）で正しく分離することが可能な分類対象となっている．これを，線形分離可能と呼ぶ．実際の分類問題では線形分離可能な場合は少ないが，線によって分類するという機械学習の性質が良く表されている．また，線形分離できない問題であっても線形分離の組み合わせによって適切な決定境界を得たり，入力データを何らかの形に変換したりすることで線形分離可能な状態にすることがよく行われる．回帰問題の場合も線を引くという点でまったく同様であり，入力と出力の対応を表す線のパラメータを求めることで未知の入力に対する推定値を得ることができるようになる．

また，多値の場合も線が複数になるだけであり，本質的には同じ問題である．アルゴリズムによって他クラスにそのまま適用可能なものと2クラス分類の組み合わせが必要なものがあり，2クラス分類を組み合わせる方法には1対他（Onve-vs-the-rest）や1対1（One-vs-One）といった組み合わせ方法がある．

3.1 過学習と汎化性能

先ほど述べたように，実際のパターン認識の問題では線形分離できることは稀である．そのため，機械学習のアルゴリズムは複雑な決定境界を表現できるようになっている．しかし，複雑な決定境界が常に良い性能の識別器になっているとは限らない．

教師あり学習では未知の新たな入力に対して正しい識別を行うために，既知の正解付きデータから決定境界を求める．学習用のデータが図5(a)のような状態であった場合，緩やかな曲線で決定境界を表現すると図5(b)のようになる．一方，複雑な曲線で表現すると，図5(c)のような決定境界も作成可能である．学習用のデータに対する性能という点では，(c)の方が誤識別がなく良い決定境界と言えるが，図5(b)で誤識別となっているデータは何らかの理由で誤識別となるような値のデータとなったのかもしれない．ここで何らかの理由とは，データの計測時のエラー，教師ラベルの付け間違い，そもそも特徴量が良くない等，さまざまな理由が考えられる．実際のパターン認識の問題ではさまざまな理由でこのような外れ値が発生している．このような外れ値を識別するために複雑な決定境界を用いるとその外れ値付近で誤識別が発生しやすくなり，学習データに対する性能は良いが，未知のデータに対しての性能は悪くなるという状態になる．このような学習を過学習(Overfitting)と呼び，未知のデータに対する性能を汎化性能と呼ぶ．

過学習にならないために，交差検証(Cross validation)という技術が用いられる．交差検証には幾つか方法があるが，代表的な方法は学習データをK個に分割し，一つを評価用残りを学習用のデータとして学習と評価を行う．これをK個に分割されたデータセットについてそれぞれ行い，その平均を用いる方法である．

3.2 ハイパーパラメータ

後述する機械学習のアルゴリズムは学習によって決定されるパラメータの他に，あらかじめ与えるパラメータを持っている．例えば，多項式回帰の多項式の次数，K最近傍法のKの値等が該当する．深層学習におけるネットワークの深さや，活性化関数の選択，畳み込みニューラルネットのチャネル数等も含まれる．これらは総称してハイパーパラメータと呼ばれ，その値によって識別器の汎化性能や処理時間が大きく変化することが多い．そのため，実際の問題では単に識別器に学習データを入力するだけでなく，ハイパーパラメータのチューニングが必要になる．このチューニングは人手による試行錯誤によって行われることが多く，ハイパーパラメータの数が増えるほどチューニングは難しくなる．また，一度の学習とテストで確認できるのは一つのハイパーパラメータの値なので，学習に時間がかかる場合，パラメータの試行錯誤そのものが難しい場合もある．深層学習が黒魔術と呼ばれる原因の一つはこの

