パターン認識実験

2024 年 9 月 24 日版

実験の概要

パターン認識とは、実世界の画像・音声などの信号を、あらかじめ定められた複数の概念 (クラス) のうちの一つに対応させる処理のことです。パターン認識実験では、文字認識を題材として、パターン認識の基本的な手順の習得と、背景にある統計的識別の理論を理解することを目的とします。

表 1　各回の内容

週	提出物	実験内容
1		特徴抽出
2		特徴の評価
3	レポート (1,2 週分)	数字識別
4		識別性能の評価
	レポート (3,4 週分)	

パターン認識概論

パターン認識システムは、通常図 1 に示すようなモジュールで構成されます。

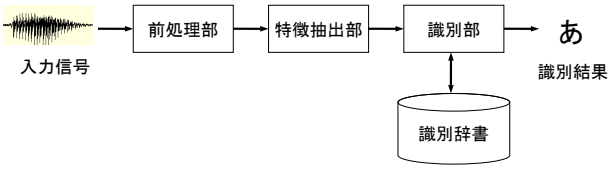

図 1　パターン認識システムの構成

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ィジタル信号に変換されます。ここでは、このようなディジタル化と、後の特徴抽出処理を容易にするノイズ除去などの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ィジタル化されたデータを入力し、パターンの識別[*1]に役立つ情報を取り出します。これは逆に言うと、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。特徴抽出処理では、パターンの変動（識別に直接影響のない特徴の変化）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**に格納されている各クラスについての情報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場合もありますし、特徴ベクトルを入力として、それが「あ」である確率、「い」である確率、... を計算する関数が入っている場合もあります。後者のような確率を求める関数を用いる方法を統計的識別手法とよび、その確率値が最も高いクラスが、認識結果として出力されます。

識別辞書の内容をデータから自動的に決定することは**機械学習**とよばれます。いくつかの入力事例とそれらに対して正解を付けたデータの集合を**学習データ**とよび、この学習データから機械学習を行って識別器を構成することが、パターン認識技術の中心的な課題となります。

1　特徴抽出（第 1 週）

第 1 週は数字画像を認識するにあたり、画像から特徴を抽出する実験を行います。図 1 の特徴抽出部の実装に関する実験です。

1.1　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識別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。

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影響されにくい情報でなければなりません。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。枠の真ん中に書こうが端っこに書こうが、大きく書こうが小さく書こうが、黒で書こうが赤で書こうが、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、

[*1] 本稿では、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」とよぶことにします。

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。

また、1 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。例えば、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ことです。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。

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^T \tag{1}$$

これは d 個の特徴を表現した d 次元ベクトルです。この d 次元空間を**特徴空間**とよび、$\boldsymbol{x}$ 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[*2](図 2)。

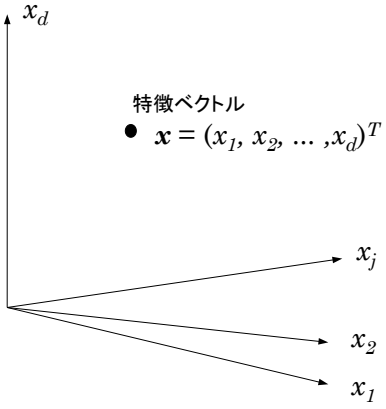

図 2　特徴空間と特徴ベクトル

この特徴ベクトルが特徴抽出部の出力になります。

1.2　特徴抽出の手順

ここでは、文字認識を題材として特徴抽出の実装を行ってみましょう。

実験で用いる数字画像の例を図 3 に示します。字体は限定されていますが、撮影状態の異なる 10 種類の画像が各数字（0〜9）毎に 10 枚用意されています。また、それぞれの画像には白色雑音やバイナリー雑音も加わっているので、撮影状態に影響されにくい特徴量の選択が重要となります。本実験では、各数字の特徴量を計算し特徴ベクトルを求めます。

