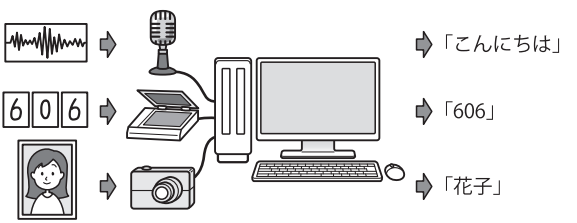
第1部

パターン認識の基礎



パターン認識とは文字を読んだり，音声を聞き分けたりする技術です．でもそれだけではありません．顔・指紋・目の虹彩などで個人を認証するバイオメトリクスもパターン認識の重要な応用分野ですし，実現が大いに期待されている自動車の自動運転を支える主要な技術でもあります．

このように便利なパターン認識技術ですが，いったいどのようにして実現されているのでしょうか．本書の第 1 部では，このパターン認識技術の基礎理論について学びます．

第1章

パターン認識って何？

1.1 パターン認識とは

パターン認識は英語の pattern recognition の訳語です．人間や動物が知覚できる実世界の画像・音・匂いなどの情報を**パターン** (pattern) といいます．recognition という単語は re（もう一度）と cognition（認めること）とに分けられます．入ってきた情報を，すでにもっている知識と照らし合わせて，「あっ，あれだ」と判断するという意味になります．すなわちパターン認識とは，目や耳で知覚したパターンを既知の概念（**クラス**）に対応させる処理のことです．

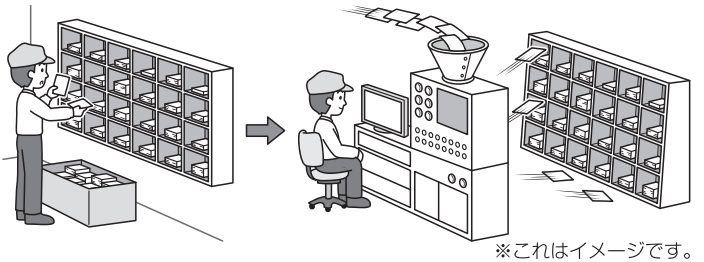
たとえば，人間は目から画像情報を得て，見ているものが何であるかがわかります．また，知っている人であれば誰であるかがわかります．耳から音を聞いて，自分が普段使っている言語であれば，何をいっているのかがわかります．玄関のドアを開けたら，ぷーんと匂いがしてきて，「今日はカレーだ」とわかったりします．この場合は，「キッチンからの匂い」という「パターン」を，「カレー」という「クラス」に識別しているのです．このような処理がパターン認識です．

一般に「パターン認識技術」というときは，この識別する主体はコンピュータです．人間が行っているパターン認識をコンピュータに代行させて，人間は楽をしようというわけです．

たとえば 1970 年頃の郵便物の仕分けは，葉書や封筒に書かれている郵便番号を人が読み取って，行き先ごとの棚に手で振り分けていました．いまでは手書き数字をパターン認識する機械が郵便番号を読み取って，自動的に振り分けてくれます（図 1.1(a)）．

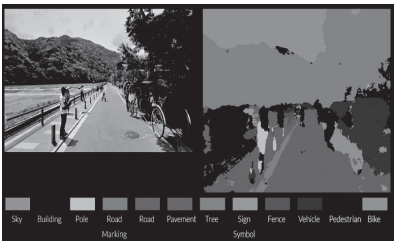
また，スマートフォンでは音声対話アプリが利用可能になり，アラーム設定・スケジュール登録などの操作や，乗換案内・天気予報などの情報取得が音声入力によってできるようになりました．また，いくつかの対話アプリが個性を競うようになり，気の利いた雑談機能も備えるようになってきています（図 1.1(b)）．

近年では，人間の代わりに車を運転してくれる自動運転技術の開発が盛んです．歩行者・他の車両・道路の情報などを認識して車を自動操縦することで，人間が楽にな



（a）郵便物の仕分けの変遷

（b）音声対話アプリ



（c）道路状況認識システム

図 1.1　さまざまなパターン認識システム

り，さらには人間が運転するよりも安全で事故が少なくなるような車社会の実現が目指されています（図 1.1(c)†）.