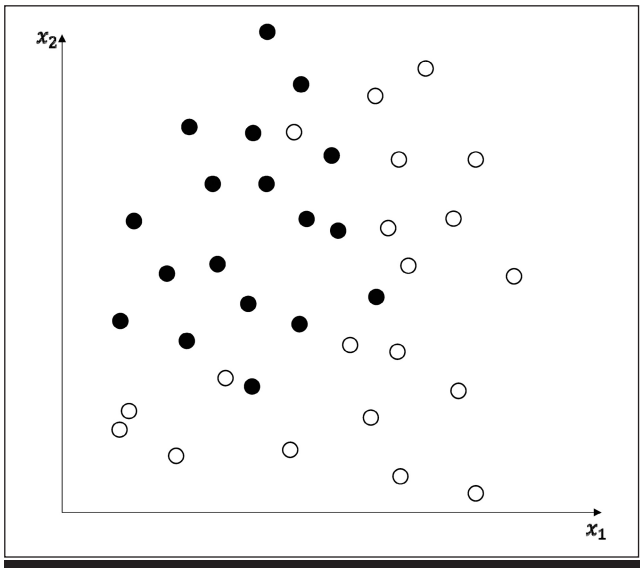

図5(a) 複雑な境界となる問題の例

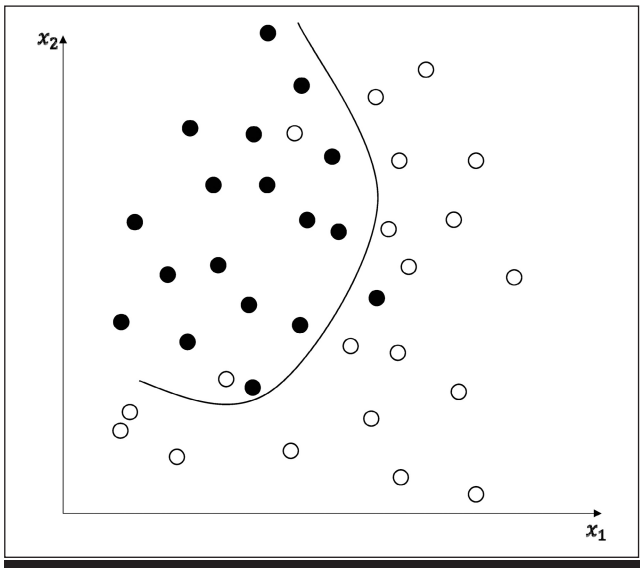

図5(b) 緩やかな曲線による決定境界

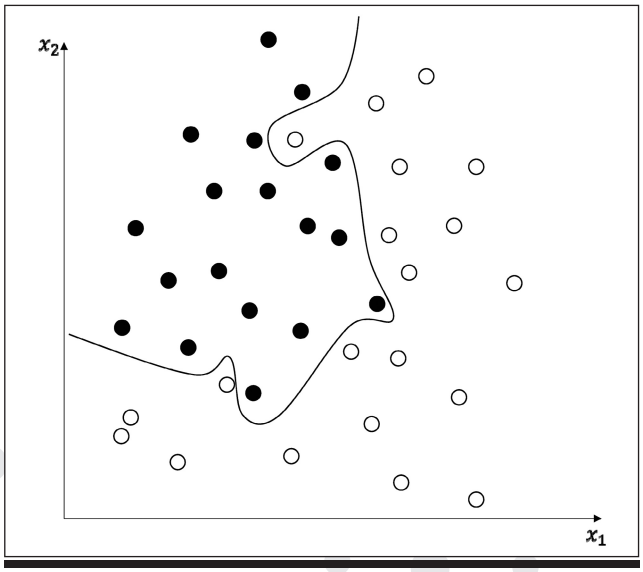

図5(c) 複雑な曲線による決定境界

チューニングの難しさにある．一方，後述するサポートベクターマシンやRandom Forest等は比較的ハイパーパラメータが少なく，チューニングが容易である．

4．機械学習のアルゴリズム

4.1　線形判別分析

先ほど直線を引くと述べたが，図1のような分布の場合，与えられているサンプルデータに対して正しく分離することができる直線は，図6に示すようにいくつも存在する．例えば，図6の線aや線bのような決定境界の場合，片方のラベル対しては確実に分類できるが，もう片方のラベルについては学習データから少し離れた位置の入力に対して誤識別をすることがわかる．また図6の線cもサンプル点に近く，その近いサンプル点から少し外れただけで誤識別が起きそうである．このデータの場合，線dのような決定境界が良さそうである．このように，無数に考えられる線の候補の中で未知のデータに対しても良い結果を出力できる線を決定する必要がある．適切な決定境界を決定する方法の一つが，線形判別分析（Linear Discriminant Analysis: LDA）である．線形判別分析では，クラス間の分散を分母に，クラス内の分散を分子にとった評価関数を最小化するような評価軸を作成し，その評価軸上の正負で2クラスの判別を行う．分散が大きくなる軸，小さくなる軸とは図7のような軸であり，データの広がりが大きい方向と小さい方向と考えればよい．先ほどの例で，クラス内の分散が小さくなる軸とは図8の軸aやbであり，クラス間分散を大きくする軸とは図8の軸cのような．各クラスの中心の距離が大きくなる軸になる．これらを総合した結果，図8の軸dに示すように，分母であるクラス間の分散を大きく，クラス内の分散を小さくとるような軸で入力を評価する．言い換えればその軸と直交する超平面（図8の線e）が決定境界となる．なお，図6の線aやbを悪い決定境界としたが，実際にはそうでない場合もある．例えば，誤識別に対するリスクが極端に異なる場合，そのリスクを考慮すると図6のaやbのような決定境界が適切となる場合もある．