入力となる PGM (Portable Graymap Format) 画像ファイルは、白の背景に黒で数字が描かれています。最も黒い画素の画素濃度が 0 で最も白い画素の画素濃度が 255 です。識別においては黒の数字が画像内でどのように分布しているのかが重要なので、関心のある黒の画素値を大きく、関心のない背景の画素値を小さくするようにします。したがって、特徴量の計算では、以下のように画素値を反転させた画像 $\tilde{I}_{x,y}$ を用います。ここで、$I_{x,y}$ は対象となる画像の座標 (x, y) での画素値です。

$$\tilde{I}_{x,y} = 255 - I_{x,y} \tag{2}$$

次に、画像毎の黒画素数の違いが、特徴のスケールに影響しないように、1 枚の画像に関して画素値の総和が 1 になるように画素濃度正規化処理を行います。

$$\hat{I}_{x,y} = \frac{\tilde{I}_{x,y}}{\sum_{x=1}^{X}\sum_{y=1}^{Y}\tilde{I}_{x,y}} \tag{3}$$

この $\hat{I}_{x,y}$ を用いて、画像の重心・分散・ゆがみ・扁平度を求めます。ここで、w, h はそれぞれ画像の横方向、縦方向の画素数を表します。また、添え字の X, Y はそれぞれ横方向、縦方向を示しており、たとえば μ_X は横方向の重心、μ_Y は縦方向の重心を表します。

重心 (μ)

$$\begin{aligned}\mu_X &= \sum_{x=1}^{w}\sum_{y=1}^{h} x\hat{I}_{x,y} \\ \mu_Y &= \sum_{x=1}^{w}\sum_{y=1}^{h} y\hat{I}_{x,y}\end{aligned} \tag{4}$$

分散 (σ^2)

$$\begin{aligned}\sigma_X^2 &= \sum_{x=1}^{w}\sum_{y=1}^{h}(x-\mu_X)^2\hat{I}_{x,y} \\ \sigma_Y^2 &= \sum_{x=1}^{w}\sum_{y=1}^{h}(y-\mu_Y)^2\hat{I}_{x,y}\end{aligned} \tag{5}$$

ゆがみ (skewness)(S)

$$\begin{aligned}S_X &= \frac{1}{\sigma_X^3}\sum_{x=1}^{w}\sum_{y=1}^{h}(x-\mu_X)^3\hat{I}_{x,y} \\ S_Y &= \frac{1}{\sigma_Y^3}\sum_{x=1}^{w}\sum_{y=1}^{h}(y-\mu_Y)^3\hat{I}_{x,y}\end{aligned} \tag{6}$$

扁平度 (flatness)(F)

$$\begin{aligned}F_X &= \frac{1}{\sigma_X^4}\sum_{x=1}^{w}\sum_{y=1}^{h}(x-\mu_X)^4\hat{I}_{x,y} \\ F_Y &= \frac{1}{\sigma_Y^4}\sum_{x=1}^{w}\sum_{y=1}^{h}(y-\mu_Y)^4\hat{I}_{x,y}\end{aligned} \tag{7}$$