1.2 パターン認識システムの構成

パターン認識を行うプログラムは，認識の対象（音声・静止画像・動画像など）にかかわらず，一般に図 1.2 に示すようなモジュール構成で実現します.

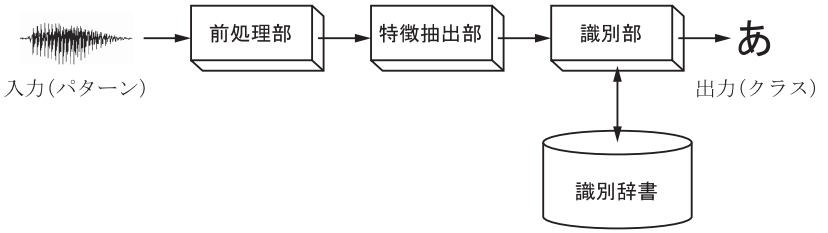


図 1.2　パターン認識システムの構成

† 道路状況認識システムは，ケンブリッジ大学 SegNet のデモンストレーションページでの実行例です．画像ファイルをアップロードすると，認識結果を表示してくれます（http://mi.eng.cam.ac.uk/projects/segnet/）.

前処理部は，入力されたパターンであるアナログ信号をディジタル信号に変換します．特徴抽出部は，そのディジタル信号から識別に役立つ特徴を取り出し，ベクトルの形式で出力します．識別部は受け取ったベクトルを識別辞書の内容と照らし合わせることによって，結果としてクラスを出力します．

以下では，各モジュールの役割と内部の処理を簡単に説明します．

1.3 前処理部

パターン認識の対象は実世界の信号です．音声は空気の疎密波であり，画像は 2 次元に広がった光の強度分布です．このような信号は連続的に変化するので，当然アナログ信号です．一方，コンピュータが処理できるのはディジタル信号です．したがって，パターン認識の最初の処理である**前処理**の役割は，アナログ信号をディジタル信号に変換することになります（図 1.3）．

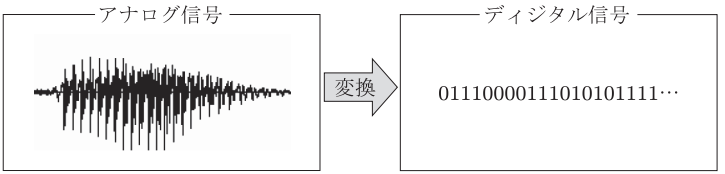


図 1.3 アナログ - ディジタル変換

アナログ信号は，コンピュータに接続された入力デバイスを通じて取り込みます．音声の場合は，マイクを使って空気の密度の変化を電気信号に変換します．画像の場合はカメラを使って画像素子が感知した光の強さを電気信号に変換します．

音声は，取り込むハードウェア（パソコンの場合はオーディオデバイス）の性能の上限まで細かな情報を取り込めます．また，画像はカメラの画素数の上限まで細かい情報を取り込めます．当然，アナログ信号はできるだけ忠実にディジタル化したほうが望ましいと考えられます．しかし，もとの情報を忠実に保持しようとすればするほど，表現に必要なデータ量は多くなります．データ量が多くなると，後段の特徴抽出処理の負担が大きくなり，認識のスピードが落ちてしまう可能性があります．後のことを考えると，ディジタル化した信号は，認識に必要な情報を保持しながら，できるだけ少ないデータ量で押さえるほうが望ましいわけです．

