J. Plasma Fusion Res. Vol.92, No.11 (2016) 839-849

講座 画像再構成とパターン認識の数理

3．パターン認識入門～人工知能の基盤技術

3. Pattern Recognition and Image Informatics

内田誠一
UCHIDA Seiichi
九州大学大学院システム情報科学研究院

（原稿受付：2016年6月23日）

パターン認識とは，画像などのメディアデータおよび各種センサ信号を対象として，それらの分類および類似性解析の根幹を担う技術である．計算機科学の分野では，人工知能の中心的課題の一つとして非常に活発な研究がなされている．本章では，パターン認識の基本的な考え方を平易かつ直感的に論ずる．また，パターン認識の根幹課題である「識別」について，最近傍識法，サポートベクトルマシン（SVM），ディープニューラルネットワーク（DNN）という代表的な3つの手法の概要を紹介する．

Keywords:
Pattern recognition, classification, feature extraction, nearest neighbor, support vector machine, deep neural networks

3.1 はじめに

パターン認識とは，「人間の持つ多様な認識機能」を計算機において実現することを目的とした理論・技術である．「そもそも『認識』とは何か」という議論は古くギリシャ哲学からの命題にもなっているが，本稿の範囲では「これは何ですか？」という問いに答える分類技術と思って差し支えない．例えば，バスが写った画像を計算機に入力して「これはバスです」と答えさせるタスクがパターン認識（この例の場合は画像認識）である．なお，認識は知能の重要な機能であるため，パターン認識はいわゆる人工知能の一分野と位置付けられている．

プラズマ研究の分野でのパターン認識応用としては，画像認識を連想するのではないだろうか．プラズマのイメージング計測ではプラズマを様々な手法で画像化する．このためイメージ画像から物理法則を抽出する際に，パターン認識の技法は非常に有用なツールになると考えられる．また画像だけでなく，物理現象の様々な測定値もやはりパターンとみなすことができる，それらを分類したり，同一視したりする場合には，本章で述べる方法が利用できる．

パターン認識が扱う問題を形式的に定義すると以下のようになる：「入力されたパターン $\boldsymbol{x}$ が，事前に設定された C 個のクラス（分類先）のうちのどの1つに属するかを決定する」．パターンという用語はデータと読み替えたほうが分かりやすいかもしれない．冒頭の例では，パターン $\boldsymbol{x}$ が画像であり，C 個のクラスが認識したい物体（Visual object）の種類の集合である．この様子を**図1**に示す．左側が入力パターン $\boldsymbol{x}$ の例であり，右側がクラスである．

パターン認識の興味深い点は，「人間にはほとんど無意識にできるのに，計算機には非常に難問である」という事実である．本稿の読者は今まさに文字を読んでいる．すなわちこれは各文字（画像）がどの文字種であるかを逐次認識していることになる．実はその前に，どこに文字があるかという検出問題（画像の一部が文字か文字でないかを分類する問題に相当）や，文字行から各文字に「切り出す」処理もやっている．そのことを意識しながら本稿を読んでいる読者はいないと思われる．一方，計算機にとってこれは難問中の難問である．実際，認識のために「ここに1文字がある」と切り出すためには，その前に「その黒画素の塊が1文字である」と認識しなくてはならない．したがってパターン認識は，切り出しと認識という，いわゆる「卵と鶏の問題」を内包しているのである．紙面上の文字はまだ簡単なほうである．我々が日々目にしている情景は実に様々な物体で構成されており，それらを個々に切り出しな

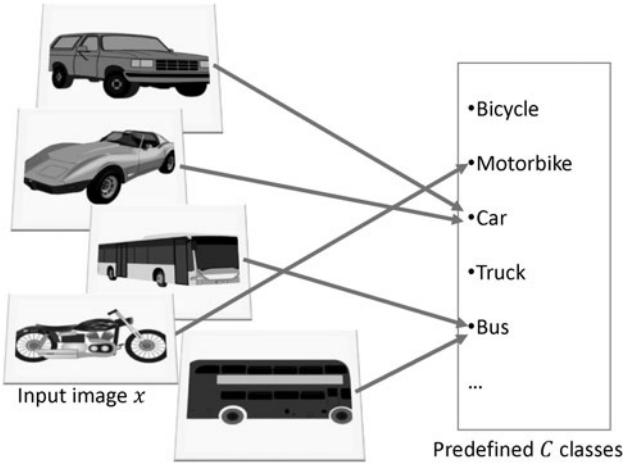

図1　パターン認識の目的（画像認識の場合）．

Graduate School and Faculty of Information Science and Electrcal Engineering, Kyushu University, FUKUOKA 819-0395, Japan
author's e-mail: uchida@ait.kyushu-u.ac.jp

がら認識する技術は，現在でもやはり難問として君臨している．

難問とはいえ，計算機によるパターン認識が実応用されるケースはますます増えている．ハガキの郵便番号認識，バーコードリーダ，デジカメの顔検出，指紋照合などは，実用化されて久しい．昨今では，これらに加えて，スマートフォンの音声入力，画像検索（画像をクエリとした検索），自動運転，家庭用ロボットなどが挙げられる．これらいわゆる人工知能として喧伝されている実用技術の多くは，最近の劇的な認識精度向上により初めて実用化に至ったものである．

こうした認識精度向上の背後には，次の三つの理由が存在する．すなわち，第一に大規模なデータセットの整備，第二にディープニューラルネットワークのような複雑な認識手法の開発，そして第三にオープンサイエンス，である．オープンサイエンスの考え方により，これら大規模データと認識手法のソースコードが「誰でもダウンロードしてすぐに使える」ようになっている．こうして参入障壁が低くなったことで，パターン認識研究は今後しばらく加速度的な進歩を遂げるものと予想されている．

本稿の主目的は，パターン認識の基本的な考え方を平易かつ直感的に論ずることにある．したがって具体的なソフトウエアやライブラリの使い方といったハウツーについては論じないし，数式を用いた手法の詳細解説についても割愛する．本稿で興味を持った読者は，必要と思われる手法について，Web や専門書を当たっていただければ幸いある．そういう場合に，本稿で述べた基本的考え方が役に立つことを願うばかりである．なお，本稿は完全なる初学者を対象としており，物理学や数学のエキスパートである本誌の読者には，極めて冗長に思われる個所も多いと思われる．専門分野の違う筆者による解説ということでご容赦いただきたい．またパターン認識の中心的役割を果たす識別法については，最近傍識法，サポートベクトルマシン（SVM），ディープニューラルネットワーク（DNN）という代表的な 3 手法について各々章を設け，極力平易に解説した．

パターン認識についてさらに学びたい読者は，入門書として定評のある[1]や，より実用的な入門書として[2]などがある．さらに体系的に学びたい場合は，バイブルとされる[3,4]を勧める．後述のサポートベクトルマシンやディープニューラルネットワークのように個別の手法についても，それらの重要性から，良質の入門書が近年多数発刊されている．

本誌においても今年 5 月号に，“核融合プラズマ研究におけるデータマイニングの活用”という小特集が組まれている．同特集の文献[5]においては，パターン認識を含めた様々な解説がなされていて参考になる．同文献に比べて，本稿は特にパターン認識にフォーカスし，その基本となる「考え方」に重点を置いて解説したものである．

3.2 パターン認識の問題設定

パターン認識は，「入力されたパターン $\boldsymbol{x}$ が，事前に設定された C 個のクラスのうちのどの 1 つに属するかを決定する」問題である．このシンプルな問題でも，何を認識対象とするか（3.2.1節），その場合のクラスは何か（3.2.2節），パターン $\boldsymbol{x}$ をどのように表現するか（3.2.3節），どのようにクラスに割り当てるか（3.2.4節）といった，バリエーションがある．さらに実際にパターン認識システムを実装して性能評価までする場合には，どのようなデータを準備する必要があるか（3.2.5節），またどのように認識手法を評価するか（3.2.6節），といった課題もある．人工知能が喧伝され，「何もかも自動にできる」ように誤解されがちであるが，実際にはこのように事前に考えなくてはならないことは結構ある．再考や試行錯誤など泥臭い作業も往々にして必要になる．

以下，**図 2** に従って，最初に考えるべきことから順に，どのようにパターン認識問題を設定していくかを解説する．その後次節以降にて，実際の問題の解き方について解説する．

