そのような理由でeを底とした指数関数がいろいろな場面で使われているのでしょう。4.7.5 項から述べるシグモイド関数やソフトマックス関数、ガウス関数でも一般的にeが使われています。

4.7.4 対数関数の微分

対数関数の微分は、反比例の式 4-111 になります（リスト 4-4-(5)、図 4.32）。

$$y' = (\log x)' = \frac{1}{x} \tag{4-111}$$

In

```
# リスト 4-4-(5)
x = np.linspace(0.0001, 4, 100)  # 0以下では定義できない
y = np.log(x)
dy = 1 / x

plt.figure(figsize=(4, 4))
plt.plot(x, y, 'gray', linestyle='--', linewidth=3)
plt.plot(x, dy, color='black', linewidth=3)
plt.ylim(-8, 8)
plt.xlim(-1, 4)
plt.grid(True)
plt.show()
```

Out

```
# 実行結果は図 4.32 を参照
```


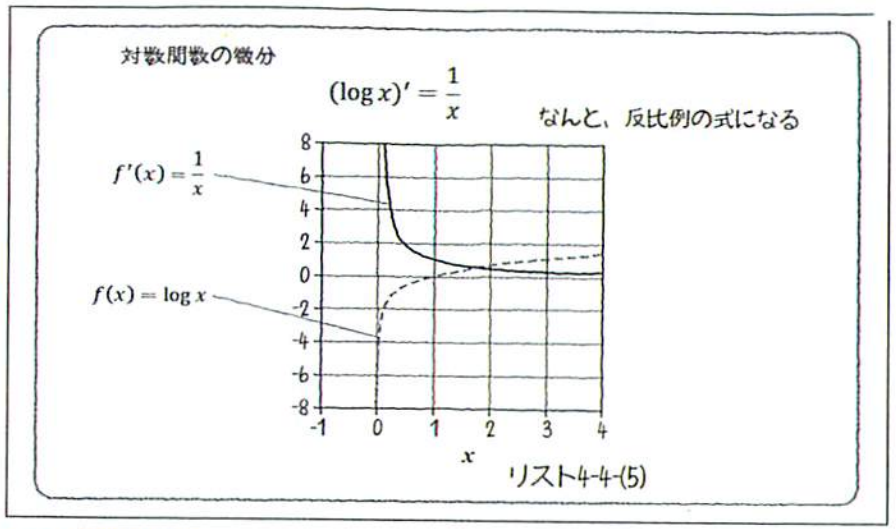

図 4.32：対数関数の微分

6.1 節では、$\{\log(1-x)\}'$という形の微分も出てきますが、$z = 1 - x$とおいて、

$$y = \log z, \qquad z = 1 - x$$

としておけば、連鎖律を使って、式 4-112 のように導き出せます。

$$\frac{dy}{dx} = \frac{dy}{dz} \cdot \frac{dz}{dx} = \frac{1}{z} \cdot (-1) = -\frac{1}{1-x} \tag{4-112}$$

4.7.5 シグモイド関数

シグモイド関数は、式 4-113 のように定義される滑らかな階段のような関数です。

$$y = \frac{1}{1 + e^{-x}} \tag{4-113}$$

e^{-x}は$\exp(-x)$とも書けるので、式 4-114 のように表される場合もあります。

$$y = \frac{1}{1 + \exp(-x)} \tag{4-114}$$

グラフに描くと（リスト 4-4-(6)）、図 4.33 のようになります。

In

```
# リスト 4-4-(6)
x = np.linspace(-10, 10, 100)
y = 1 / (1 + np.exp(-x))

plt.figure(figsize=(4, 4))
plt.plot(x, y, 'black', linewidth=3)

plt.ylim(-1, 2)
plt.xlim(-10, 10)
plt.grid(True)
plt.show()
```