また，信号処理レベルで行えるノイズ除去など，後の特徴抽出処理を容易にする処理を前処理に含める場合もあります．

パターンの前処理に関しては，第 2 章で詳しく説明します．

1.4 特徴抽出部

特徴抽出とは，後段のパターンの識別[†1] に役に立つ情報を，入力されたデータから取り出す処理のことです．

これは逆にいうと，パターンの識別に役に立たない情報を捨てるということです．音声認識の場合は，話された音声がどの文字に対応しているかを識別する処理なので，誰が話しているのかということや，どのような大きさの声なのかということなどは，識別には関係しない情報として捨てることになります．文字認識の場合では，文字の位置・大きさ・色などが識別には関係のない情報です．枠の真ん中に書いても端っこに書いても，大きく書いても小さく書いても，黒で書いても赤で書いても，  （こちらは画像信号です）というパターンは「あ」（こちらは記号です）という文字です．このような識別に無関係な情報を**パターンの変動**とよんで，識別に用いる特徴と区別します．パターンの識別に用いる特徴は，パターンの変動に影響されにくい情報でなければなりません．

また，特徴抽出処理によって取り出す特徴は，何を識別対象のクラスにするかによって異なります．たとえば音声認識では，何をしゃべっているかということに関係のある特徴を，誰がしゃべっているかということに関係なく取り出さなければいけません．しかし，話者認識ではその逆です．誰がしゃべっているかということに関係のある特徴を，何をしゃべっているかに関係なく取り出さないといけないのです．同じ音声を対象にしていても，識別対象が異なれば，取り出すべき情報がまったく違うわけです．

一般に，一つの特徴だけで高精度な識別が行えることは，なかなかありません．たとえば，顔画像から目の部分だけを切り出した情報を用いて，それが誰であるかを見分けるのは難しいものです．人間でも人の顔を見分けるときは，髪型・顔の輪郭・肌の色などの複数の特徴を使っていると思われます．このような複数の特徴をまとめて表現する方法として，パターンの認識に使われる特徴は，一般に以下に示すような特徴ベクトルの形式で表現されます．

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^T \tag{1.1}$$

これは d 個の特徴の並びを表現した d 次元ベクトルです[†2]．この d 次元空間を**特徴**

†1 本書では，あるデータがどのクラスであるかを判定する処理を「識別」とよび，実世界のパターンをクラスに対応付ける処理を「認識」とよびます．すなわち「識別」とその前の何段階かの処理をまとめてた場合を「認識」とよぶこととします．

†2 ベクトル表記の肩に T とあるのは転置を意味します．特徴ベクトル $\boldsymbol{x}$ は列ベクトルで表現するのが一般的なのですが，スペースを節約するために行ベクトルで書いて転置の記号を付けています．

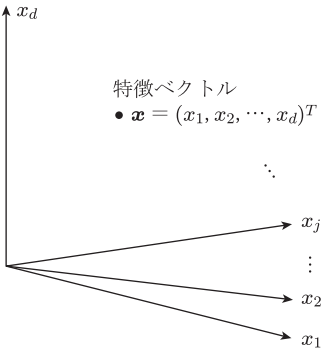


図 1.4 特徴空間と特徴ベクトル

空間とよび，$\boldsymbol{x}$ を**特徴ベクトル**とよびます．特徴ベクトルは特徴空間上の 1 点になります（図 1.4）．

この特徴ベクトルが特徴抽出部の出力です．特徴抽出処理については第 3 章で詳しく説明します．

1.5 識別部と識別辞書

パターン認識システムの最後の識別部はパターン認識処理の結果を出すところで，責任重大です．本書の第 1 部でも，説明の大半はこの識別部に関するものです．

1.5.1 基本的な識別手法

識別部は，入力された特徴ベクトルが，どのクラスに属するかを判定します．そのときに用いる情報が**識別辞書**です．識別辞書にどのような情報を格納するかに関しては，さまざまな方法があります．

もっとも基本的な方法としては，お手本となるベクトルを各クラス一つずつ格納しておき，識別したい入力をそのお手本と比較するという方法があります．このお手本となるベクトルを，**プロトタイプ**とよびます．