3.2.1 対象とするパターン ～何を認識するか？

最初に考えるべきことは，当然ながら，「何を認識するか」すなわち対象の設定である．3.1章で述べた実応用例からもわかるように，パターン認識の対象には実に様々なものがある．画像や音声といったメディアデータ，行動データ，人流や交通流データ，気象観測データ，文書データ，体重・身長・血圧のようなバイタルデータもある．物理実験で得られる測定データも対象となりうるであろう．詳細に見ればさらにバリエーションは多い．実際，画像だけを取り上げても，さらに多様なものがある．顔画像，文字画像，航空写真や衛星画像，情景画像，監視画像，人物画像，物

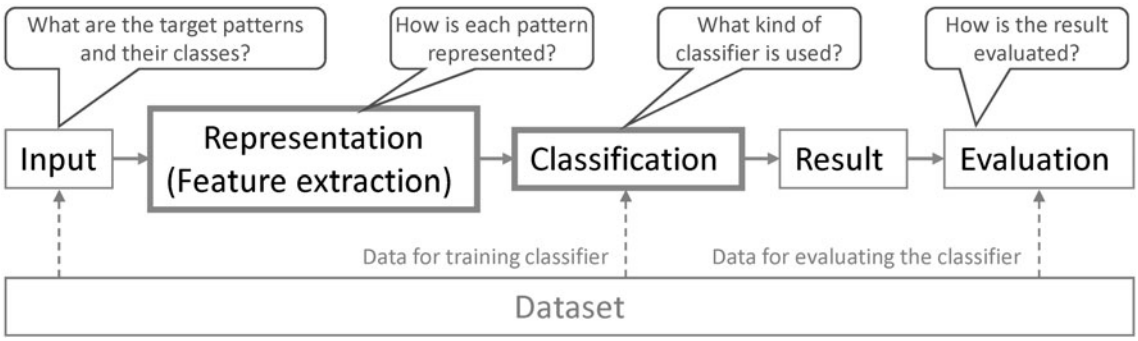

図 2 パターン認識の問題設定．

体画像（例えば商品画像），虹彩・指紋・掌静脈，そして動画像といった具合である．

3.2.2 クラスの設定 ～何に認識するか？

次に考えるべきことは，パターンを「何に認識するか」すなわち分類先としてのクラスの設定である．対象とするパターンとクラスが定まれば，まずはパターン認識問題の目的が定まることになる．

冒頭で述べたように，パターン認識ではC 個のクラスを「事前に」設定しておく必要がある．性別認識の場合，男と女の2クラスとなる．手書き数字認識の場合，“0”から“9”までの10クラスである．同じ文字認識でも漢字認識となると数千のクラスとなる．**図1**の例では，乗り物の種別がクラスとなっている．音声認識では，数十クラスの音素（およそ単一の発音記号に対応するレベル）が最初の認識対象である．ただしそれだけで終わることはなく，次に一連の音素の認識結果を組み合わせて単語を認識する．さらに一連の単語の認識結果を組み合わせて文を認識する場合もある．

以上のことからわかるように，パターン認識では事前に設定されたクラス以外の認識結果を与えることはできない．人間であれば，想定外の入力についても「なんとなく想像で」認識できるのとは対照的である．なお，「事前に想定されたどのクラスでもない」という$C+1$ 番目のクラスを設定することもある．これは棄却（リジェクト）と呼ばれる．無理にいずれかのクラスに認識してしまうよりは，「わからない」と答えてほしい場合には有効である．

問題によっては，クラスの設定そのものが自明でないこともある．例えば感情認識をする場合，我々の様々な感情を有限のクラスで表現する必要がある．マウスや線虫などの生物の行動様態を認識する場合も，妥当なクラスを事前に確定できないだろう．こうした場合は，次のようなボトムアップな方針でクラスを設定する方法が考えられる．まずパターン（データ）を大量に集める．次に，それらを似たものどうしを集めて幾つかのグループに分ける（いわゆるクラスタリング）．そして最後に各グループについて，そこに属するパターン群に適したクラス名をつければよい．グループ分けには，k-平均法[5]などのクラスタリング法が利用できる．ただしこの方針でも，いくつのグループに分けるかは自明ではないので，試行錯誤が必要になる場合もあるだろう．

やや特殊な形態として，1クラスだけを仮定する場合もある．異常認識はその例である．異常認識では，どんな異常が起こるかが事前にわからないことが多い．そこで，計算機には「正常なケース」だけを教えて認識できるようにしておき，正常と認識できなければ異常と判断するという対応が採られる．結果的に正常・異常の区別をしているので2クラス認識とも言えるが，その原理から1クラス認識と呼ばれることが多い．

もう一つの特殊な形態として，顔画像からの年齢認識や画像内の対象計数のように，クラスが数値・数量で表される状況がある．これは，入力パターン$\boldsymbol{x}$ に対して適切な数値 $y \in \mathbb{R}$ を返す関数 $y = h(\boldsymbol{x})$ を決定する問題である．従って，C 個の（すなわち離散有限個の）クラスに分類することを目的としたパターン認識というよりは，一種の回帰問題として扱うほうが自然であろう．

3.2.3 特徴抽出～パターンを表現する

ここまで「何を何に認識するか」という目的を定めたので，次は内側に入って「どのように認識するか」を考えることになる．この「内側」で最初に考えるべきことは，特徴抽出（feature extraction）と呼ばれる作業である．これはパターンをどう表現するかという問題である．簡単な作業に見えるかもしれないが，実は認識性能を左右する非常に重要な鍵となる．

3.2.3.1 特徴ベクトル

最も一般的なパターンの表現形態は，複数の数値の組すなわちベクトル表現である．これは「特徴ベクトル」と呼ばれる．例えば，風邪をひいているかいないかを認識する場合は，各人を（体温，咳の回数）の2次元ベクトルもしくは（体温，咳の回数，喉の赤さ）の3次元特徴ベクトルで表現することが考えられる．高血圧か否かを推定する場合なら，血圧値という1次元ベクトルすなわちスカラーで表現してもよいだろう．

画像のようなメディアデータは，高次元ベクトルとなることが多い．例えば100×100画素からなるグレースケール画像ならば，100×100＝1万個の画素値（各画素の明るさ）があるので，それらすべてを列挙したベクトル（ビットマップ特徴と呼ばれる）は1万次元となる．通常我々が生きる物理空間内の1点が高々数次元の低次元ベクトルで表現されるのに対し，パターン認識では非常に高次元なベクトルが現れることも珍しくない．

注意すべきは，多くの場合，次元の異なるパターンを同一の認識手法では扱えない点である．例えば50×50画像は2500次元ベクトルとなるので，100×100画像と一緒に扱うことはできない．したがって，大きさの異なる画像を扱う場合には，事前に拡大縮小処理を行って同一の大きさ（すなわち次元数）にしておく必要がある．

関連して，系列データ，特に時系列データからの特徴抽出も単純ではない．各時刻の特徴がD 次元のベクトルであり，系列長がT で一定ならば，全体を1つの $D \times T$ 次元ベクトルと見なせる．これをそのまま特徴ベクトルにはせず，文献[6]のように周波数解析を行い，低周波数のスペクトルを改めて特徴ベクトルとしてもよい．いずれにせよ，実際には，系列長T が一定しない場合のほうが多い点が問題になる．この場合，上記のサイズの違う画像を扱う場合と同様，長さが強制的にT となるように正規化する（もしくは打ち切る）方法が最も単純である．また系列の順序情報を放棄し，T 個の D 次元ベクトルの集合と見なし，T 個中どのようなベクトルが何回出現しているかをヒストグラム表現する方法もある．一方，後述する最近傍法の枠組みならば，系列長の異なるパターンであっても，それらの間に距離（もしくは類似度）さえ計算できれば識別可能であるので，特徴抽出の段階では次元を統一する必要はない．実際，音声認識やジェスチャ認識の多くはこの方法に依っている．

3.2.3.2 特徴空間

各パターンをD次元のベクトルで表現できれば，1つのパターンはD次元空間の1点と見ることができる．この空間のことを（D次元）「特徴空間」と呼ぶ．先の（体温，咳の回数）の特徴の場合，各人は2次元空間の1点になる．同様に，100×100画像ならば，**図3**上側に示すように，1万次元特徴空間の1点として表される．このような高次元空間に違和感を覚える読者も多いかもしれない．しかし，パターンの分布（3.2.3.3項）や識別手法を理解するうえで，非常に便利なので，是非慣れていただきたい．