Out

```
# 実行結果は図 4.33 を参照
```

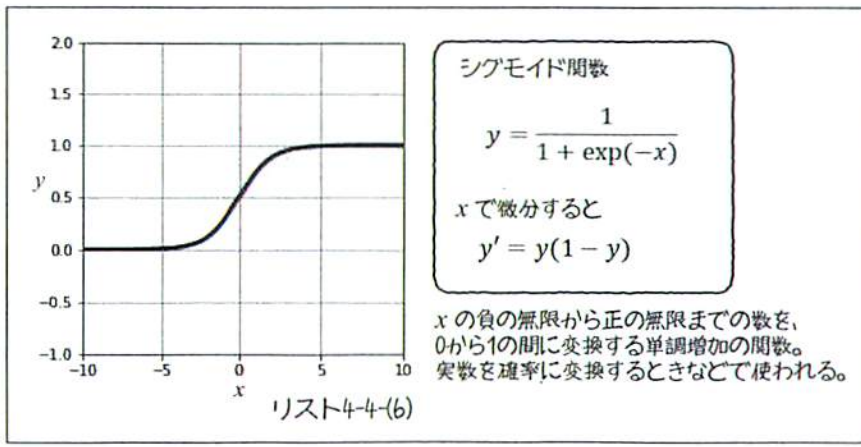

図 4.33：シグモイド関数

シグモイド関数は、負から正の実数を 0 から 1 までの間に変換しますので確率を表すときによく使われます（この関数は、出力の範囲が 0 から 1 になるように無理やり作ったものではなく、ある条件のもとで自然に導出されます）。

シグモイド関数は、第 6 章の分類問題で登場します。また、第 7 章のニューラルネットでも、ニューロンの特性を表す重要な関数として登場します。第 6 章や第 7 章でシグモイド関数の微分を使う場面がありますので、ここで導き出しましょう。

微分の公式 4-115 を考えます。これが、式 4-113 に当てはまるように、$f(x) = 1 + \exp(-x)$と考えます。

$$\left(\frac{1}{f(x)}\right)' = -\frac{f'(x)}{f(x)^2} \tag{4-115}$$

この$f(x)$の微分は$f'(x) = -\exp(-x)$です。よって、式 4-116 を得ます。

$$y' = \left(\frac{1}{1 + \exp(-x)}\right)' = -\frac{-\exp(-x)}{(1 + \exp(-x))^2} = \frac{\exp(-x)}{(1 + \exp(-x))^2} \tag{4-116}$$

式 4-116 を、少し変形します（式 4-117）。

$$\begin{aligned} y' &= \frac{1}{1 + \exp(-x)} \cdot \frac{1 + \exp(-x) - 1}{1 + \exp(-x)} \\ &= \frac{1}{1 + \exp(-x)} \cdot \left\{1 - \frac{1}{1 + \exp(-x)}\right\} \end{aligned} \tag{4-117}$$

ここで、$1/(1 + \exp(-x))$は、yそのものであったので、yで書き換えると、式 4-118 のようにすっきりした形になります。

$$y' = y(1 - y) \tag{4-118}$$

4.7.6 ソフトマックス関数

例えば、3 つの数、$x_0 = 2$、$x_1 = 1$、$x_2 = -1$ があって、これらの数の大小関係を保ったまま、それぞれを確率を表す y_0、y_1、y_2 に変換したいとします。確率ですので、0 から 1 までの数値でなくてはいけません。また、すべてを足したら 1 となっている必要があります。

このようなときに使われるのがソフトマックス関数です。まず、各x_i の exp の和 u を求めておきます（式 4-119）。

$$u = \exp(x_0) + \exp(x_1) + \exp(x_2) \tag{4-119}$$

変換式は、式 4-119 を使って式 4-120 のようになります。

$$y_0 = \frac{\exp(x_0)}{u}, y_1 = \frac{\exp(x_1)}{u}, y_2 = \frac{\exp(x_2)}{u} \tag{4-120}$$

実際にプログラムでソフトマックス関数を作り、テストしてみましょう（リスト 4-4-(7)）。

In

```
# リスト 4-4-(7)
def softmax(x0, x1, x2):
    u = np.exp(x0) + np.exp(x1) + np.exp(x2)
    return np.exp(x0) / u, np.exp(x1) / u, np.exp(x2) / u



# test
y = softmax(2, 1, -1)
print(np.round(y,2))  # (A) 小数点以下２桁の概数を表示
print(np.sum(y))      # (B) 和を表示
```

Out

```
[ 0.71  0.26  0.04]
1.0
```

先ほどの例の$x_0 = 2$、$x_1 = 1$、$x_2 = -1$が、$y_0 = 0.71$、$y_1 = 0.26$、$y_2 = 0.04$に変換されました。確かに、順番を保ったまま 0 から 1 の数値が割り当てられています。すべてを足すと 1 になることも確かめられました。

ソフトマックス関数を図示するとどのようになるでしょうか？　入力と出力が 3 次元ですので、そのまま図示するというわけにはいきません。そこで、x_2だけ$x_2 = 1$と固定して、いろいろなx_0とx_1を入力したときのy_0とy_1をプロットしてみましょう（リスト 4-4-(8)、図 4.34）。