4.2　最近傍法，K最近傍法

サンプルデータを直接利用したシンプルな手法が最近傍法やK最近傍法である．最近傍法は入力された未知のデータと最も近い距離にある教師データと同じラベルや値を出力する方法，K最近傍法は最も近い上位K個のデータから出力を決定する方法である．K個の最近傍サンプルを等しく扱ってもよいし，距離に応じて重みを付けることもできる．回帰問題の場合は，距離の逆数を重みにする等して出力値の重み付き平均をとれば連続値としても値を得ることができる．K最近傍法によって得られる決定境界はKが小さい場合には細かく複雑になり，Kが大きくなるとゆるやかな曲線となる．また，K最近傍法は上述のような決定境界のパラメータを持たずに決定境界が決まるため，ノンパ

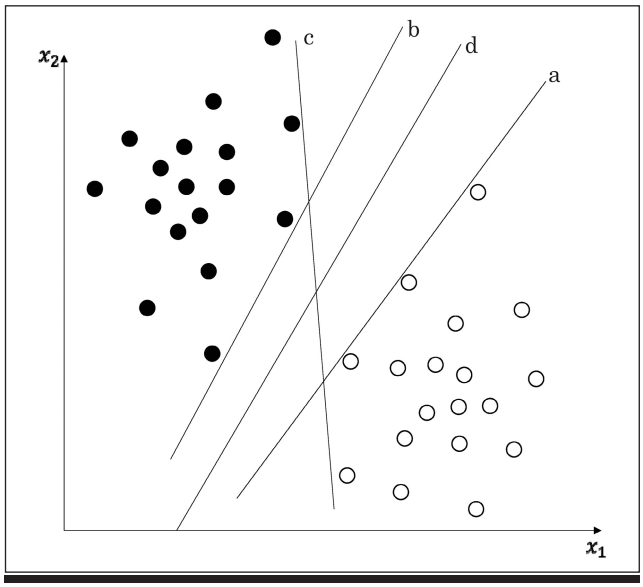

図6　複数の決定境界候補の例

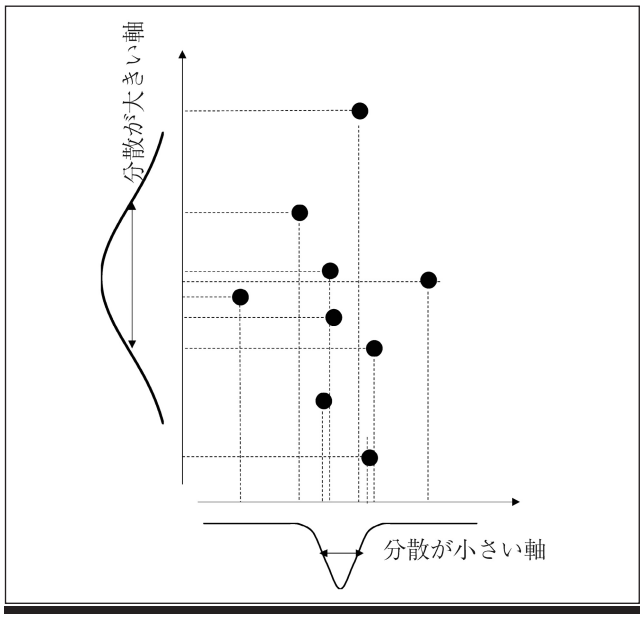

図7　分散が小さい軸と大きい軸

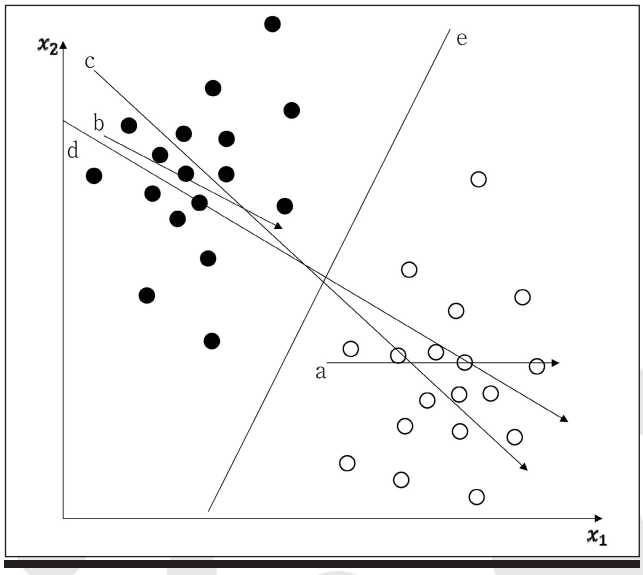

図8　線形判別分析の例

ラメトリックモデルと呼ばれる手法の一つである．

最近傍法では，未知のデータが入力される毎に各教師データとの距離を計算し，それらをソートして最近傍点を決定するため計算量が大きい．最近傍法による識別器の性能を上げるにはなるべく多くの教師付データを用意する必要があるため，計算時間と性能のトレードオフが問題となる．計算を効率化するためにkd-treeといった探索アルゴリズムがある．

最近傍法に限らないが，距離の計算についても注意が必要な場合がある．例えば，長さと重さといった同一に扱うことができない多次元の値を入力とする場合，重み係数を決める等の方法で一つの距離尺度にする必要があり，その距離尺度によって性能が左右される可能性がある．