3.2.3.3 パターンの分布

今，N個のパターンを準備したとすると，それらの特徴空間においてN個の点群となる．もしそれらが同一クラスのパターンの集合ならば，互いに類似した特徴を持つだろうから，ベクトルとしても類似したものとなり，結果的に特徴空間において密集した点群になるだろう．さらに異なるクラスのパターン集合を準備したなら，それらは先の点群から離れたところに，やはり密集した点群を成すであろう．

この密集の様子すなわち「パターンの分布」を把握できれば，認識精度の予測が可能になる．例えば，2つのクラスAとBを仮定し，各クラスのパターン分布が正規分布だったとする．その平均はそのクラスの最も平均的なパターンであり，それに類似したパターンが頻出することになる．分散はそのクラスに属するパターンの変動の大きさに相当する．もし2クラスの平均が近ければ，分布がオーバーラップしている可能性が高い．分散が大きければなおさらである．この分布がオーバーラップしている部分は，要するにクラスAでもBでもありうるわけで，区別がつかない部分であり，したがって原理的に正しく認識できない部分である．よって，このA，Bの2クラス認識問題は難しいということになる．逆に2分布が離れていれば容易であろう．

3.2.3.4 特徴が一意でないこと

ここで，「パターンの特徴表現は一意ではない」という事実を今一度確認しておきたい．先述の例では，風邪を認識するために（体温，咳の回数）“もしくは”（体温，咳の回数，喉の赤さ）とした．これ以外にも，体温のみを特徴としてもよいし，血糖値を加えてもよい．さらに，奇異に見えるかもしれないが，（体温，咳の回数，体温×咳の回数）や（体温，咳の回数×体温，咳の回数×体温×体温）でもよい[1]．このように「特徴はこれにすべし」という絶対的な決まりはない．実際，極端な例ではあるが，風邪を認識するために，身長を使っても，形式的には何の問題もない．

画像についても，多種多様な特徴表現が存在する．**図3**は画像の特徴表現の例である．上段は最も基本的なビットマップ特徴である．先述したように，100×100画像が1万次元の特徴ベクトルとして表現されている．いわゆるフィルタリング処理により，画像をぼかしたり，ノイズ除去を行ったり，逆にエッジ（明るさが急激に変化する個所）を強調したりすると，画像の大きさは変わらないために次元数は変わらず1万であるが，パターンの分布の様子は大きく変わることになる．したがってこのような画像加工も一種の特徴抽出と言える．さらに**図3**下段のように，抜本的に異なる表現形式もある．これは画素値ヒストグラム特徴であり，画像中に画素値（明るさ）iの画素の個数を列挙してできたベクトルである．グレースケール画像の多くは256段階の明るさで表現されているので，この特徴ベクトルは256次元となる．

このように特徴表現が変われば，特徴空間の次元が変わる．その変わり方は時に劇的ですらある．上記の画像の例では1万次元空間と256次元空間である．空間の大きさは次元数のべき乗なので，単に「約40倍」と言うだけでは済まない差異がある．

さらに重要なことは，特徴表現が変われば，パターンの

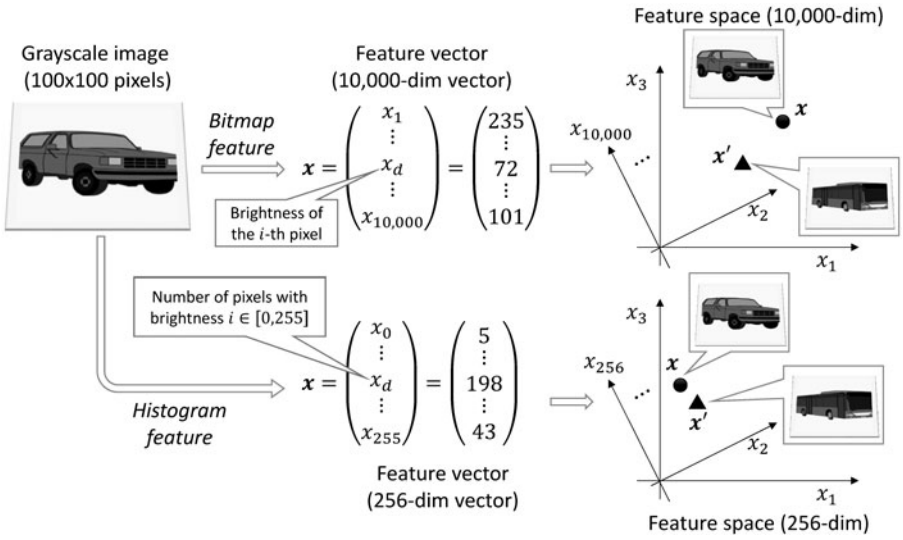

図3 特徴ベクトルと特徴空間．

1 この「最初からある特徴を組み合わせて別の（より高次元の）特徴を作る」という考え方は，パターン認識において重要なテクニックの一つである．形式的には，原特徴ベクトル$\boldsymbol{x}$に対しベクトル値関数$f(\boldsymbol{x})$を作用させていることになる．本稿では詳述できないが，パターン認識において「カーネル」という用語が出てくれば，このテクニックが暗に使われていることが多い．

分布も変わる点である．特に，分布が変われば，3.2.3.3で述べたように，オーバーラップの状況も変わる点に注意されたい．各人を身長という1次元特徴だけで表現したとき，風邪クラスと健康クラスの2クラスは大いにオーバーラップするだろう．一方，同じ1次元でも体温のほうがよりオーバーラップが少ないだろうし，（体温，咳の回数）の2次元にすればさらにオーバーラップは少ないだろう．オーバーラップの度合いはそのまま認識精度に影響する．以上より，特徴表現が認識性能を左右する非常に重要な鍵となることがわかる．

3.2.3.5 どのような特徴が望ましいのか？

パターン認識の困難さの一つは，各認識タスクについて，どのような特徴が最適なのか，自明ではないという点である．上に述べたように一意でないので，様々に考えうる特徴表現の中から，何とかして望ましいものを見出す必要がある．ここでどのような特徴が望ましいのであろうか？　一般には以下の3つの性質を満たすものである．

1．クラスが違えば大きく異なる．
2．同じクラスであれば，多少異なるパターンであっても類似する．
3．次元が低い．

第一の性質は自明であろう．風邪クラスの人と健康クラスの人で，身長はそう変わらないだろうが，体温は大きく変わるだろうから，後者のほうが望ましい．「分布の平均が各クラスで大きく異なる特徴」と言ってもよいだろう．その言い方を借りれば，第二の性質は，「各クラスの分布の分散が小さくなる特徴」となる．これは，同じクラスに属するならば，個々のパターン間の差異に鋭敏に反応することなく，なるべく似たような特徴になってほしい，という意味である．そうすれば結果的に各クラスはコンパクトな分布となり，オーバーラップは減るだろう．

困ったことに，この第一と第二の性質は矛盾した要請となっている．第一の性質を「パターンの差異に対して敏感」，第二の性質を「パターンの差異に対して不感」と言い直せば，その矛盾がより明確になる．ただし，全く絶望的というわけでもない．すなわち，前者は「クラス間の差異」，後者は「クラス内の差異」という違いがある．従って，我々が為すべきは，クラス内で起こりそうな差異には不感でクラス間の差異には敏感な特徴を見出すことになる．

第三の性質「次元が低い」は，要するに，限られたパターンでは，高次元空間内のパターン分布を把握できないことに起因した要請である[2]．先に認識精度を確保するためにはパターンの分布の把握が重要であると述べた．我々は無限にパターンを準備することができないので，有限のN個のパターンから，分布を推定する必要がある．ここで，パターンの次元DがNより大きければ，必ずN個のデータは超平面上に分布するように見えてしまう．（例えば2次元空間の2点は必ず直線上にあり，3次元空間の3点は必ず平面上にある．）したがって，真の分布が非常に複雑であっても，パターン数Nが少なければ，その真の分布を正しく把握できず，単に（D次元空間の線形部分空間である）超平面上にあるように誤解してしまう．