[*2] 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。

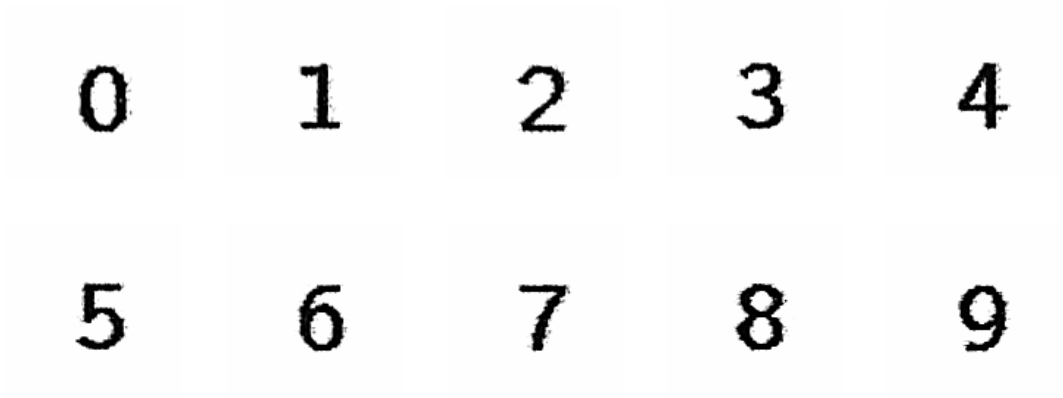

図 3　数字画像の例

1.3　特徴量の標準化

前節の説明に従って算出した特徴量は、それぞれ絶対値や分散が大きく異なります。これをベクトルとして組み合わせてそのまま評価・識別に用いると、絶対値の大きい特徴量の寄与が大きくなりすぎるという問題があるので、値のスケールを合わせる必要があります。ただし、単純にスケールを合わせるだけでは、もし大きく外れる値があった場合に、その他の値が狭い範囲に押し込められるという問題があります。

そこで各データから平均値を引き、その値を標準偏差で割るという操作をします。この操作を**標準化**とよびます。標準化によって、各次元は平均 0、分散 1 の分布となります。

1.4　プログラム作成上の注意

本実験では、Google Colaboratory を用いて PGM 画像を読み込み、特徴抽出を行います。サンプルのプログラムと画像データは Moodle からダウンロードし、使用してください。

次に Google Colaboratory に接続して、ファイルブラウザで作業用フォルダを指定し、サンプルプログラムを実行してみてください。sample1.ipynb は、特徴量計算の参考となるプログラムです。このプログラムでは PGM 画像を読み込み、画像を白黒反転させディスプレイに出力しています。なお、読み込む画像の変更が容易になるように、文字列と変数を結合して画像ファイル名を作成しています。

本実験の場合、im は 120×120 の行列となり、各要素は対応するピクセルの濃淡値になります。また、w は画像の横の画素数、h は縦の画素数になります。

1.5　PGM 画像を扱う上での注意

PGM 形式はシンプルで扱いやすいのですが、Unix 環境で主に使用されてきたため、Windows では標準でサポートされていません。画像を表示させるためには PGM 形式に対応した画像ビュワー（IrfanView など）を用います。また、レポート作成などのために PGM 形式を他の形式（JPG,PNG 等）に変換するには、画像ビュワーのセーブ機能を用います。

1.6　第 1 週の実験課題

- 入力画像の特徴量を計算し、CSV ファイルとして出力するプログラムを Google Colaboratory で作成せよ。
- （発展課題） 前処理部として画像に対するノイズフィルタを実装し、適用せよ。

レポートのポイント

- 作成したプログラムには適切なコメントを付けること（第 4 週の実験まで同様）。
- 特徴量の計算で示した重心・分散・ゆがみ・扁平度はそれぞれ画像のどのような特徴を表しているかを調べて解答せよ。
- 発展課題で複数のノイズフィルタを実装した場合は、それぞれの効果の違いを考察せよ。

2　特徴の評価（第 2 週）

第 2 週は数字画像を認識するにあたり、抽出した特徴を評価する実験を行います。

選択した特徴が識別に役立つのかどうかは、学習データが特徴空間でどのように分布しているのかを調べることでわかります。図 4 左のようにクラス毎にまとまっていて、かつ他のクラスのまとまりとは離れているような特徴の方が、図 4 右のように入り乱れている特徴よりも識別に役立ちそうだということがわかります。

2.1　特徴の定量的評価

特徴の良さを定量的にはかる手段として、**クラス内分散・クラス間分散比**というものがあります。

クラス内分散 σ_W^2 は、各クラスでデータの分散を計算

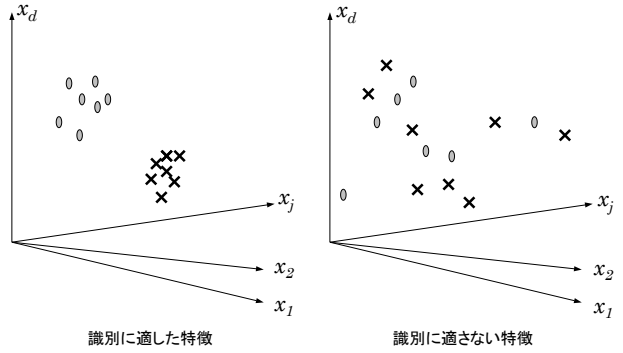

図 4　特徴の比較

し、それを全クラスについて足し合わせたものです。

$$\sigma_W^2 = \frac{1}{n}\sum_{i=1}^{c}\sum_{\boldsymbol{x}\in\chi_i}(\boldsymbol{x}-\boldsymbol{m}_i)^T(\boldsymbol{x}-\boldsymbol{m}_i) \qquad (8)$$

