

## 1

# パターン認識って何？

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といいます。パターン認識とは、このようなパターンをあらかじめ定められた複数の概念（クラス）のうちのひとつ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。コンピュータに人間が行っているパターン認識を代行させて、人間は楽をしようというわけです。

たとえば、昔（1970年頃）は葉書に書かれている郵便番号は、人が読み取って行き先ごとに別々の棚に振り分けていたのです。今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）。

また、電話で航空機のチケットや電車の指定券を予約することができる音声対話システムが実用化されています（図 1.2）。録音された音声案内に従って電話の数字キーで情報を入力するシステムは以前からありました。しかし、携帯電話では電話器を耳からいったん離して入力しなければならないことや、ガイダンスが不明確なときには自分がしたいことに対応すると思われる数字をガイダンスが終わるまで憶えていなければならないことなどの問題点があり、あまり使い勝手のよいものではありません。

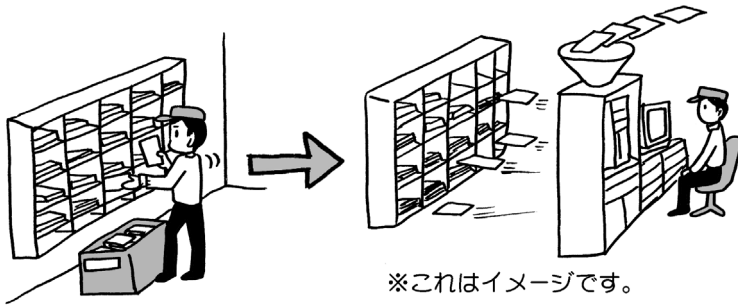


図 1.1 郵便物の仕分けの変遷



図 1.2 音声対話システム

した<sup>\*1</sup>。一方、音声認識に対応したシステムでは希望の日時や席種などを声によって入力できます。また、家電製品の電話サポートなどでは、最初に製品コードを音声認識することによって、迅速に担当部署に振り分けるシステムがあります。

この第1章ではパターン認識システムがどのようにして作られているのかということと、そこで用いられている基本的な方法について解説します。

## 1.1 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）に関わらず、一般に図 1.3 に示すようなモジュール構成で実現します。

前処理部には、コンピュータにつないだカメラやマイクから認識対象の信号が入力

<sup>\*1</sup> 「…は1と#を, …は2と#を, …は3と#を, オペレータとお話になりたい場合は8と#を, 終了する場合は9と#を押して下さい」というようなガイダンスです。自分の目的が案内されたメニューに該当するかどうか確信がないときは、これかなと思う番号を憶えておきながらガイダンスを最後まで聞かなくてはなりません。「これまでの選択肢を全部聞きたい場合は49と#を, 何のために電話をかけたか忘れてしまった場合は50と#を…」というジョークまであります。

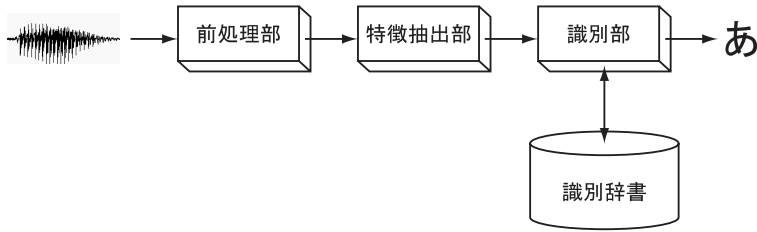


図 1.3 パターン認識システムの構成

されます。カメラやマイクなどの入力装置から入力された実世界のアナログ信号は、サウンドボードやキャプチャボードによって、コンピュータ内部で処理可能なデジタル信号に変換されます。ここでは、このようなデジタル化と、後の特徴抽出処理を容易にする処理を含めて前処理とよびます。

特徴抽出部は、前処理部の出力であるデジタル化されたデータを入力し、パターンの識別<sup>\*2</sup>に役立つ情報を取り出します。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ことです。文字の場合では、文字の色や大きさなどは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、また、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、あ（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。音声の場合では、誰が話しているのかということや声の大きさなどは、話された音声がどの文字に対応しているかという識別には関係のない情報です。特徴抽出処理では、このようなパターンの変動に影響されない情報で、かつ識別に役立つ情報をいくつか、通常はベクトルの形式で抽出します。これを特徴ベクトルといいます。

識別部では、この特徴ベクトルを識別辞書中の各クラスのお手本ベクトルと比較して識別結果を決めます。識別辞書中には「あ」に対応するお手本ベクトル、「い」に対応するお手本ベクトル、…といったものが格納されていると考えて下さい。通常、特徴ベクトル、お手本ベクトルともに実数値を要素としますから、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」ものを選ぶわけです。この選ばれたお手本ベクトルの属するクラスが、認識結果として出力されます。