In

```
# リスト 4-4-(8)

from mpl_toolkits.mplot3d import Axes3D

xn = 20
x0 = np.linspace(-4, 4, xn)
x1 = np.linspace(-4, 4, xn)

y = np.zeros((xn, xn, 3))
for i0 in range(xn):
    for i1 in range(xn):
        y[i1, i0, :] = softmax(x0[i0], x1[i1], 1)

xx0, xx1 = np.meshgrid(x0, x1)
plt.figure(figsize=(8, 3))
for i in range(2):
    ax = plt.subplot(1, 2, i + 1, projection='3d')
    ax.plot_surface(xx0, xx1, y[:, :, i],
                    rstride=1,  cstride=1, alpha=0.3,
                    color='blue', edgecolor='black')
    ax.set_xlabel('$x_0$', fontsize=14)
    ax.set_ylabel('$x_1$', fontsize=14)
    ax.view_init(40, -125)

plt.show()
```

Out

```
# 実行結果は図 4.34 を参照
```

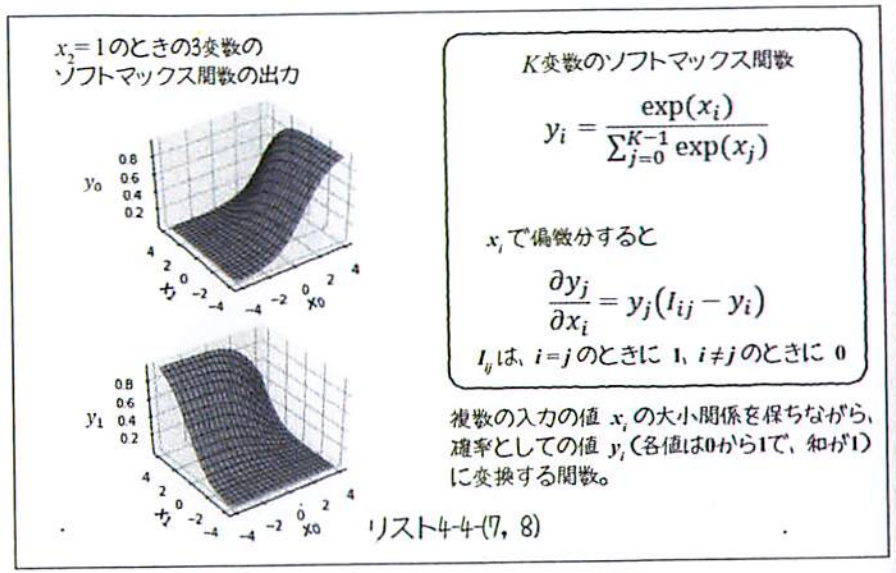

図 4.34：ソフトマックス関数

x_2を 1 に固定して、x_0 とx_1を動かすと、y_0、y_1 は 0 と 1 の間の値で変化します（図 4.34 左）。x_0が大きくなれば、y_0は 1 に近づき、x_1が大きくなればy_1 が 1 に近づきます。y_2は図示していませんが、y_2は、1 からy_0とy_1を引いた残りですので、なんとか想像できるでしょう。

ソフトマックス関数は、3 つの変数だけでなくそれ以上の変数にも使えます。変数の数を K としたら、式 4-121 のように表すことができます。

$$y_i = \frac{\exp(x_i)}{\sum_{j=0}^{K-1}\exp(x_j)} \tag{4-121}$$

ソフトマックス関数の偏微分は、第 7 章で出てきますので、ここで求めておきます。まず、y_0をx_0で偏微分してみましょう（式 4-122）。

$$\frac{\partial y_0}{\partial x_0} = \frac{\partial}{\partial x_0}\frac{\exp(x_0)}{u} \tag{4-122}$$

ここで注意しなくてはならないのは、uもx_0の関数だということです。ですので、式 4-123 の微分の公式を使って、$f(x) = u = \exp(x_0) + \exp(x_1) + \exp(x_2)$、$g(x) = \exp(x_0)$とします。

$$\left(\frac{g(x)}{f(x)}\right)' = \frac{g'(x)f(x) - g(x)f'(x)}{f(x)^2} \tag{4-123}$$

ここで式 4-124 のように、$f'(x) = \partial f/\partial x_0$、$g'(x) = \partial g/\partial x_0$と考えます。

$$\begin{aligned} f'(x) &= \frac{\partial}{\partial x_0}f(x) = \exp(x_0) \\ g'(x) &= \frac{\partial}{\partial x_0}g(x) = \exp(x_0) \end{aligned} \tag{4-124}$$