これは，いわゆる「次元の呪い[6]」と呼ばれる状況である．特徴空間の体積は，パターンの次元Dに対して指数関数的すなわち爆発的に増加する．この広漠なる空間に少数の点群しかなければ，分布が把握できず，その状態で認識規則を設定しても，十分な認識精度が得られない可能性が高い．先の（体温，咳の回数）と（体温，咳の回数，喉の赤さ）についても，パターン数が少なければ，前者のほうが高い認識精度を達成する可能性すらある．また，画像を1万次元で表すと，相当量のパターンがなければ，それらの分布を正しく把握できないことになる．

3.2.3.6 望ましい特徴をどう見出すのか？

ここまでの議論を受け，読者は，「結局，望ましい特徴をどうやって見出すのか？」と疑問に思うだろう．「特徴抽出に王道なし」という言葉があるように，残念ながら，この問題に対する絶対的な方法を存在しない．ただし，無策というわけでもない．具体的な対処法は次の通りである．

1．対象とするパターンの性質をよく見て，経験的に（すなわち職人技で）決める（ヒューリスティクス）
2．なるべく多くの特徴を一旦列挙しておいて，その中から有効そうなものだけを選択する（特徴選択）
3．なるべく多くの特徴を一旦列挙しておいて，それらから良さそうな特徴を合成する（低次元部分空間）
4．すべてを機械学習の枠組みに任せる．

紙数の都合上，各対処法について詳述はできないが，機械学習による特徴抽出の方法については，3.5章でその一例を述べる．特徴選択については，利用する特徴を試行錯誤しながら徐々に増やしていく方法（山登り法）が最も単純である．要するに加えて性能が上がる特徴を選択する方法である．より洗練された方法であるRandom Forestや決定木（"C4.5”が代表的な構成アルゴリズム）に関する解説を当たっていただきたい．低次元部分空間については，主成分分析（PCA）と強く関連する部分空間法[1]やFisherの線形判別法[1]が基本である．

3.2.4 パターンのクラスを決定する〜パターンの識別

特徴抽出が終われば，いよいよ「パターン$\boldsymbol{x}$がC個のクラスのうちのどの一つに属するか」を決定することになる．この処理を（狭義の）パターン認識と呼ぶこともあるが，本稿ではより明確にすべく「識別」（classification）と呼ぶ．この決定規則すなわち$\boldsymbol{x}\in\mathbb{R}^D$から$c\in[1,C]$への写像は，識別器（classifier）と呼ばれる．

図4は特徴空間による識別の様子である．このように特徴空間は，識別境界（classification boundary）と呼ばれるクラス間境界により分断される．入力$\boldsymbol{x}$のクラスは，分断された領域のいずれに入るかによって定まる．また同図に示しているように，識別器は学習パターン（後述するように「例題」）を用いることで学習すなわち設計される．そしてその識別器により識別境界が与えられることになる．

パターン認識研究の長い歴史の中で，実に様々な識別手

2　本誌記事[5]で触れられている「過学習」も本質的には同じ原因で生じている．

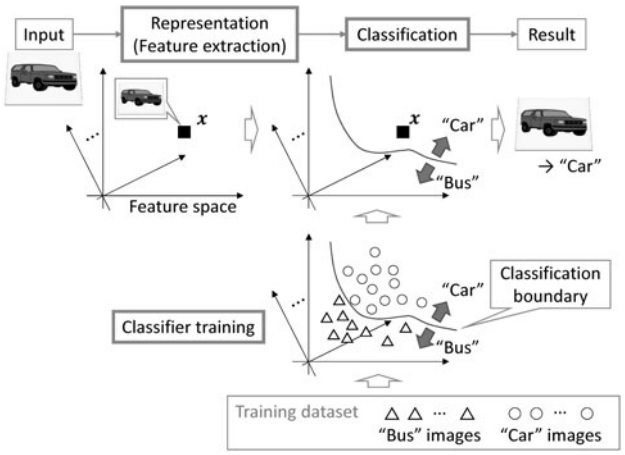

図4　識別器の学習と利用.

法が提案されてきた.「いろいろな識別手法があるようだが，自分の課題に対して一体どれを使えばいいのか？」―

実は各課題に対してどの識別手法を選ぶかは中々の難問である．準備可能な学習パターンの量，対象の分布の複雑さ，クラス数，実装の手間など，様々な要因によって選択すべき認識手法も変わってくる.

本稿では，互いに性質が大きく異なり，かつ代表的な次の3つの手法について解説する.

・最近傍法（3.3章）...長所：原理が非常に単純．学習パターン（後述）が多ければ，高い認識性能が見込める.
・サポートベクトルマシン（SVM）（3.4章）...長所：学習パターンが少ない場合に有効．識別に真に必要な学習パターンだけを自動選択するので，効率的.
・ディープニューラルネットワーク（DNN）（3.5章）...長所：学習パターンが十分あれば，非常に高い認識性能が得られる.

3.2.5　学習パターンとテストパターン

3.2.5.1　学習パターン

ある認識課題に対して，実際に特徴を設計したり識別器を設計したりするためには，何らかの「例題」が必要である．ここで例題とは，「このパターン $\boldsymbol{x}$ なら，このクラス c に認識してほしい」という，事前に正解クラスが分かっているパターンのことである．これを学習パターンと呼ぶ．手書き数字認識ならば，複数人に依頼して書いてもらった“0”から“9”の画像が学習パターンとなる．N 個の学習パターンの集合を $\{\boldsymbol{x}^1,\cdots,\boldsymbol{x}^n,\cdots,\boldsymbol{x}^N\}$ と表す[3].各学習パターンに正解が付随していることを明示したければ，$\{(\boldsymbol{x}^1,c^1),\cdots,(\boldsymbol{x}^n,c^n),\cdots,(\boldsymbol{x}^N,c^N)\}$ と表してもよい．これら学習パターンをできる限り正しく認識できるように（すなわち $\boldsymbol{x}^n$ の写像結果が c^n となるように），特徴抽出や識別器を構築することになる.

一般に，学習パターンは多ければ多いほど良い．先述の通り，学習パターンは「例題」であるから，なるべく見習うべきお手本が多いほうがよいわけである．その一方で，学習パターン数 N が数万，数十万となると，それぞれに正解クラスを付与するのも大変になるし，そもそも集めること自体が困難な場合も多い[4].

3.2.5.2　テストパターン

構築したパターン認識システムの性能を評価する場合，学習パターンとは別のパターン集合を準備する必要がある．これはテストパターンと呼ばれ，それらをどのぐらい正しく認識できるかによって，識別性能を評価する．このため，テストパターンについても，学習パターン同様，正解クラスがわかっている必要がある．また，正しい精度評価のためには，やはりなるべく多くのテストパターンが必要になる.

3.2.5.3　学習パターンとテストパターンの関係

正しい精度評価のためには，学習パターンとテストパターンを混ぜないほうがよい．すなわち，評価に用いるテストパターンを，識別器設計用の学習パターンとして利用してはならない．簡単に言えば，混ぜてしまうと「カンニング」になってしまう．例えば3.3章で述べる最近傍法の場合，学習パターンをそのままテストパターンとして用いると，すべてのテストパターンは100%精度で正しく認識できてしまう．これは決して歓迎すべき状態ではない．すなわち，本当に未知のパターンに対する評価になっていないので，現実的な状況での精度評価になっていない.

一方，学習パターンとテストパターンが大きく異なる分布を持っても，妥当な精度評価にはならない．例えば，A国で集めた顔画像を学習パターンとして顔認識システムを構築し，その精度評価をB国民でやると，精度は過少評価されるだろう．やはり同一の国に住む別の人の顔を使うべきであろう.

3.2.5.4　パターンが少ない場合の対処法

もし少数の学習パターンしか集まらない場合，高い識別性能を望むのは難しくなる．3.2.3.5項で述べた「次元の呪い」の問題も発生し，パターンの分布が把握できなくなるためである．とはいえ，物理や生物などサイエンスの分野では，実験コストもあり，パターンを大量に集めることが難しいケースも多いと思われる．もちろん，認識率が多少低くても客観的な評価としては十分な場合もあるだろう．しかし，やはり高い認識率が必要な場合もあるだろう.