以下では、もう少し詳しく各モジュールの役割を説明します。

<sup>\*2</sup> 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た場合を「認識」とよぶことにします。

## 1.2 パターンの前処理

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ジタル信号です。したがって、パターン認識の最初の処理はアナログ信号をデジタル信号に変換することです(図1.4)。

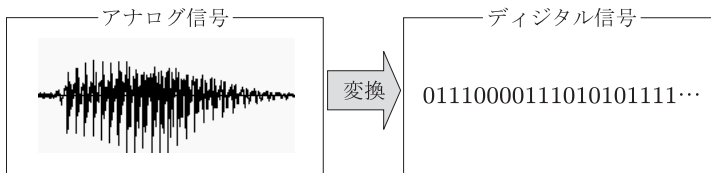


図 1.4 アナログ→デジタル変換

アナログ信号はコンピュータに接続された入出力デバイスを通じて取り込みます。音声の場合はマイクロフォンを使って空気の疎密波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。

音声は取り込みのハードウェア（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みます。また、画像の場合はカメラの画素数の上限まで細かい情報を取り込みます。当然、アナログ信号はできるだけ忠実にデジタル化した方が望ましいと考えられます。しかし、元の情報を忠実に再現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、これ以降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、デジタル化した信号は、認識に必要な情報は落ちていないという前提で、できるだけ小さくするほうが望ましいわけです。

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。

## 1.3 特徴抽出とは

特徴抽出とは、入力されたデータからパターンの認識に役に立つ情報を取り出す処理のことです。必然的にこの処理は、入力パターンが何であるか（画像なのか、音なのか）、どのようなクラスに分類するのか（顔、文字、単語など）によって異なります。具体的な事例は第3章で説明するので、ここでは一般的な考え方を述べます。

1.1節で説明したように、パターンを識別するための特徴は、そのパターンの変動に

## 6 第1章 パターン認識って何？

影響されにくい情報でなければなりません。これは何を識別対象のクラスにするかに依存します。たとえば音声認識だと、何をしゃべっているか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、話者認識だとその逆です。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も、取り出すべき情報がまったく違うわけです。

また、ひとつの特徴ですべてのクラスの分類がうまくいくものはなかなかありません。たとえば、目だけを切り出した情報から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・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。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。

$$\mathbf{x} = (x_1, x_2, \dots, x_d)^t \quad (1.1)$$

これは  $d$  個の特徴を表現した  $d$  次元ベクトルです\*3。この  $d$  次元空間を特徴空間とよび、 $\mathbf{x}$  を特徴ベクトルとよびます。この特徴ベクトルは特徴空間上の1点になります (図 1.5)。

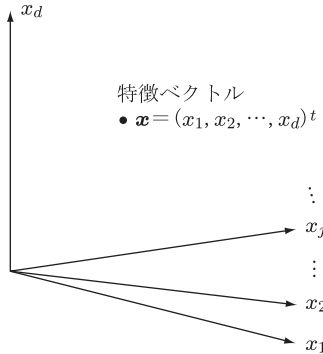


図 1.5 特徴空間と特徴ベクトル

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

\*3 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  $\mathbf{x}$  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。

## 1.4 識別部の役割

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、非常に責任重大です。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。

### 1.4.1 識別部の構成

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別します。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、さまざまな手法があります。

最も簡単な方法としては、各クラスのお手本となるベクトルを格納しておく方法があります。このお手本となるベクトルを以後、**プロトタイプ**とよびます。ここで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 $\omega_1, \omega_2, \dots, \omega_c$ と表します。そうすると、プロトタイプはそれぞれのクラスに対応してひとつずつ用意され、それぞれ $p_1, p_2, \dots, p_c$ となります（図1.6）。

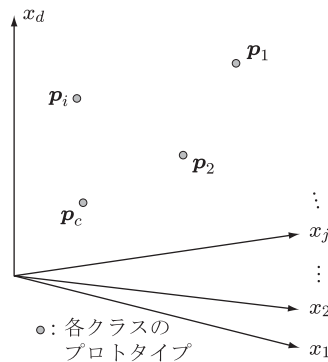


図 1.6 特徴空間上のプロトタイプ

次に、識別したいデータに対応する特徴ベクトルを $\mathbf{x}$ とします。これは特徴抽出部の出力です。この $\mathbf{x}$ がどのクラスに識別されるかを判定する際に、 $\mathbf{x}$ と各クラスのプロトタイプとの距離を測り、いちばん近いプロトタイプ $p_i$ の属するクラス $\omega_i$ を正解とする方法が考えられます（図1.7）。この方法を**最近傍決定則**（Nearest Neighbor 法、略して**NN 法**）といいます。