よって、式 4-123 は、式 4-125 のようになります。

$$\begin{aligned} \frac{\partial y_0}{\partial x_0} = \left(\frac{g(x)}{f(x)}\right)' &= \frac{\exp(x_0)u - \exp(x_0)\exp(x_0)}{u^2} \\ &= \frac{\exp(x_0)}{u}\left(\frac{u - \exp(x_0)}{u}\right) \\ &= \frac{\exp(x_0)}{u}\left(\frac{u}{u} - \frac{\exp(x_0)}{u}\right) \end{aligned} \tag{4-125}$$

ここで、$y_0 = \exp(x_0)/u$であったことを使って、式 4-126 のように表します。

$$\frac{\partial y_0}{\partial x_0} = y_0(1 - y_0) \tag{4-126}$$

驚いたことに、シグモイド関数の微分（式 4-118）と同じ形になりました。
それでは、次に、y_0をx_1で偏微分してみましょう（式 4-127）。

$$\frac{\partial y_0}{\partial x_1} = \frac{\partial}{\partial x_1}\frac{\exp(x_0)}{u} \tag{4-127}$$

ここでも、$f(x) = u = \exp(x_0) + \exp(x_1) + \exp(x_2)$、$g(x) = \exp(x_0)$として、公式 4-123 を使います。

$$\left(\frac{g(x)}{f(x)}\right)' = \frac{g'(x)f(x) - g(x)f'(x)}{f(x)^2} \tag{4-123}$$

ここで、$f'(x) = \partial f/\partial x_1$、$g'(x) = \partial g/\partial x_1$と、$x_1$による偏微分を考えます。

$$f'(x) = \frac{\partial}{\partial x_1} f(x) = \exp(x_1)$$
$$g'(x) = \frac{\partial}{\partial x_1} \exp(x_0) = 0$$

よって式 4-128 のようになります。

$$\frac{\partial y_0}{\partial x_1} = \frac{g'(x)f(x) - g(x)f'(x)}{f(x)^2} = \frac{-\exp(x_0)\exp(x_1)}{u^2} = -\frac{\exp(x_0)}{u} \cdot \frac{\exp(x_1)}{u} \tag{4-128}$$

ここで、$y_0 = \exp(x_0)/u$、$y_1 = \exp(x_1)/u$であったことを使うと、式 4-129 が得られます。

$$\frac{\partial y_0}{\partial x_1} = -y_0 y_1 \tag{4-129}$$

式 4-126 と式 4-129 をまとめて、式 4-130 のように表すこともできます。

$$\frac{\partial y_j}{\partial x_i} = y_j(I_{ij} - y_i) \tag{4-130}$$

ここで、I_{ij}は、$i = j$のときに 1、$i \neq j$のときに 0 となる関数です。I_{ij}は、δ_{ij}とも表され、クロネッカーのデルタと呼ばれています。

4.7.7 ソフトマックス関数とシグモイド関数

それにしても、ソフトマックス関数とシグモイド関数は似ています。これは、どういう関係なのでしょうか？　ここで、その関係を考えてみましょう。2 変数の場合のソフトマックス関数は、式 4-131 のようになります。

$$y = \frac{e^{x_0}}{e^{x_0} + e^{x_1}} \tag{4-131}$$

分母分子にe^{-x_0}を掛けて整理すると、$e^a e^{-b} = e^{a-b}$という公式を使って、式 4-132 を得ることができます。

$$y = \frac{e^{x_0}e^{-x_0}}{e^{x_0}e^{-x_0} + e^{x_1}e^{-x_0}} = \frac{e^{x_0 - x_0}}{e^{x_0 - x_0} + e^{x_1 - x_0}} = \frac{1}{1 + e^{-(x_0 - x_1)}} \tag{4-132}$$

ここで、$x = x_0 - x_1$とおけば、式 4-133 のようにシグモイド関数になりました。

$$y = \frac{1}{1 + e^{-x}} \tag{4-133}$$

つまり、2 変数のソフトマックス関数の入力x_0、x_1を、その差$x = x_0 - x_1$で表したものがシグモイド関数なのです。シグモイド関数を多変数に拡張したものがソフトマックス関数だとも言えます。