ただし、n は全データ数、c はクラス数、χ_i はクラス ω_i に属するデータ集合、$\boldsymbol{m}_i$ はクラス ω_i の平均ベクトルを表します。この σ_W^2 の値が小さければ小さいほど、データがまとまっているといえるので、よい特徴だということができます。

クラス間分散 σ_B^2 は、クラスの平均ベクトルの（データ個数による重み付きの）分散です。

$$\sigma_B^2 = \frac{1}{n}\sum_{i=1}^{c} n_i(\boldsymbol{m}_i-\boldsymbol{m})^T(\boldsymbol{m}_i-\boldsymbol{m}) \qquad (9)$$

ただし、n_i はクラス ω_i のデータ数、$\boldsymbol{m}$ は全データの平均ベクトルを表します。この σ_B^2 は大きければ大きいほど、クラスの中心同士が離れていることになるので、区別がつきやすい特徴だということができます。

これら 2 つを総合した、以下の式で表されるクラス内分散・クラス間分散比 J_σ は大きければ大きいほど、よい特徴空間だということができます。

$$J_\sigma = \frac{\sigma_B^2}{\sigma_W^2} \qquad (10)$$

ただし、J_σ はクラス間の平均的な距離しか考慮しておらず、識別にとって重要な評価尺度であるクラス間の重なりが反映されていないという欠点があります。従って、J_σ は選択した特徴量評価の目安にはなるものの、その有効性は別途確認してみないとわからないということになります。

2.2　特徴の評価の手順

まず 2 次元特徴空間のクラス内分散・クラス間分散比をすべての次元の組み合わせについて求めます。これが定量評価です。しかし、この定量評価はクラスの重なりを表現できないので、全面的に信用することはできません。そこで、定量評価値の高い 3 つ程度の組み合わせに対して 2 次元散布図を作成し、なるべくきれいにクラスが分離できている組み合わせを目視で求めます。これが定性評価になります。

sample2.ipynb は、特徴量評価のためのグラフを作成する際に参考にする Python プログラムです。このプログラムでは、CSV 形式の iris データ[*3]を読み込み、そこから特徴を 2 次元取り出して散布図としてプロットしています。使用するデータを今回の実験のものに差し替えて、必要な部分の修正を行って散布図を出力してください。

2.3　第 2 週の実験課題

- クラス内分散・クラス間分散比を用いて、識別に有効であると思われる 2 次元特徴の組み合わせを 3 つ程度求めよ（定量評価）。
- クラス内分散・クラス間分散比の値が大きい 3 つ程度の組み合わせで 2 次元散布図を作成し、識別に最も有効だと考えられる特徴の組み合わせを求めよ（定性評価）。
- （発展課題） 特徴ベクトルの次元数を 3 次元とし、評価の高い組み合わせを求めよ。

レポートのポイント

- 特徴量の有効性を評価した手順、およびその評価結果を報告すること。
- （発展課題）3 次元以上の組み合わせで評価が変化した場合、なぜそうなったか考察せよ。

3　数字画像の識別（第 3 週）

第 3 週の実験では、第 2 週の実験で得られた特徴ベクトルを入力として、パターンの識別を行います。識別手法として、最近傍決定則と統計的識別を実装します。

3.1　最近傍決定則による識別

第 2 週の実験結果から、識別能力の高いことが期待される 2 次元特徴量が見つかっているものとします。この選択した特徴で識別空間を構成し、学習データをプロットすると図 5 のように同じクラスのデータはある程度かたまっているはずです。最近傍決定則では、この塊を構成しているデータを基にクラス毎に 1 つのプロトタイプ（お手本）を設定し、識別したいデータと全プロトタイプとの距離を計算して最も近いものを識別結果とします。