このように，最近傍法は機械学習のアルゴリズムの中でも最も単純な方法であり効率的ではないが，かといって性能が悪いとは限らない．例えば，充分に密度の高い教師データが得られており，処理時間に制限がない場合，優秀な識別器となる．このことは機械学習において重要であり，最新の複雑な識別器を高性能な計算機で時間をかけて学習した結果，最近傍法の方が汎化性能が良いということも起こりえる．特に次元の低いデータは比較的教師データの密度を上げやすいため，最近傍法を基準として性能を評価することも考慮すべきである．

4.3　サポートベクターマシン

深層学習が大ブームとなる以前はサポートベクターマシン（Support Vector Machine）が良く用いられる識別器の一つであった．SVMはマージン最大化という考え方に基づいて決定境界が求められる．線形判別分析でも述べたように，線形分離可能な2クラス分類問題であったとしても，その決定境界は一つではなく，どのように定めるかによって識別器の汎化性能が変化する．これについて，SVMでは図9に示すように，決定境界と各クラス内のサンプルとの最短距離（マージン）を最大化する基準によって汎化性能の良い決定境界を得る．

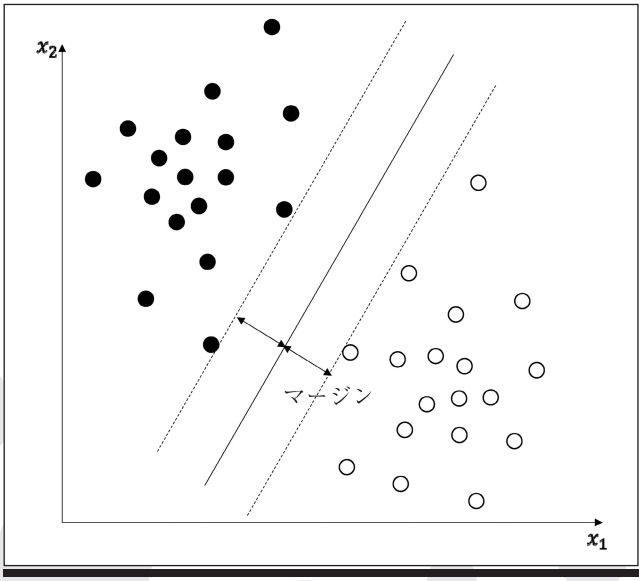

図9　サポートベクターマシンにおけるマージン

上記の説明は線形分離可能な場合についてであったが，多くの問題は線形分離可能ではない．SVMではそのような問題に対応するため，ソフトマージンとカーネル法と呼ばれる技術を用いる．ソフトマージンは決定境界を越えるようなサンプルをある程度許容する方法である．カーネル法は線形分離できない場合，つまり直線で二つのクラスを分離できない場合に，xをカーネル関数と呼ばれる関数によって別の空間に変換し，その空間内で線形に分離する技術である．