4.7.8 ガウス関数

ガウス関数は、式 4-134 のような関数です。

$$y = \exp(-x^2) \tag{4-134}$$

予測精度が増すということです。

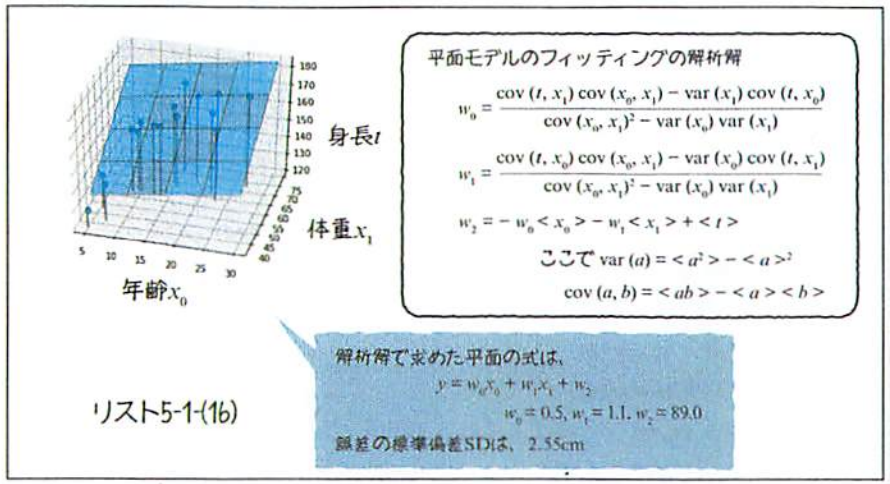

図 5.11：解析解による平面モデルのフィッティング結果

5.3 D 次元線形回帰モデル

それでは、**x** が 3 次元、4 次元、いや、もっと多くの次元だったらどうでしょうか？異なる次元に対してすべての公式を導いていたら大変な手間になってしまいます。そこで、D 次元として次元数も変数としたまま公式を導くことを考えます。

実は、ここは本書での重要な「峠」となります。5.1 節と 5.2 節の 1 次元データ、2 次元データでの解析解の導出は、この D 次元での解析解の導出を行う準備運動のようなものでした。この難所を超えることができれば、難解な機械学習の教科書を読む力が「ぐっ」とつくはずです。気合を入れて進みましょう。

5.3.1 D 次元線形回帰モデル

1 次元入力で扱った直線モデル、2 次元入力で扱った面モデルは、まとめて**線形回帰モデル**と呼ばれる同じ種類のモデルです。一般的には、式 5-37 のように表すことができます。

$$y(\mathbf{x}) = w_0 x_0 + w_1 x_1 + \cdots + w_{D-1} x_{D-1} + w_D \quad (5\text{-}37)$$

最後の w_D は切片を表し、x が掛けられていないことに注意してください。しかし、まずは、簡単のために切片の項を含めないモデルで考えていきます（式 5-38）。

$$y(\mathbf{x}) = w_0 x_0 + w_1 x_1 + \cdots + w_{D-1} x_{D-1} \quad (5\text{-}38)$$

切片 w_D がモデルに含まれていないと、どのような **w** でも原点 **x** =[0, 0, ..., 0] を代入したときに y が 0 になります。つまり、このモデルでは、どんな **w** でも原点を通る平面（高次元空間の面のようなもの）になります。切片がないとグラフが上下に平行移動できないからです。

さて、このモデルを行列表記を使って短くまとめると、式 5-39 の右辺のように $\mathbf{w}^T\mathbf{x}$ と表すことができます。

$$y(\mathbf{x}) = w_0 x_0 + w_1 x_1 + \cdots + w_{D-1} x_{D-1} = [w_0 \ \cdots \ w_{D-1}] \begin{bmatrix} x_0 \\ \vdots \\ x_{D-1} \end{bmatrix} = \mathbf{w}^T\mathbf{x} \quad (5\text{-}39)$$

w は、以下のようになります。

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{D-1} \end{bmatrix}$$

$\mathbf{w}^T$ はこれを横にした横ベクトルです。

5.3.2 パラメータの解析解

さて、ここから解析解を求めていきます。これまでと同様に平均二乗誤差 J を式 5-40 のように表します。

$$J(\mathbf{w}) = \frac{1}{N}\sum_{n=0}^{N-1}(y(\mathbf{x}_n) - t_n)^2 = \frac{1}{N}\sum_{n=0}^{N-1}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n - t_n)^2 \qquad \textbf{(5-40)}$$

上記をおなじみの連鎖律を使って、w_i で偏微分すると、式 5-41 のようになります。

$$\frac{\partial J}{\partial w_i} = \frac{1}{N}\sum_{n=0}^{N-1}\frac{\partial}{\partial w_i}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n - t_n)^2 = \frac{2}{N}\sum_{n=0}^{N-1}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n - t_n)\,x_{n,i} \qquad \textbf{(5-41)}$$

なお、$\mathbf{w}^{\mathrm{T}}\mathbf{x}_n = w_0 x_{n,0} + \cdots + w_{D-1} x_{n,D-1}$ を w_i で偏微分すると、$x_{n,i}$ だけが残ることに注意してください。