特徴ベクトルやプロトタイプは連続値を要素とするので，それらがぴったり一致することはあまりありません．なんらかの基準で「近い」ものを選びます．この選ばれたプロトタイプの属するクラスが，認識結果として出力されます．

まず，識別したいデータに対応する特徴ベクトルを $\boldsymbol{x}$ とします．これは特徴抽出部の出力で，式 (1.1) で示した d 次元ベクトルです．

次に，識別したいクラスが c 種類あるとして，それらをそれぞれ $\omega_1, \omega_2, \ldots, \omega_c$ と

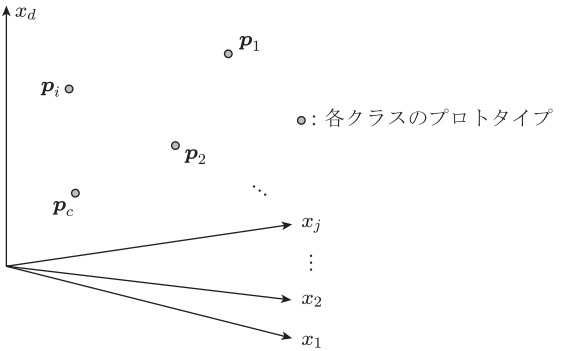


図 1.5　特徴空間上のプロトタイプ

表します．プロトタイプはそれぞれのクラスに対応して一つずつ用意される d 次元ベクトルで，それぞれ $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_c$ と表します（図 1.5）．

ここで，特徴ベクトル $\boldsymbol{x}$ がどのクラスに識別されるかを判定する方法として，$\boldsymbol{x}$ と各クラスのプロトタイプ $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_c$ との距離を測り，一番近いプロトタイプ $\boldsymbol{p}_i$ が属するクラス ω_i を正解とする方法が考えられます（図 1.6）．この方法を**最近傍決定則**（nearest neighbor 法，以後 **NN 法**）とよびます．NN 法の詳細については第 4 章で解説します．

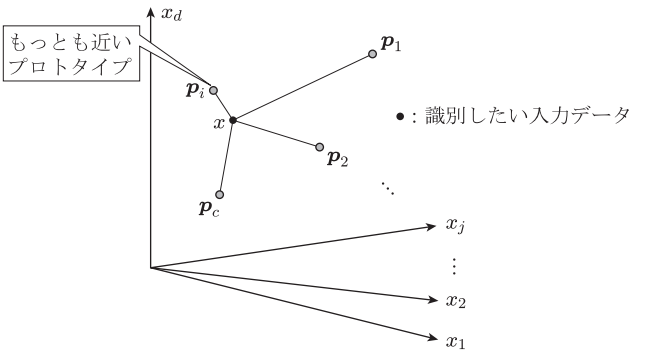


図 1.6　最近傍決定則

1.5.2　識別辞書の中身

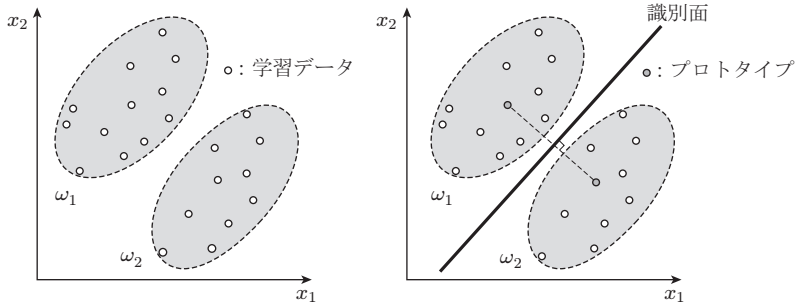
識別部が参照する識別辞書の中にはプロトタイプの情報が入っています．それでは，プロトタイプの位置はどうやって決めればよいのでしょうか．一般に，パターン認識では多くのサンプルを集めて，その情報からプロトタイプの位置を決めるという方法が用いられています．