パターンが少ない状況に対する万能薬は存在しないが，3.4章で述べるSVMは，少数学習データでも比較的良好な識別性能を与える手法である．また例えば手元にある学習パターンから人工的にパターンを生成することは，データ拡張（data augmentation）とも呼ばれ，工学の分野（特にDNNを使う際に）でよく行われる．画像認識の場合，学習パターンの画像を上下左右に反転させたり，数画素ずらしたり，回転させたりすると，学習パターンを相当数増

3　本稿では下添え字がベクトルの何番目の要素かを表すため，上添え字により何番目の学習パターンかを表す．べき乗と混乱しないよう注意されたい.

4　ただし最近ではCrowd Sourcingと呼ばれる方法で，大量の学習データが集められている．これは，インターネット上に正解クラスを付与してほしいパターンを置いておき，実際に付与してくれた人に謝礼等の報償を払うシステムである．Amazon Mechanical Turkがその代表格である.

やせる．

精度評価の際に，学習・テストパターンの両方を合わせても少数しかない場合もある．この場合，パターン集合の使い方を工夫することで，極力妥当な精度評価を行える[1]．例えば交叉検定法（cross-validation）では，まず全パターン集合を L 個の部分集合に分割する．そしてその部分集合のうちの1つを除いた $L-1$ 個分を学習パターン集合として識別器を構築し，その除いた1つをテストパターン集合として評価を行う．評価に用いる部分集合を変えながら，同様の評価を L 回繰り返し，得られたL個の評価値を平均することで，最終的な評価値と見なす．

3.2.6 評価指標

構築したパターン認識システムを性能評価する際，最も代表的な指標は「認識率」（recognition rate）である．要するに，すべてのテストパターンのうち，正しく識別されたものの割合である．

より詳細な評価が必要な場合，混同行列（confusion matrix）も便利である．例えば病気の診断の場合，健康な人を病気と誤認識した場合と，病人を健康と誤認識した場合とでは，同じ誤認識であっても後者のほうが深刻である．このように誤認識毎に重要度が異なる場合は，認識率という単一指標よりも詳細な評価が必要になる．混同行列は，クラス c を c' に誤認識した個数を c 行 c' 列の要素とした $C\times C$ 行列である．どのクラスに誤認識が多いかだけでなく，どのクラスがどのクラスに間違いやすいかもわかる．理想的な識別器ならば，混同行列が対角行列になる．

認識に要する時間すなわち計算量を評価指標とする場合もある．後述の最近傍認識の場合，一般にデータ数を増やせば増やすほど認識率は向上する．その一方で計算量が膨大になり，実用に供するのが難しくなる．

3.3 最近傍法によるパターン認識

3.3.1 最近傍法の原理

先に3.2.4節で「『わかりやすさ優先』ならば最近傍法」と述べたように，最近傍法は最も基本的かつ初歩的なパターン認識の原理である．いま N 個の学習パターン $\{\boldsymbol{x}^1,\cdots,\boldsymbol{x}^n,\cdots,\boldsymbol{x}^N\}$ があったとする．最近傍法とは，その名の通り，入力 $\boldsymbol{x}$ に最も近い学習パターン $\boldsymbol{x}^n$ のクラスを識別結果とする方法である．より形式的には，$\min_n \|\boldsymbol{x}-\boldsymbol{x}^n\|$ を与える $\boldsymbol{x}^n$ のクラスに識別する．ここで $\|\cdots\|$ はベクトルのノルムである．ユークリッドノルムであれば，$\|\boldsymbol{x}-\boldsymbol{x}^n\|$ は入力 $\boldsymbol{x}$ と学習パターン $\boldsymbol{x}^n$ のユークリッド距離である．このため，最近傍法は最短距離法とも呼ばれる．

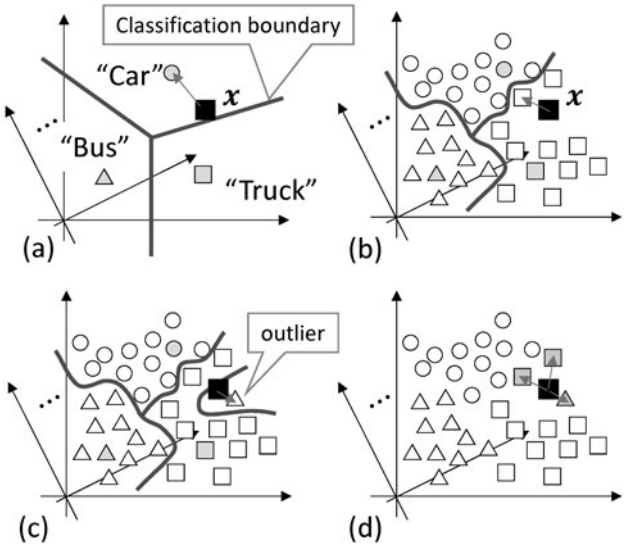

図5　最近傍法．(a)学習パターンが少数の場合．(b)多数の場合．(c)アウトライアの影響．(d)k－近傍法（$k=3$）．

図5 (a)は最近傍法による識別の様子である．ここでは，各クラスに1つの学習パターンが与えられた状況での，ユークリッドノルムによる識別を示している．同図で入力 $\boldsymbol{x}$ への最近傍は"Car"のクラスの学習パターンなので，入力 $\boldsymbol{x}$ は"Car"に識別される．このように，入力 $\boldsymbol{x}$ があるクラスに識別されたのは，最近傍の学習パターンがそのクラスに属していたことに因る．したがって同パターンを観察すれば識別の理由を容易かつ直感的に理解できる．この点は後述する他の手法にはない長所である．

最近傍法によるクラス間の識別境界は，同図(a)に示すように，隣接する学習パターン間の垂直二等分線（正確には超平面）で構成される．このことは，垂直二等分線がそれら学習パターンからの等距離点の集合であることから理解できる．全学習パターンとして見ると，学習パターンの位置を母点として特徴空間をボロノイ分割したことに相当する．

以上のように $\boldsymbol{x}$ と $\boldsymbol{x}^n$ の距離を測る代わりに，近さすなわち類似度を測る方法もある．この場合，内積 $\langle \boldsymbol{x},\boldsymbol{x}^n\rangle$ もしくはそれを正規化した $\langle \boldsymbol{x},\boldsymbol{x}^n\rangle/\|\boldsymbol{x}\|\,\|\boldsymbol{x}^n\|$ が利用される．後者は2つのベクトル $\boldsymbol{x}$ と $\boldsymbol{x}^n$ がなす角度 θ に関する $\cos\theta$ に等しいので，コサイン類似度とも呼ばれる．

3.3.2 最近傍法と学習データ数

もし学習パターンが十分にあり，かつ計算量が多くても構わない場合，最近傍法は単純ながら強力な認識手法になりうる．**図5** (b)は同図(a)から学習パターンが増えた状況を表している．1クラスあたり多数の学習パターンが与えられたことで，複雑で正確な識別境界が得られる．この図の例では，入力 $\boldsymbol{x}$ は"Truck"に識別されることになり，結局(a)の状況で"Car"に識別されたのは誤りだったことになる．

ただし，学習パターンが増えると，その中にアウトライア（outlier）が含まれるリスクも増える．アウトライアとは観測ノイズ等の影響で，本来のクラス分布から離れたところに（往々にして孤立的に）存在するパターンを指す．**図5** (c)は，アウトライアによる悪影響の様子である．このように入力 $\boldsymbol{x}$ はただ一つのアウトライアにより誤識別に転じる可能性がある．さらに悪いことに，同図では，アウトライアのクラスに誤識別される領域が右側に広がっていることも示している．

3.3.3 最近傍法の拡張

最近傍法は非常に単純な方法であるため，拡張性も高い．例えば，先のアウトライアについては，k 近傍法[1]に拡張することで解決できる．これは，最近傍だけでなく上

位 k 個の近傍パターンを求め，それらの多数決でクラスを決定する方法である．**図 5** (d)はその様子である．近傍数 $k=3$ とすることで，アウトライアの悪影響を回避できていることがわかる．k 近傍法は単純な拡張ながら非常に強力な方法である．ただし近傍数 k の設定については絶対的な指標はなく，試行錯誤的に定める必要がある．

他の拡張として，ユークリッド距離以外の距離の利用が考えられる．具体的には，絶対値距離（L_1 距離，マンハッタン距離とも呼ばれる）やマハラノビス距離[1]が挙げられる．後者は $(\boldsymbol{x}-\boldsymbol{x}^n)^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{x}^n)$ と表される．これは，$\boldsymbol{x}^n$ を平均とし，Σ を共分散行列とする（D 次元）正規分布を考えたとき，その分布の広がり方に比例した距離となっている．