J を最小にする $\mathbf{w}$ では、すべての w_i の方向について傾きが 0、つまり偏微分（式 5-41）が 0 となるので、式 5-42 が、$i = 0 \sim D-1$ で成り立つはずです。

$$\frac{2}{N}\sum_{n=0}^{N-1}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n - t_n)\,x_{n,i} = 0 \qquad \textbf{(5-42)}$$

つまり、この D 個の連立方程式を各 w_i について解けば、答えを得ることができるというわけです。まず、両辺を $N/2$ 倍して、少しだけシンプルにした式 5-43 を考えていきましょう。

$$\sum_{n=0}^{N-1}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n - t_n)\,x_{n,i} = 0 \qquad \textbf{(5-43)}$$

それにしても、これまでは D を $D = 1$、$D = 2$ と具体的に決めて $\mathbf{w}$ を導出してきましたが、これを D という変数のまま $\mathbf{w}$ を求めることなどできるのでしょうか？

ここで行列の出番になります。行列を使うと D は D のまま答えが出せるのです。

それではまず、式 5-43 全体をベクトル形式としてまとめます。式 5-43 は、すべての i で成り立つので、それぞれを丁寧に書いていけば、式 5-44 のようになります。

$$\begin{aligned}
&\sum_{n=0}^{N-1}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n - t_n)\,x_{n,0} = 0\\
&\sum_{n=0}^{N-1}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n - t_n)\,x_{n,1} = 0\\
&\vdots\\
&\sum_{n=0}^{N-1}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n - t_n)\,x_{n,D-1} = 0
\end{aligned} \qquad \textbf{(5-44)}$$

最後の x の添字だけが 0 から D-1 まで変わっています。これらの式を、ベクトルとして 1 つにまとめて、式 5-45 のように表すことができます。右辺は D 次元の 0 ベクトルです。

$$\sum_{n=0}^{N-1}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n - t_n)[x_{n,0}, x_{n,1}, \cdots, x_{n,D-1}] = [0 \quad 0 \quad \cdots \quad 0] \qquad \textbf{(5-45)}$$

そして、$[x_{n,0}, x_{n,1}, \cdots, x_{n,D-1}]$ は $\mathbf{x}_n^{\mathrm{T}}$ なので、式 5-46 のようになります。

$$\sum_{n=0}^{N-1}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n - t_n)\mathbf{x}_n^{\mathrm{T}} = [0 \quad 0 \quad \cdots \quad 0] \qquad \textbf{(5-46)}$$

これで、式 5-43 がベクトル形式として変換されました。行列でも $(a + b)c = ac + bc$ の法則（分配法則）が成り立つので、式 5-47 のように展開することができます。

$$\sum_{n=0}^{N-1}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}} - t_n\mathbf{x}_n^{\mathrm{T}}) = [0 \quad 0 \quad \cdots \quad 0] \qquad \textbf{(5-47)}$$

そして、和を展開して式 5-48 のようになります。

$$\mathbf{w}^{\mathrm{T}}\sum_{n=0}^{N-1}\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}} - \sum_{n=0}^{N-1}t_n\mathbf{x}_n^{\mathrm{T}} = [0 \quad 0 \quad \cdots \quad 0] \qquad \textbf{(5-48)}$$

この左辺は、式 5-49 のように行列の式で表すことができます。

$$\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X} - \mathbf{t}^{\mathrm{T}}\mathbf{X} = [0 \quad 0 \quad \cdots \quad 0] \tag{5-49}$$

ここで $\mathbf{X}$ は、すべてのデータをまとめて表した行列、式 5-50 です。データ行列 $\mathbf{X}$ は、個々のデータベクトルの転置 $\mathbf{x}_n^{\mathrm{T}}$ を縦方向に並べたベクトルのベクトルとしてみなすことができます。

$$\mathbf{X} = \begin{bmatrix} x_{0,0} & x_{0,1} & \cdots & x_{0,D-1} \\ x_{1,0} & x_{1,1} & \cdots & x_{1,D-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N-1,0} & x_{N-1,1} & \cdots & x_{N-1,D-1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_0^{\mathrm{T}} \\ \mathbf{x}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_{N-1}^{\mathrm{T}} \end{bmatrix} \tag{5-50}$$

式 5-48 から式 5-49 への変換には、以下の式 5-51、5-52 を使いました。

$$\sum_{n=0}^{N-1} \mathbf{x}_n \mathbf{x}_n^{\mathrm{T}} = \mathbf{X}^{\mathrm{T}}\mathbf{X} \tag{5-51}$$

$$\sum_{n=0}^{N-1} t_n \mathbf{x}_n^{\mathrm{T}} = \mathbf{t}^{\mathrm{T}}\mathbf{X} \tag{5-52}$$