[*3] 3 種類のアヤメの識別のためのデータ。個々のデータは萼の長さ・幅、花弁の長さ・幅からなる 4 次元ベクトル。
https://archive.ics.uci.edu/ml/datasets/iris

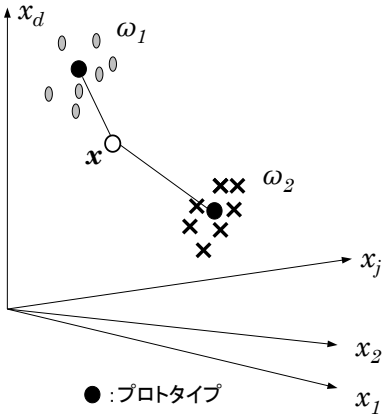

図 5　最近傍決定則の概念

3.2　統計的識別

統計的識別手法とは、特徴ベクトル $\boldsymbol{x}$ に対して、条件付き確率 $P(\omega_i|\boldsymbol{x})$（ただし ω_i はクラス、$1 \leq i \leq c$）を計算し、もっとも確率値の高いものを識別結果とするものです。ここで $P(\omega_i|\boldsymbol{x})$ は、特徴ベクトル $\boldsymbol{x}$ を観測した後で、それがクラス ω_i である確率をあらわしていることから、**事後確率**とよばれます。また、この判定法を事後確率最大法あるいは**ベイズ識別法**とよびます。

すべての特徴ベクトル $\boldsymbol{x}$ に対して統計を取って直接的にこの事後確率が求まればよいのですが、$\boldsymbol{x}$ は通常多次元の連続値ベクトルなので、これがぴったり一致するデータを数多く集めることは不可能です。従って、間接的に求めることにしましょう。

ベイズの定理という便利な確率論の定理があります。今、事象 A,B があるとして、その起こる確率をそれぞれ $P(A), P(B)$ で表わします。これらは確率なので 0 以上 1 以下の実数値を取ります。事象 A が観測されたもとで、事象 B が起こる確率を条件付き確率 $P(B|A)$ で表します。逆に、事象 B が観測されたもとで、事象 A が起こる確率は $P(A|B)$ となります。これらの値も確率なので 0 以上 1 以下の実数値です。これらの $P(A), P(B), P(B|A), P(A|B)$ の間には、以下の関係（ベイズの定理）があります。

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{11}$$

ここで、A に ω_i、B に $\boldsymbol{x}$ を入れると以下のようになります[*4]。

$$P(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)P(\omega_i)}{p(\mathbf{x})} \tag{12}$$

最大値を与える i の値を返す argmax 記法を用いると、ベイズ識別法は以下のように表せます。

$$\arg\max_i P(\omega_i|\mathbf{x}) = \arg\max_i \frac{p(\mathbf{x}|\omega_i)P(\omega_i)}{p(\mathbf{x})} \tag{13}$$

右辺を見てみると、その分母である $p(\boldsymbol{x})$ の値自体は i をいろいろ変えても変化しません。ということは、分母を取っ払って、以下の式でいいということになります。

$$\arg\max_i P(\omega_i|\mathbf{x}) = \arg\max_i p(\mathbf{x}|\omega_i)P(\omega_i) \tag{14}$$

ここで、$P(\omega_i)$ は、認識結果である各クラスが、それぞれどれくらいの確率で出現するかを求めればよいことになります。これは認識対象のデータが一定量得られれば、そこに含まれる各クラスの要素の割合で求めることができます。

$$P(\omega_i) = \frac{n_i}{n} \tag{15}$$

ただし、n は全データ数、n_i はそのうちのクラス ω_i に属するデータ数です。この確率は特徴ベクトル $\boldsymbol{x}$ の項を含んでいません。すなわち、特徴を観測する前に求めることができる確率なので、**事前確率**と言います。

$p(\boldsymbol{x}|\omega_i)$ は、クラス ω_i が特徴ベクトル $\boldsymbol{x}$ を生成する確率で、**尤度**とよびます。この尤度を計算するにあたっては、何らかの分布を仮定して、その分布のパラメータを与えられたデータから推定します。ここで正規分布を仮定すると以下の式のようになります。

$$p(\boldsymbol{x}|\omega_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^t\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)\} \tag{16}$$