3.2.3.1項で触れたように，距離を工夫することで，長さが一定しない系列パターンの認識にも利用できる．具体的には，編集距離（edit distance）もしくはほぼそれと等価なDPマッチング距離[7]を使えば，$\boldsymbol{x}$ と $\boldsymbol{x}^n$ の系列長が違ったとしても，両者間の距離を計算できるため，最近傍法が適用可能となる．

3.3.4 最近傍法の高速化

先述の通り，最近傍法による識別精度は，学習パターン数 N の影響を直接受ける．そして一般に N が大きいほど精度は良くなる．ただし，最近傍パターンを見つけるための計算量は N に比例するため，N が数百万といったオーダーになると，計算量を低減させて高速化を図る必要が生じる．

高速化の第一の方針は，学習パターンのうち，不要なパターンを除外する方法である．**図 5** (b)の識別境界を見ると，その構成に影響しているのはクラス境界付近のパターンだけであり，それ以外の影響はないことがわかる．すなわちクラス境界以外のパターンを除外しても識別境界は変わらず，結果的に識別性能も変わらない．実際には，高次元の特徴空間において，どの学習パターンがクラス境界にあるかは自明ではない．そこでCondensingやEditingと呼ばれる手法[8]が伝統的に使われている．

高速化の第二の方針では，真に最近傍のパターンを見つけなくてもよいと考える．近似最近傍探索（Approximate Nearest Neighbor，ANN）とも呼ばれ，k-d treeのような空間分割に基づく方法[8]が代表的である．他にもハッシュ関数に基づく方法（特にLocality Sensitive Hashing，LSH）がある[9]．

3.4 サポートベクトルマシン

3.4.1 サポートベクトルマシンの原理

SVMも非常によく用いられる識別器の一つである．その原理は，最近傍法と大きく異なる．具体的には，最近傍法では学習パターン $\boldsymbol{x}^n$ をそのまま使っていたのに対し，SVMでは学習パターンを使って「識別関数 $g(\boldsymbol{x})$」なるものをあらかじめ構成しておき，その識別関数 $g(\boldsymbol{x})$ のみを用いて識別を行う．具体的には，入力 $\boldsymbol{x}$ について $g(\boldsymbol{x})>0$ であればクラス1に，$g(\boldsymbol{x})<0$ であればクラス2に識別する．このように，SVMは基本的に2クラス識別（すなわち $C=2$）を対象としている．なお $g(\boldsymbol{x})=0$ がクラス間の境界となる．

SVMの最も簡単な場合は「線形SVM」と呼ばれ，$g(\boldsymbol{x})=\boldsymbol{w}^T\boldsymbol{x}+b$ と表される．要するに傾き $\boldsymbol{w}$，切片 b の（超）平面である．この場合，識別境界は $g(\boldsymbol{x})=\boldsymbol{w}^T\boldsymbol{x}+b=0$ であるから，やはり線形である．2次元特徴であれば，その2次元特徴空間（平面）を真っすぐな直線で分断することになり，3次元特徴であれば平面で分断することになる．要するに真っすぐな包丁で特徴空間をクラス1と2に分けているイメージである．

特徴ベクトル $\boldsymbol{x}$ が1次元の場合（例えば血圧だけから高血圧か否かを決める場合）はさらに簡単で，$g(\boldsymbol{x})=\boldsymbol{w}\boldsymbol{x}+b$ という直線となる．ここで傾き $\boldsymbol{w}$，切片 b の値は事前にわかっていないので，学習パターン $\{\boldsymbol{x}^1,\cdots,\boldsymbol{x}^n,\cdots,\boldsymbol{x}^N\}$ を使って設定する必要がある．その際の制約条件は，$\boldsymbol{x}^n$ がクラス1に属するのなら $g(\boldsymbol{x}^n)=\boldsymbol{w}\boldsymbol{x}^n+b\geq 1$ となり，クラス2なら $g(\boldsymbol{x}^n)=\boldsymbol{w}\boldsymbol{x}^n+b\leq -1$ とする[5]．（**図 6** では，△で表されたクラス1のパターンの箇所については，直線 $g(\boldsymbol{x})$ が1より上を通ってほしい，ということになる．）すべての学習パターン $\boldsymbol{x}^n$ についてこの条件を満たせば，先の「入力 $\boldsymbol{x}$ について $g(\boldsymbol{x})>0$ であればクラス1に，$g(\boldsymbol{x})<0$ であればクラス2に識別」が（少なくとも学習パターンについては）実現することになる．

SVMの原理の面白い点の一つは，以上の条件を満たす $\boldsymbol{w}$ と b の値を，最適化の枠組みで求めていることである．具体的には傾き $\boldsymbol{w}$ を最小化する問題を解く．この様子を**図 6** に示す．全部の学習パターンについて条件制約を満たす範囲で傾きを最小化する[6]と，最終的には図中の"Optimal solution"が識別関数として求まる．これ以上傾きを最小化しようとすると，制約条件が満たせなくなる点に注意されたい．

3.4.2 サポートベクトルマシンの長所

こうして得られた識別関数について重要な点は次の2つの性質である．

1．識別境界がクラス1と2の分布の間隙の中央を通過している．
2．全部で $N=8$ 個の学習パターンのうち，識別関数の構成に寄与しているのは，クラス境界に面している2個だけである．

この第一の性質こそがSVMの識別性能を高めている理由である．どちらかのクラスのパターン分布辺縁付近ぎりぎりではなく，両方のクラスからもできる限り遠いところに識別境界があることを意味する．学習パターンを分離識別するだけの意味であれば，識別境界が中央にある必要はない．しかし，認識システムの実際の使用状況を考えると，学習パターンよりもより曖昧すなわち他のクラスに混同さ

5 これら不等式制約において，≥ 1 や ≤ -1 の代わりに $\geq\varepsilon$ や $\leq -\varepsilon$ としても結果は変わらない．ここで ε は任意の正数である．

6 **図 6** の例における学習パターンは完全に2分割できるので，すべての制約条件を満たす解がある．しかし2クラスのパターンが境界付近時入り混じると，このような解は存在しなくなる．次節3.4.2においてこうした場合の対処法について述べる．

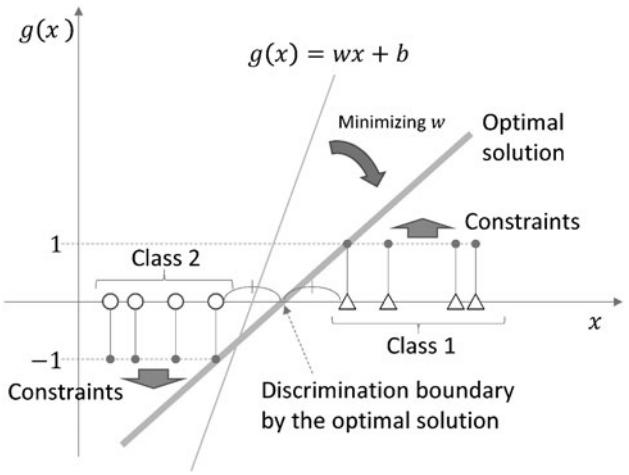

図6　SVMの原理.

れそうなパターンが入力されうる．識別境界を中央に置くことで，こうしたパターンが誤識別されるリスクを最小化できる．こうような（学習パターン以外の）入力に対しても高い識別性能を維持することを「汎化」(generalization)と呼ぶ.

第二の性質は，SVMの効率の良さを示している．3.3.4節でも述べたように，識別に重要なのはクラス分布の中央付近ではなく，むしろ境界付近の（曖昧な）パターンである．**図6**からもわかるように，SVMが自動的に選択し利用するのはまさにこうした境界付近の少数のパターンである[7]．識別関数がこの少数のパターンだけで構築されるということは，識別時にはその程度のパターンの情報だけが利用され，他のパターンは不要ということになる.

SVMは，以上の2つの性質により，比較的少数の学習パターンでも効率的にクラス間境界を捉える能力および汎化能力を併せ持っていることがわかる．この点が，少ない学習パターンでも高い識別精度を達成できるというSVMの強みを生んでいる.