カーネル関数による変換では通常，非線形で入力よりも高次元の空間に変換され，その場合元の空間での決定境界は非線形なものになる．図10のような線形分離できない場合に，図10の×印からの距離の二乗を加えた3次元の空間で考えると，線形分離できる．×印からの距離を横軸にとり，縦軸にその2乗をとると図11のようになり，線形分離が可能になっていることがわかる．そして，図10の空間

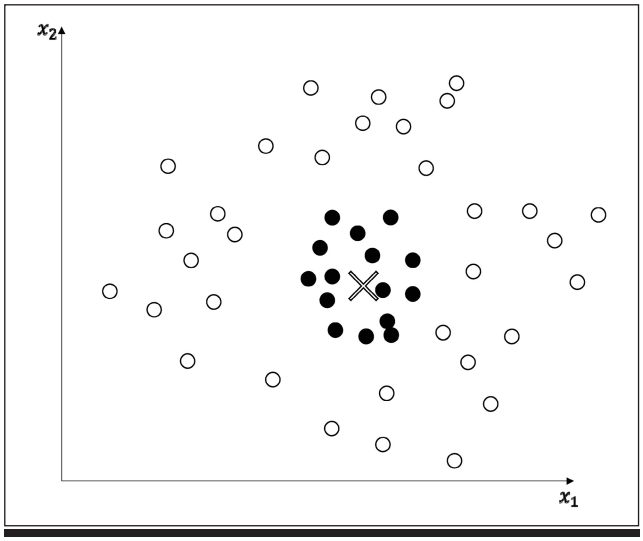

図10　線形分離できない例

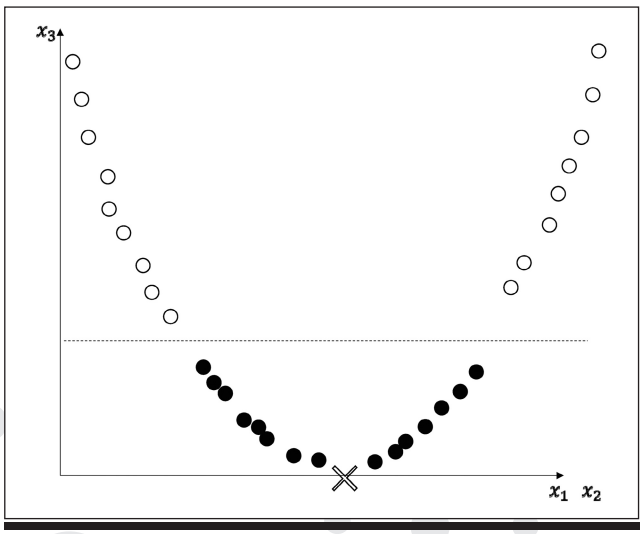

図11　カーネル法による線形分離の例

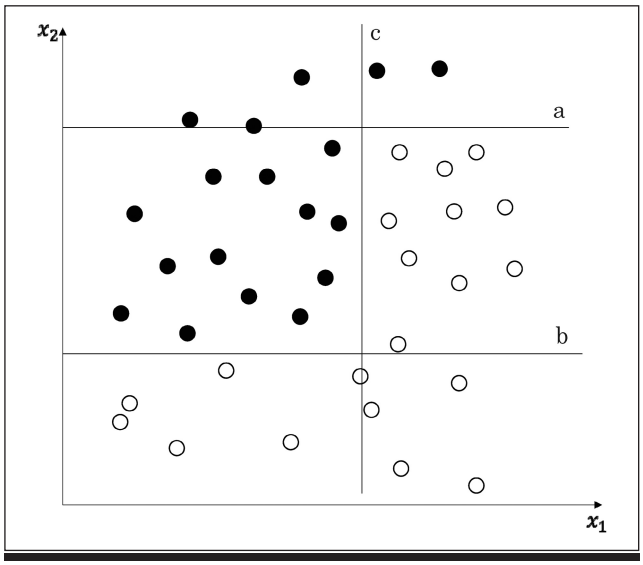

図12　弱識別器の多数決による識別例

では×印を中心に円を描くような曲線が描かれる．このようなカーネル法はSVMに限らず，特徴量ベクトルを拡張する方法としてさまざまな識別器に応用される．

上記の例は，カーネル法をシンプルに表した一例であり，カーネル関数はさまざまなものがある．そのため，問題によって適切なものを選択する必要がある．実際にSVMを用いている例ではRBF（Radial Basis Function）カーネル（別名，ガウシアンカーネル）を用いることが多い．SVMはチューニングの必要なハイパーパラメータが少なく汎化性能に優れている識別器を作成しやすいとされており，先述のように深層学習以前には良く用いられる識別器の一つであったと思われる．

4.4　Adaboost

Adaboostはboostingと呼ばれる技術を用いた識別器である．boostingでは多数の弱識別器（weak classifier）の合議（committee）によって良い識別性能を実現する．例えば，ある2クラス分類問題に対して一つの弱識別器の性能が50%を少し越える程度であったとしても，それらを多数組み合わせることで良い性能が得られる．図12に簡単な例を示す．図12では直線a～cがそれぞれ弱識別器である．個々の直線では●と○を正しく分類できていないが，直線の組み合わせによって正しく識別されることがわかる．Adaboostでは前の識別器で誤識別されたサンプルの重みをつけることで新たな識別器を適応的（Adaptive）に生成する．

Adaboostは前回の講座でも紹介したViola and Jonesの顔識別[2]でも用いられている．この手法では，Haar like特徴量とAdaboostの組み合わせが良く機能しており，高速な処理を実現している．また，Adaboostの適応的な性質を利用したEnsemble tracking[3]では，画像中の追跡対象の変化に応じて弱識別器を再構成することで適応的な物体追跡を実現している．

4.5　Random Forest