たとえば，手書き数字認識の場合は，何人かに数字を書いてもらって，その特徴ベクトルと，それがどのクラスに属するかという情報（**正解クラスラベル**）を記録しておきます．このデータを使って識別部を賢くしてゆくので，このようなデータを**学習データ**とよびます．

特徴抽出部が識別に必要な特徴をうまく取り出せているとすれば，同じクラスに属する学習データは，書いた人のクセで多少ばらつきはあったとしても，特徴空間上でひとかたまりになっているはずです．このようなかたまりのことを**クラスタ**とよびます．それぞれのクラスに対応するクラスタの中から，プロトタイプとして代表的なものを一つずつ選ぶものとします．

ただし，適当にクラスタの真ん中あたりを選べばよいかというと，そうではありません．

例として，2 次元の特徴空間における 2 クラスの識別問題を考えてみましょう（図 1.7(a)）．NN 法は入力された特徴ベクトル $\boldsymbol{x}$ を，近いほうのプロトタイプが属するクラスに分類するので，これは，特徴ベクトル $\boldsymbol{x}$ がプロトタイプから等距離にある線，すなわち垂直二等分線[†1] のどちら側であるかを判定していることになります．したがって，プロトタイプの位置を決めるということは，クラス間の境界を決めるという問題に等しくなります（図 1.7(b)）．以後，この境界のことを**識別面**[†2] とよびます．



（a）**2 次元の 2 クラス識別問題**　　（b）**正しく設定された識別面**

図 1.7　プロトタイプと識別面の関係

図 (b) のようにプロトタイプの位置を決めることができれば，学習データをすべて正しく識別する識別面を求めることができます．しかし，プロトタイプの位置がまずければ，図 1.8 のように間違った識別面が設定されてしまうかもしれません．

---

†1 一般に d 次元では，2 点間の垂直二等分 $d-1$ 次元超平面になります．

†2 2 次元の特徴空間を考えている際には，識別「面」とよぶとわかりにくくなる場合があるので，決定境界とよばれることもあります．

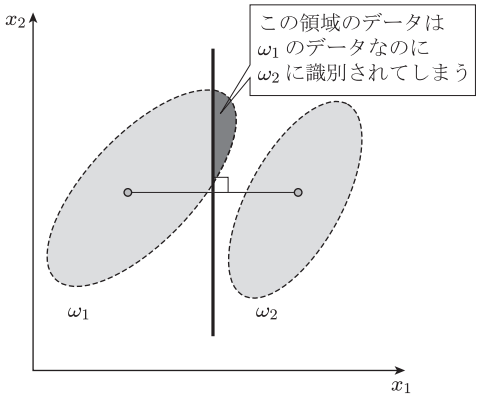


図 1.8 間違った識別面の例

すべての学習データをきれいにクラスごとに分けるように識別面を決めるにはどうすればよいのでしょうか.

実は，パターン認識でもっとも難しいのは，この識別面をどう決めるかということなのです．識別面は平面になるのか，またはぐにゃぐにゃした非線形曲面になるのか，そもそも識別面が決められるのか（クラスが重なっていないか）など，さまざまな場合を考えなければなりません．この識別面を決めるために，学習データを利用します．一般に，学習データが多ければ多いほど識別面は信用できるものになります．しかし，識別するクラスがどのように特徴空間に分布しているのかという情報を使い，そしてその分布に適した学習方法を選ばなければ，いくらデータがたくさんあってもうまくゆかないこともあります.

例題 1.1 図 1.9(a) に示す 25 次元ベクトル（要素が 0（白），1（黒）からなる縦 5 マス×横 5 マス = 25 次元）をプロトタイプとして，図 1.9(b) に示す入力パターンがどのクラスに識別されるかを，最近傍決定則（NN 法）を用いて求めよ．また，この結果が直観と反する場合，なぜそのような結果になったかを考察せよ.