3.4.3　サポートベクトルマシンの拡張

SVMについて，次の3つの拡張法についてその概要を述べる．第一にソフトマージン（完全に分離できない場合を許す），第二に非線形化（すなわち識別境界を非直線に），そして第三に多クラス化である．いずれも概念的な記述に留める．数式を含めた詳細については，専門書（例えば[2,10]）をご覧いただきたい．その際，本稿での直感的な議論が役に立つことを願う.

3.4.3.1　ソフトマージン

先述の通り，基本的なSVMでは，「$\boldsymbol{x}^n$がクラス1なら$g(\boldsymbol{x}^n)$は1以上，クラス2なら−1以下」を絶対的な制約条件として$\boldsymbol{w}$と$\boldsymbol{b}$を求めていた．しかし，**図6**のいずれかの○と）が入れ替わったとしたら，こうした制約条件を満たすことは不可能になる．したがってSVMは構築できないことになってしまう.

そこで開発されたのがソフトマージンである．要するに，「制約は絶対満たさなくてもいいが，もし満たせなかったら，満たせなかっただけペナルティを与える」という考え方である．一種の緩和であり，SVMとしては$\boldsymbol{w}$と$\boldsymbol{b}$の最適化に加えて，ペナルティの最小化も同時に解くことになる.

3.4.3.2　カーネルによる非線形化

3.4.1節で述べた線形SVMでは，識別境界も真っすぐになる．しかし分布が複雑に入り組んでいる場合は，真っすぐな境界で2クラスを正しく分断できるとは限らない．ソフトマージンで対処してもよいが，もう一つの考え方はSVMの非線形化である．包丁の比喩を用いるならば，線形SVMが真っすぐな包丁であったのに対し，非線形なSVMならば曲がった包丁で特徴空間を二分できる.

SVMの非線形化の方法は少々トリッキーである．基本となる考え方は，パターンの高次元化である．すなわち$\boldsymbol{x}$そのままではなく，それを高次元化した$\varphi(\boldsymbol{x})$を考える．例えば$\boldsymbol{x}=(x_1,x_2)^T$について$\varphi(\boldsymbol{x})=(x_1,x_2,x_1x_2)^T$と3次元化することを考える．これにより，元の2次元の特徴空間では線形SVMで識別できなかった場合でも，3次元の空間では線形識別できる可能性が上がる[8]．このように識別関数側を複雑にせずに，識別される側を高次元化により識別されやすくする．さらにSVMでは高次元化により増える計算量を抑えるために，カーネルトリックという巧妙な方法を利用している，

3.4.3.3　多クラス化

SVMは基本的に2クラス認識問題用であるため，多クラスを扱う場合には工夫が必要となる．最もよく用いられるのが1-vs-othersという方式である．具体的には，「クラスcとそれ以外」という2クラス用SVMをすべての$c\in C$について準備しておく．そして，入力$\boldsymbol{x}$をそれらC個のSVMで識別し，最も高い識別関数値を与えたクラスcに識別する．他には，総当たり型の1-vs-1方式や，最初から多クラスを前提として提案された「多クラスSVM」があるが，1-vs-othersが利用されることが多い.

3.5　ディープニューラルネットワーク

3.5.1　古くて新しいディープニューラルネットワーク

「ディープラーニング」「深層学習」などの呼称で，昨今では人工知能の代名詞とされているディープニューラルネットワーク（DNN）を最後に紹介したい．DNNはパターン認識にも利用できる．そして，DNNの一形態である畳み込みニューラルネットワーク（Convolutional neural networks，CNN）は，画像認識において従来を大きく上回る

7　特徴空間が高次元になると，境界を構成するパターンの割合は，一般に増加する．したがってこの利点は多少弱まることになるが，それでもできる限り境界を選択しようとするSVMの方針は有効である．なお境界の割合が多くなる理由は，「次元の呪い」の副次的効果として起こる「球面集中現象[1,6]」と呼ばれるもので，要するにすべてのパターン間が似たような距離になる状況である.

8　例えば$\boldsymbol{x}^1=(0,0)^T$と$\boldsymbol{x}^2=(1,1)^T$がクラス1，$\boldsymbol{x}^3=(1,0)^T$と$\boldsymbol{x}^4=(0,1)^T$がクラス2の場合，元の2次元空間では直線で二分できないが，このφによる3次元空間なら平面で二分可能となる.

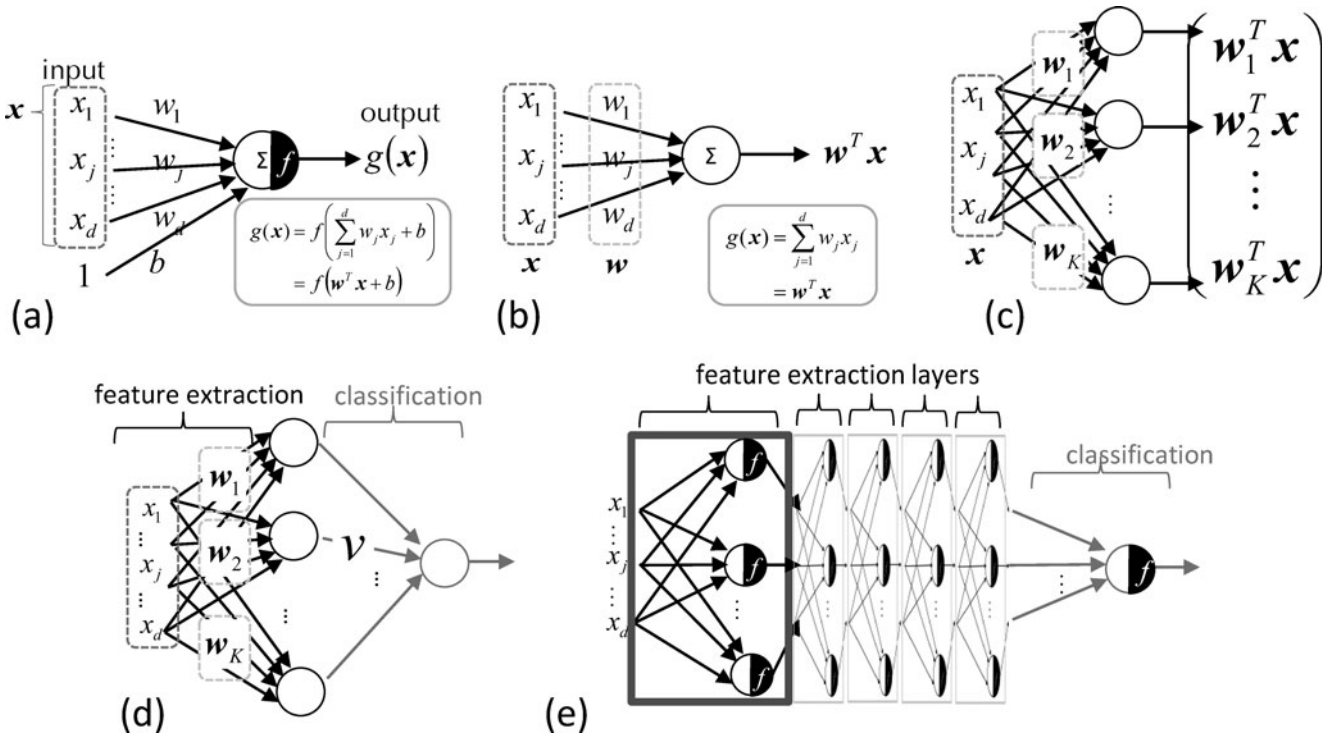

図7 DNNの概要.