Random Forest[4]は決定木（decision tree）を弱識別器として多数用い，合議による識別を行う識別器である．決定木は第1回で解説したif-thenルールのように，ある値についての判断を繰り返して求める出力を行う方法であり，データからそのルールを学習させることもできる．決定木は処理の過程が単純で理解しやすいが，識別器としての性能は良くないことが多い．Random Forestではこの決定木を多数用いることで高性能な識別器を実現する．Random Forestや先述のBoostingのように，多数の識別器を用いる方法を集団学習（ensemble learning）と呼び，多数の識別器の合議によって汎化性能が良くなることが知られている．また，Random Forestはハイパーパラメータが少なく使いやすい識別木と言える．

5．むすび

今回の講座では，教師付学習による識別モデルについて解説した．識別モデルと生成モデル，分類と回帰等，機械学習の基本的概念を説明したのち，代表的なアルゴリズムについて簡単な解説を行った．各アルゴリズムの詳細については，第1回で引用した書籍を参照して頂きたい．ニューラルネットも機械学習のアルゴリズムであり，同列に解説することもできるが，本講座の後半で解説予定のため今回の講座では省略した．

機械学習のアルゴリズムは幾つも存在するが，その性質はそれぞれ異なる．深層学習ブームで機械学習＝深層学習と考えたり，深層学習が常に最良の選択と考えたりする人もいるようだが，問題によっては最近傍法が最も適切な場合もあるし，決定木が最適な手法の場合もある．第1回の講座でも述べたように用意できる教師データの数，実際に使える計算リソース，性能によって適切なものを選択すべきであり，そのためには識別器と問題の特徴をよく理解することが重要である．

（2018年3月13日受付）

〔文　献〕

1）Y. Freund and R.E. Schapire: "A decision-theoretic generalization of on-line learning and an application to boosting", Journal of computer and system sciences, **55**, 1（1997）, pp.119-139
2）P. Viola and M.J. Jones: "Robust real-time face detection", International journal of computer vision, **57**, 2, pp.137-154（2004）
3）S. Avidan: "Ensemble tracking", IEEE transactions on pattern analysis and machine intelligence, **29**, 2（2007）
4）L. Breiman: "Random forests", Machine learning, **45**, 1, pp.5-32（2001）

間下（ましたに）　以大（ともひろ）　大阪大学大学院基礎工学研究科博士後期課程修了．現在，同大学サイバーメディアセンター准教授．2012年，オーストリア・グラーツ工科大学客員研究員．コンピュータビジョン，機械学習，拡張現実に関する研究に従事．博士（工学）．正会員．