ここで、$\boldsymbol{m}_i$ はクラス ω_i の平均ベクトル、Σ_i はクラス ω_i の共分散行列、d は特徴ベクトルの次元数、π はおなじみの円周率です。また、$|\Sigma_i|$、Σ_i^{-1} はそれぞれ共分散行列の行列式、逆行列をあらわします。図 6 に特徴ベクトルが 2 次元の場合の 2 クラスの分布の例を示します。

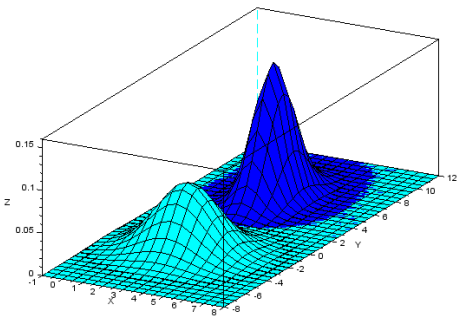

図 6　2 次元正規分布の例

[*4] $P(A)$ はある事象 A が生起する確率を表し、$p(x)$ は連続量 x に対して定義される確率密度関数を表します。

$\boldsymbol{x}$ が 1 次元の場合の正規分布の式は、平均を m_i、分散を σ_i^2 とすると、

$$p(x|\omega_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\{-\frac{(x-m_i)^2}{2\sigma_i^2}\} \qquad (17)$$

となります。1 次元の正規分布は図 7 のようになります。

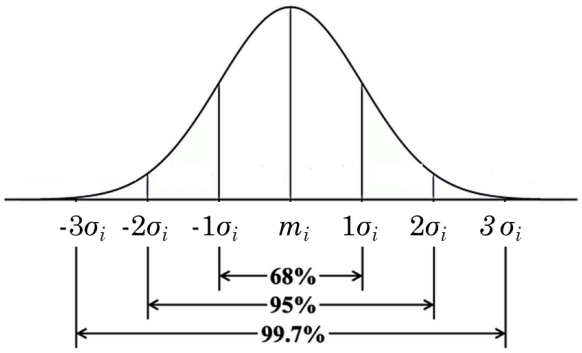

図 7　1 次元正規分布の例

多次元の正規分布は共分散行列を推定しなければならず、推定するパラメータの数が多くなります。そこで、各特徴の独立性を仮定することで、以下の式のように尤度を 1 次元の正規分布の積で近似することができます。

$$p(\boldsymbol{x}|\omega_i) \approx \prod_j p(x_j|\omega_i) \qquad (18)$$

ここで x_j は $\boldsymbol{x}$ の要素です。このような近似を行った識別をナイーブベイズ識別とよびます。

3.3　第 3 週の実験課題

- 2 週目で抽出した特徴ベクトルを元に最近傍決定則による識別を行うプログラムを Google Colaboratory で作成し、識別実験を行え。プロトタイプの設定法は各自で十分検討すること。なお、第 3 週の実験では、学習に用いたデータで識別性能を評価してもよい。
- 上記実験の識別器をナイーブベイズによる識別に置き換え、同様の識別実験を行え。
- （発展課題）統計的識別において、ナイーブベイズ仮定を採用せずに、2 次元正規分布を推定する方法を用いた識別器を作成せよ。
- （発展課題）2 週目において 3 次元の特徴空間を評価した場合は、選択した 3 次元特徴空間に対して最近傍決定則、ナイーブベイズ、3 次元正規分布の推定のそれぞれを行え。

レポートのポイント

- プロトタイプの決め方を複数検討し、最終的に何を根拠に決めたかを説明せよ。
- ナイーブベイズ識別では、何が学習結果として得られたかを明確にせよ。

4　識別器の評価（第 4 週）

第 4 週では、第 3 週で作成した識別器を評価する方法について学びます。

第 3 週では、識別器の動作確認のために学習に用いたデータで識別性能を評価しました。しかしこれではパターン認識器の評価にはなりません。入力と出力の対をすべて記憶しておけば識別率 100% の識別器が作れますし、そのような方法でなくてもパラメータの多い複雑な識別器では容易に識別率 100% とすることができます。そのようなシステムは学習データに過剰に適応してしまっている（過学習を起こしている）可能性が高く、実運用の際に入力となる新しいデータに対しては誤認識を頻繁に起こしてしまうということがよくあります。