データ行列 $\mathbf{X}$ が、式 5-50 の右式のように $\mathbf{x}_n^{\mathrm{T}}$ で表せることに注意すると、行列計算に慣れた方なら式 5-51 と式 5-52 が成り立つことが見えてくるかもしれません。

もちろん、成分表記にして確かめることもできます。式 5-51 は、左辺と右辺を成分表記の行列にすると、どちらも式 5-53 のようになることで、等号が成り立つとわかります。$N = 2$、$D = 2$ を想定すると確かめるのも楽になります。

$$\begin{bmatrix} \sum_{n=0}^{N-1} x_{n,0}^2 & \sum_{n=0}^{N-1} x_{n,0}x_{n,1} & \cdots & \sum_{n=0}^{N-1} x_{n,0}x_{n,D-1} \\ \sum_{n=0}^{N-1} x_{n,1}x_{n,0} & \sum_{n=0}^{N-1} x_{n,1}^2 & \cdots & \sum_{n=0}^{N-1} x_{n,1}x_{n,D-1} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{n=0}^{N-1} x_{n,D-1}x_{n,0} & \sum_{n=0}^{N-1} x_{n,D-1}x_{n,1} & \cdots & \sum_{n=0}^{N-1} x_{n,D-1}^2 \end{bmatrix} \tag{5-53}$$

式 5-52 も、左辺と右辺を成分表記にすると、どちらも式 5-54 のようになることから、等号が成り立つとわかります。

$$\left[\sum_{n=0}^{N-1} t_n x_{n,0} \quad \sum_{n=0}^{N-1} t_n x_{n,1} \quad \cdots \quad \sum_{n=0}^{N-1} t_n x_{n,D-1}\right] \tag{5-54}$$

さて、ここからは、式 5-49 を変形して $\mathbf{w} =$ の形に持っていくことを考えます。まず、両辺を転置すると、式 5-55 のようになります。

$$(\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X} - \mathbf{t}^{\mathrm{T}}\mathbf{X})^{\mathrm{T}} = [0 \quad 0 \quad \cdots \quad 0]^{\mathrm{T}} \tag{5-55}$$

上記の左辺の 2 つの項に外側の T を作用させると式 5-56 のようになります。$(\boldsymbol{A} + \boldsymbol{B})^{\mathrm{T}} = \mathbf{A}^{\mathrm{T}} + \mathbf{B}^{\mathrm{T}}$ という関係式を使いました。

$$(\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X})^{\mathrm{T}} - (\mathbf{t}^{\mathrm{T}}\mathbf{X})^{\mathrm{T}} = [0 \quad 0 \quad \cdots \quad 0]^{\mathrm{T}} \tag{5-56}$$

更に $(\mathbf{A}^{\mathrm{T}})^{\mathrm{T}} = \mathbf{A}$ という関係式と、$(\mathbf{AB})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}$ という公式（4.6.7 項）を使って、式 5-57 を得ます。

$$(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{\mathrm{T}}(\mathbf{w}^{\mathrm{T}})^{\mathrm{T}} - \mathbf{X}^{\mathrm{T}}\mathbf{t} = [0 \quad 0 \quad \cdots \quad 0]^{\mathrm{T}} \tag{5-57}$$

左辺の第 1 項では、$\mathbf{w}^{\mathrm{T}} = \mathbf{A}$、$\mathbf{X}^{\mathrm{T}}\mathbf{X} = \mathbf{B}$ と考えました。更に、左辺第 1 項は、式 5-58 のように整理できます。

$$(\mathbf{X}^{\mathrm{T}}\mathbf{X})\mathbf{w} - \mathbf{X}^{\mathrm{T}}\mathbf{t} = [0 \quad 0 \quad \cdots \quad 0]^{\mathrm{T}} \tag{5-58}$$

上記の $\mathbf{X}^{\mathrm{T}}\mathbf{t}$ を右辺に移行して、式 5-59 のようになります。

$$(\mathbf{X}^{\mathrm{T}}\mathbf{X})\mathbf{w} = \mathbf{X}^{\mathrm{T}}\mathbf{t} \tag{5-59}$$