性能を達成し，昨年（特定の認識課題ではあるが）人間の画像認識能力を超えたという報告もなされている．

実はDNNの考え方は古い．実際，CNNの原型であるネオコグニトロン[11]はすでに1979年に発表されている．それが再度注目されている理由は大きく二つある．第一は大量の学習パターンが入手可能になった点である．今や学習パターン数Nは時としてトータル数万～百万のオーダーで利用可能であり，それがDNNを構成する大量のパラメータの学習を可能にしている．第二は数々の細かな工夫である．例えば，従前はニューラルネットワークの構造を大規模化（正確には深層化）すると，学習が思ったように進まないという問題（Vanishing gradientと呼ばれる）があったが，最近のDNN/CNNでは解決が図られている，

3.5.2 ディープニューラルネットワークの基本的考え方

最近DNN/CNNについては解説書が数多く出版されている[12,13]．以下では，それら解説書よりもさらに平易な解説を試みる．

3.5.2.1 ニューロンのモデルと内積

DNNやCNNに限らず，およそニューラルネットワークと名がつく数理モデルは，**図7**(a)に示す「ニューロン」を組み合わせたものである．数式的には $g(\boldsymbol{x})=f(\boldsymbol{w}^T\boldsymbol{x}+b)$ であり，SVMのところ（3.4.1節）で述べた線形識別関数との違いは，わずかに非線形関数 f が最後に施されているか否かである．各ニューロンについて，その $\boldsymbol{w}$ と b を適切に定めることをニューラルネットの「学習」と呼ぶ．

ここで理解を容易にするために f と b を無視すると，**図7**(b)が得られる．これからニューロンの本質は $\boldsymbol{w}^T\boldsymbol{x}$ すなわち2ベクトル $\boldsymbol{x}$ と $\boldsymbol{w}$ の内積であることがわかる．以下に述べるように内積には二つの意味がある．この事実が，同形式のニューロンを複数組み合わせることで，パターン認識が実現することの原理を与えている．

3.5.2.2 内積の機能(1)：特徴抽出

内積 $\boldsymbol{w}^T\boldsymbol{x}$ は，2ベクトル $\boldsymbol{x}$ と $\boldsymbol{w}$ の方向が似ているほど内積値は大きくなる．要するに $\boldsymbol{x}$ と $\boldsymbol{w}$ の類似度である．ここでこの「$\boldsymbol{w}$ に対する類似度」が，$\boldsymbol{x}$ の特徴の一つとなりうる点は重要である[9]．3.2.3.1では，風邪かどうかを認識するために，各人を（体温，咳の回数）の2次元特徴で表す例を述べた．この体温も咳の回数も，その人を観察することで得られる特徴である．一方，その人が「患者Aや健常者Bに似ている」の程度も2次元特徴になりうる．前者を絶対的な特徴と呼べば，後者は相対的な特徴と言えよう．内積はまさにこの相対的な特徴である．ここで基準とする $\boldsymbol{w}$ を**図7**(c)のように K 個 $(\boldsymbol{w}_1,\cdots,\boldsymbol{w}_K)$ 考えた場合，それらとの K 個の内積値が得られ，それらは $\boldsymbol{x}$ を表現する K 次元ベクトルと考えることができる．

3.5.2.3 内積の機能(2)：識別関数

今一度，3.4.1節で述べた線形識別関数を見ると，$g(\boldsymbol{x})=\boldsymbol{w}^T\boldsymbol{x}+b$ の正負で $\boldsymbol{x}$ の識別結果が与えられるようになっていた．ここでも b を無視すると，ベクトル $\boldsymbol{x}$ と $\boldsymbol{w}$ の内積となっている．すなわち内積は，特徴抽出だけでなく，識別の機能も持っていることがわかる．

3.5.2.4 内積に次ぐ内積～多層ニューラルネットワーク

パターン認識の基本構成が特徴抽出→識別であったことを思い出せば，3.5.2.2項で特徴抽出したのちに，3.5.2.3

9 本文では非線形関数 f を無視したが，一般に f は，類似度の大きくなりすぎ，もしくは小さくなりすぎを抑制する機能を持つ．単純な機能であるが，これがなければ**図7**（d）は本当に内積すなわち線形演算だけからなるネットワークとなり，ニューロンが数多くあっても，結果的に単一のニューロン（内積）に帰着してしまう．

項で識別すれば，内積だけでパターン認識が完成することがわかる．**図 7** (d)はその全体像である．混乱を避けるために同図では識別時の内積対象を v で表している．この全体像，いわゆる多層ニューラルネットワークもしくは multi-layer perceptron と呼ばれるものである．1990年代に「ニューロ」ブームを起こしたのがこの形式である．

学習パターン $\boldsymbol{x}$ に対して正しい識別結果が得られるように $(\boldsymbol{w}_1,\cdots,\boldsymbol{w}_K)$ および $\boldsymbol{v}$ というパラメータ集合を「学習」する必要がある．本稿ではその詳細まで論じられないが，誤差逆伝搬法が一般に用いられる．これは一種の勾配法であり，識別誤りが少なくなる方向にこれらパラメータ集合を修正する方法である．勾配法に共通する弱点として局所解への収束があるが，ニューラルネットワークでも同様のリスクが存在する．

3.5.2.5 深層化のメリット

DNN とは，**図 7** (e)のように，前記の多層パーセプトロンの特徴抽出層を何重にも増やして深層化したものである．したがって，DNN といえども内積演算が基本である．では何重にもすることでどのような効果があるだろうか？先に述べたように，内積による特徴抽出では，元々 D 次元ベクトルの $\boldsymbol{x}$ を $(\boldsymbol{w}_1,\cdots,\boldsymbol{w}_K)$ との内積により K 次元ベクトルに変換していることに相当する．要するに D 次元空間を K 次元空間に（非線形）変換していることになる．空間が変換されれば分布が変わる．したがって深層化の目的は，直感的に言えば，何回も空間変換を施すことで混在して分離しにくいところを少しずつ「ほぐして」分離しやすくすることにある．もちろん，よく「ほぐれる」ように各層の $(\boldsymbol{w}_1,\cdots,\boldsymbol{w}_K)$ を決めてやる必要があり，その問題は先述のように誤差逆伝搬法で解かれる．

3.6 まとめ

本章では，パターン認識の基本的な流れと，その中心となる識別手法を 3 種類紹介した．パターン認識の具体的課題を扱う際，「まず何を考えなくてはならず，そして実際に解く際には多くの選択肢がある」ことを感じていただければ幸いである．その一方で，概念的な説明が多かったために，「自分の課題について，結局，どのような特徴抽出を行い，どのような識別方法を用いればよいか，よくわからない」というご意見もあろうかと思われる．残念ながら，パターン認識には万能薬がない．すなわち，3.2.3.6をはじめ，本章の随所で述べたように，課題の性質に応じた設計が必要になってくる．このため，その性質を見抜く経験も必要だろうし，さらに試行錯誤も必要になるであろう[10]．

本章では画像認識を例として多用したが，その前段階として必要となる画像情報処理については全く触れていない．もし画像情報処理についての概要が必要であれば，本章と同じように極力平易に書いた解説[14]をご覧いただければ幸いである．

本記事の執筆にあたり、応用力学研究所の稲垣教授に有益なコメントをいただきました．ここに感謝の意を表します．

参 考 文 献

［1］石井健一郎 他：わかりやすいパターン認識（オーム社，1998）．
［2］荒木雅弘：フリーソフトではじめる機械学習入門（森北出版，2014）．
［3］R.O. Duda *et al.*, *Pattern Classification* (2nd Edition), (Wiley-Interscience, 2000)（邦訳：パターン識別）（アドコム・メディア，2001）．
［4］C. Bishop, *Pattern Recognition and Machine Learning*, (Springer, 2006)（邦訳：パターン認識と機械学習，上・下）（丸善，2014））．
［5］前田新一：プラズマ・核融合学会誌 **92**, 334 (2016).
［6］宝珍輝尚 他：プラズマ・核融合学会誌 **92**, 347 (2016).
［7］内田誠一：電子情報通信学会技術研究報 rpPRMU2006-166 (2006).
［8］和田俊和：情報処理 **46**, 912 (2005).
［9］古賀久志：IEICE Fundamentals Review **7**, 256 (2013).
［10］竹内一郎，烏山昌幸：サポートベクトルマシン（講談社，2015）．
［11］福島邦彦：電子通信学会論文誌 J62-A, 658 (1979).
［12］岡谷貴之：深層学習（講談社，2015）．
［13］山下隆義：イラストで学ぶディープラーニング（講談社，2016）．
［14］S. Uchida, Development Growth and Differentiation **55**, Issue 4, 523 (2013).

内田誠一（うちだ せいいち）

1990 九大・工・電子卒．1992同大学院修士課程（情報）了．セコム株式会社勤務を経て，現在，同大学院システム情報科学研究院情報知能工学部門教授．博士（工学）．画像パターン・時系列パターンの解析・認識に関する研究に従事．最近はバイオイメージ・インフォマティクスなど分野を越えた協働研究を積極的に実施中．2007 IAPR/ICDAR Best Paper Award，2009電子情報通信学会論文賞，2016データサイエンスアワード他受賞．

10 必要あれば，著者までご遠慮なくご質問いただきたい．