この方法について、詳しくは4.1節で解説します。

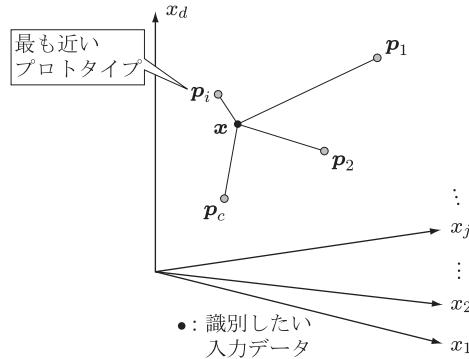


図 1.7 最近傍決定則

### 1.4.2 プロトタイプの決め方

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ですと、何人かの人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（通常はクラス番号で、これを正解クラスラベルとよびます）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いいます。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、（特徴抽出部がうまく特徴を取り出せているとすれば）特徴空間上でひとかたまりになっているはずです。このかたまりの中から、プロトタイプとして代表的なものを各クラスにつきひとつ選ぶものとします。

ただし、適当にかたまり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。

2クラスの識別問題を考えてみましょう。NN法は入力された特徴ベクトル  $\mathbf{x}$  を、近いほうのプロトタイプが属するクラスに分類するわけですから、特徴ベクトル  $\mathbf{x}$  が、プロトタイプから等距離にある線または平面\*4のどちら側であるかを判定することになります。したがって、プロトタイプの位置を決めるということは、それぞれのクラス間の境界面を決めるという問題に等しくなります。プロトタイプの位置がまずければ、図 1.8 のように間違った境界面（この場合は境界線）が引かれてしまうかもしれません。

\*4 一般に  $d$  次元では、2点間の垂直2等分  $d-1$  次元超平面になります。

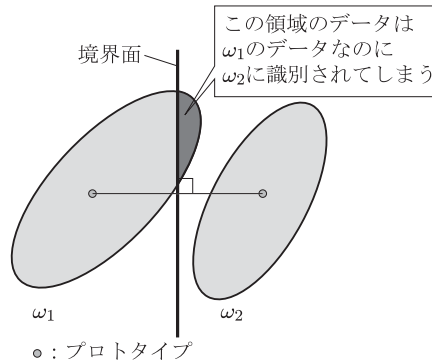


図 1.8 間違った境界面の例

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境界面を決めるにはどうすればよいのでしょうか。

実は、パターン認識で最も難しいのは、この境界面をどう決めるかということなのです。境界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、そもそも境界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。この境界面を決めるために、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境界面は信用できるものになります。いろいろな事例をたくさん教えれば賢くなるというあたりまえのことですね。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いかないこともあります。

ここまででパターン認識処理の概要は掴めたでしょうか。ずいぶん簡単だと思われたかもしれません。しかし、ここではうまくい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「うまくいかない場合」に遭遇します。その困難を過去の偉大な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。

### 例題 1.1

- 図 1.9 左側に示す 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）をプロトタイプとして、右側に示す入力パターンがどのクラスに識別されるかを NN 法を用いて求めよ。
1. 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。



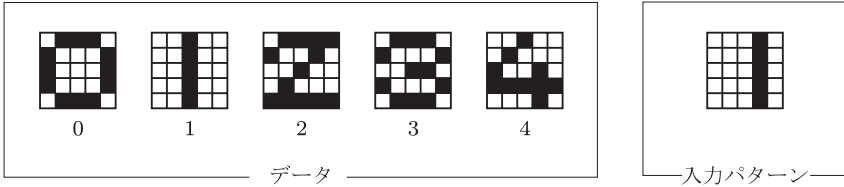


図 1.9 数字認識の例

### ▶ 解答例

1. 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、左側に並んでいる 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  $p_i (i = 0, 1, 2, 3, 4)$  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、 $p_0$  は次のようになります。

$$p_0 = (0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0) \quad (1.2)$$

この前提で、入力パターン  $x$  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  $x$  と  $p_i$  との距離は以下の式で求められます。

$$D(x, p_i) = \sqrt{(x_1 - p_{i1})^2 + (x_2 - p_{i2})^2 + \cdots + (x_{25} - p_{i25})^2} \quad (1.3)$$

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、距離最小のベクトルは、入力パターンと異なるマス目の数が最小のプロトタイプ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に示します。

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。

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

この原因は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（たとえば縦・横の直線の本数など）を計算して、それを特徴ベクトルにすべきだったのです。

**演習問題**

- 1-1 図 1.9 のデータから，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．