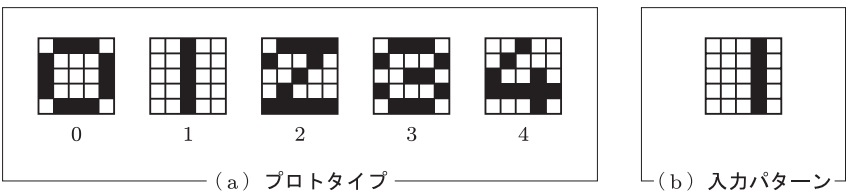


図 1.9 数字認識用のデータ

▷解答例 ここでは 0 から 4 の数字認識を行う場合を考えます.

本来は,図 (a) に並んでいる 0 から 4 のパターンに対して特徴抽出処理を行ってプロトタイプを求めなければならないのですが,ここではその処理を省略して,前処理部からの出力そのものをプロトタイプとします.すなわち,この例ではプロトタイプ $\boldsymbol{p}_i$ $(i = 0, 1, 2, 3, 4)$ は 25 次元のベクトル(各次元の要素は 0 または 1)になります.

たとえば,$\boldsymbol{p}_0$ は次のようになります.

$$\boldsymbol{p}_0 = (0,1,1,1,0,1,0,0,0,1,1,0,0,0,1,1,0,0,0,1,0,1,1,1,0)^T$$

この前提で,図 (b) の入力パターン $\boldsymbol{x}$ がどのクラスに識別されるかを NN 法で求めてみましょう.入力パターンはどう見ても「1」に見えます.さて,これが正しく識別されるでしょうか.

ベクトル $\boldsymbol{x}$ と $\boldsymbol{p}_i$ との距離 $D(\boldsymbol{x}, \boldsymbol{p}_i)$ は以下の式で求められます.

$$D(\boldsymbol{x}, \boldsymbol{p}_i) = \sqrt{(x_1 - p_{i1})^2 + (x_2 - p_{i2})^2 + \cdots + (x_{25} - p_{i25})^2}$$

ここで各次元の要素の差の 2 乗は,0 または 1 となります.したがって,距離最小のプロトタイプは,入力パターンと異なるマス目の数が最小のものだということになります.

入力パターンと異なるマス目を数えたものを以下の表 1.1 に示します.

表 1.1

クラス	0	1	2	3	4
異なるマス目の数	13	10	12	11	9

したがって,入力パターンは「4」と識別されます.

これは明らかに直観に反します.なぜ,こんな結果が出たのでしょう.NN 法は役に立たないのでしょうか.

この原因は,NN 法にあるのではなく,特徴抽出処理を省略したことにあります.この場合は特徴抽出処理として,位置の変動や大きさの変動に対してあまり変化しない量を計算して,それを特徴ベクトルにするべきだったのです.特徴抽出後に NN 法を適用する手順は,本章の演習問題を参照してください.

ここまででパターン認識処理の概要はつかめたでしょうか.ずいぶん簡単だと思われたかもしれません.しかし,ここではうまくゆく場合だけを単純化して説明しています.現実のデータを対象にして実際にパターン認識プログラムを作成すると,さまざまな「うまくゆかない場合」に遭遇します.その困難を偉大な先人研究者たちはどのようにして乗り越えてきたのかを以後の章で説明します.お楽しみに.

演習問題

1.1 図 1.9(a) のプロトタイプから，縦・横・斜めの線の数およびループの数を特徴として抽出せよ．ただし，縦・横・斜めの線とは，それぞれの方向に黒のマスが三つ以上続いた場合を数える．また，ループは縦・横・斜めのいずれかで黒のマスが途切れずに輪になっているものを数える．

1.2 演習問題 1.1 で抽出した特徴ベクトルを新たなプロトタイプとして，例題 1.1 の入力パターンを識別せよ．