最後に、左辺の $(\mathbf{X}^{\mathrm{T}}\mathbf{X})$ を消すために $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$ を両辺に左から掛ければ、式 5-60 のように解析解を得ます。

$$\mathbf{w} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t} \tag{5-60}$$

これが、まさしく、D 次元線形回帰モデルの解となります。峠を無事に超えられたでしょうか？　ごくろうさまでした。

式 5-60 は、$\mathbf{x}$ が何次元だったとしてもすべてこの形で最適な $\mathbf{w}$ が得られるという結果です。シンプルで何とも綺麗な形です。この式の右辺 $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ には、**ムーア・ペンローズの擬似逆行列**という名前が付いています。逆行列は、縦と横の長さが同じ正方行列にしか定義できませんでしたが（4.6 節「行列」を参照）、擬似逆行列は、正方行列ではない行列（ここでは $\mathbf{X}$）でも定義できる逆行列の代用バージョンになっているのです。

5.3.3 原点を通らない面への拡張

さて、保留にしていた原点を通らない面の場合へ拡張する話に戻ります。原点に固定された面の方程式は、入力データが 2 次元の場合、式 5-61 のように表されます。

$$y(\mathbf{x}) = w_0 x_0 + w_1 x_1 \tag{5-61}$$

ここに 3 番目のパラメータ w_2 を加えれば、面が上下に移動できるので、原点を通らない面が表現できます（式 5-62）。

$$y(\mathbf{x}) = w_0 x_0 + w_1 x_1 + w_2 \tag{5-62}$$

そこで、$\mathbf{x}$ は 2 次元ベクトルでしたが、常に 1 をとる 3 次元目の要素 $x_2 = 1$ を追加して、$\mathbf{x}$ を 3 次元ベクトルだと考えます。すると、式 5-63 のように表すことができ、原点にしばられない面を表現できることになります。

$$y(\mathbf{x}) = w_0 x_0 + w_1 x_1 + w_2 x_2 = w_0 x_0 + w_1 x_1 + w_2 \tag{5-63}$$

このように、常に 1 をとる次元を入力データ $\mathbf{x}$ に追加してから、式 5-60 を適用することで、原点にしばられない面を求めることができます。これは、D 次元の $\mathbf{x}$ の問題についても同様で、$D + 1$ 次元目に常に 1 をとる要素を追加すれば、自由に動けるモデルが表現できるのです。

5.4 線形基底関数モデル

$\mathbf{x}$ が 1 次元の場合に話を戻します。ここまでは、直線のモデルを使って身長の予測を考えてきました。しかし、データを見てみると、なだらかに曲がった曲線にデータは沿っているようにも見えます（図 5.12）。曲線を使って表したほうがもっと誤差が小さくなるかもしれません。そこで、曲線のモデルを考えていきましょう。

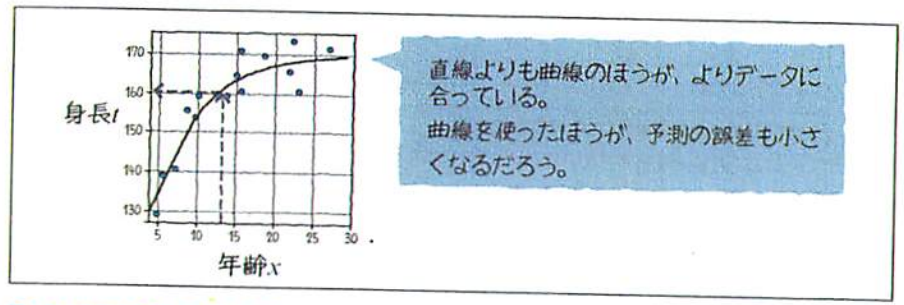

図 5.12：曲線によるフィッティング

曲線を表すモデルはいろいろな種類がありますが、ここでは汎用性の高い**線形基底関数モデル**を紹介します。基底関数とは「もとになる関数」という意味です。5.3 節で紹介した線形回帰モデルの x を基底関数 $\phi(x)$ に置き換えることで、いろいろな形の関数を作ろうというのが、線形基底関数モデルの考え方です。

まず、何を基底関数とするかを選ぶ必要があるのですが、ここではガウス関数を基底関数に選んだ線形基底関数モデルを考えます。

基底関数は $\phi_j(x)$ で表します。ϕ は、「ファイ」と呼ぶギリシャ文字です。基底関数は複数セットで使われるので、その番号を表す j というインデックスが付いています。ガウス基底関数は、式 5-64 のようになります。

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\} \tag{5-64}$$

ガウス関数の中心位置は μ_j です。これはモデルの設計者が決めるパラメータとなっています。関数の広がりの程度は s で調節されます。これも設計者が決めるパラメータです。s はすべてのガウス関数で共通のパラメータとします。

さて、第 5 章のプログラムもだいぶ長くなりましたので、ここからでも新規にプ