ここではこのような未知データに対する識別率の評価法について説明します。

4.1　分割学習法

未知データに対する認識率を求めるもっとも簡単な方法は、手元の全データ χ を学習用データ χ_T と評価用データ χ_E に分割することです（図 8）。学習用データを使って識別部のパラメータ学習を行い、評価用データで認識率を測ります。評価用データは学習に使っていないので、これを未知データとみなすわけです。この方法を**分割学習法**といいます。

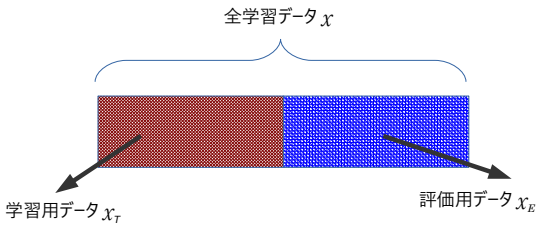

図 8　分割学習法

4.2　交差確認法

交差確認法は図 9 に示すように学習データを m 個の集合に分割し、そのうちの 1 つだけを評価用に除外し、残りの $m-1$ 個で学習を行います。さらにその除外するデータを順に交換することで、合計 m 回の学習と評価を行います。これで、全データがひととおり評価に使われ、かつその評価時に用いられる識別器は評価用データを除いて構築されたものとなっています。技術論文では m を 10 とする場合や、「データの個数」とするケースがよく見られます。m がデータの個数の場合を **1 つ抜き法**とよび、学習データの個数が少ない場合に用いられ

ます。

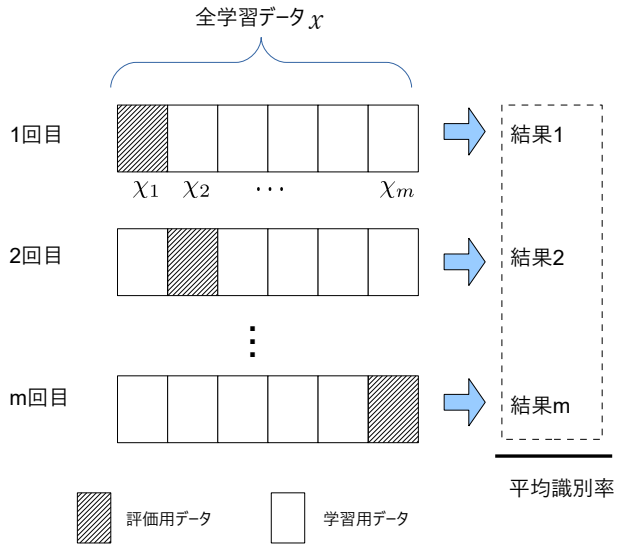

図 9　交差確認法

4.3　第 4 週の実験課題

- 第 3 週に作成した識別器について分割学習法、交差確認法のそれぞれで評価を行うコードを Google Colaboratory で作成せよ。
- 分割学習法、交差確認法それぞれの評価結果を比較し、得失を論ぜよ。
- （発展課題）最近傍決定則の発展として、近傍 k 個のデータが属するクラスの多数決を識別結果とする k-NN 法を実装し、異なる k について交差確認法で性能を評価せよ。
- （発展課題）交差確認法においてデータ分割の方法を乱数を用いて変化させ、複数回実行して平均値を求める方法を実装せよ。

レポートのポイント

- それぞれの評価法について、評価に先立って決めておかなくてはならないパラメータに関して、なぜそのように決めたかを説明すること。
- 2 つの評価結果に差が出た場合は、なぜそのような結果になったかを考察すること。

参考文献

1. 荒木 雅弘：フリーソフトでつくる音声認識システム - パターン認識・機械学習の初歩から対話システムまで -（第 2 版）, 森北出版, 2017.
2. 石井 健一郎, 前田 英作, 上田 修功, 村瀬 洋: わかりやすいパターン認識 第 2 版, オーム社, 2019.