さて、この場合、決定境界はどこに引くべきでしょうか？　やはり、白黒をはっきりさせたいこともあります。決定境界の右側ではオスと予測したほうが当たる確率が高くなり、決定境界の左側ではメスと予測したほうが当たる確率が高くなるように決定境界を設定すべきです。そう考えれば $P(t=1|x)=0.5$ となる x が決定境界となります。この例では、$x=1.2$ が決定境界となります。

ここまでの議論は、「確率で表すことが優れている」ということを説明するために、「データの真の分布を知っている」という特殊な状況を仮定しました。実際には真の分布は手持ちのデータから推定しなくてはなりません。

6.1.3 最尤推定

先の例では、$0.8 < x \leq 1.2$ のとき、$P(t=1|x)=1/3$ であることを、真の分布の情報から解析的に見積りました。しかし、実際には、この値はデータから推定すべきものです。

例えば、x が $0.8 < x \leq 1.2$ の範囲にある t に着目したら、はじめの3回は、$t=0$、4回目は $t=1$ だったとします。この情報から、$0.8 < x \leq 1.2$ での $P(t=1|x)$ を推定することを考えてみましょう。

まず、

$$P(t=1|x)=w \tag{6-3}$$

という単純なモデルを考えます。$t=1$ を確率 w で生成するというモデルです。w のとりうる範囲は、0から1までの間になります。そして、このモデルが、$\mathbf{T}=0, 0, 0, 1$ というデータを生成したとし、この情報から最も妥当な w を推定するという問題を考えます。

単純に考えてみれば、全部で4回のうち $t=1$ は1回しかないので、$w=1/4$ となりそうですが、他のモデルの場合でも対応できるように、少し一般的に、**最尤推定**（maximum likelihood）という方法で求めましょう。

まず、「クラスデータ $\mathbf{T}=0, 0, 0, 1$ がモデルから生成された確率」を考えます。この確率を**尤度**（likelihood）と呼びます。

例えば、w が0.1のときの尤度を求めてみましょう。$w=P(t=1|x)=0.1$ ですので、$t=1$ となる確率は0.1、$t=0$ となる確率は $1-0.1=0.9$ となります。よって、$\mathbf{T}$ が0、0、0、1となる確率、つまり尤度は、$0.9 \times 0.9 \times 0.9 \times 0.1 = 0.0729$ となります。

同じようにして、w が0.2のときの尤度も求めてみましょう。$w=P(t=1|x)=0.2$ ですので、$t=1$ となる確率は0.2、$t=0$ となる確率は $1-0.2=0.8$ です。よって、尤度は、$0.8 \times 0.8 \times 0.8 \times 0.2 = 0.1024$ となります。

$w=0.1$ のときの尤度は0.0729、$w=0.2$ のときの尤度は0.1024ということになりました。さて、$\mathbf{T}=0, 0, 0, 1$ というデータを生成したモデルのパラメータ w は、$w=0.1$ と $w=0.2$ のどちらかだとしたら、尤度の高い $w=0.2$ のほうがもっともらしいと言うことができます。$w=0.1$ であった場合も可能性としてはあるけれども、$w=0.2$ であった場合のほうが確率的に高いということです。

それでは、$w=0.1$ や0.2に限らず、0と1の間で最も尤度が高くなる w を解析的に求めてみましょう。$P(t=1|x)=w$ なので、$t=1$ となる確率は w、$t=0$ となる確率は $(1-w)$ です。よって、はじめの3回が $t=0$、4回目が $t=1$ となる確率、つまり、尤度は式6-4のように表すことができます。

$$P(\mathbf{T}=0,0,0,1|x)=(1-w)^3 w \tag{6-4}$$

0から1までの範囲で式6-4の値をグラフとして描くと、上向きの山のような形になります（図6.5下）。この山が最大値をとる w が最もありえた値であり、推定値として扱われます。これが最尤推定です。

問題

t は、0 か 1 の値をとるデータだとする。ある x の範囲に着目したとき、

「はじめの3回は $t=0$、4回目は $t=1$」、だったとすると、

$t=1$ となる確率は、どれくらいだったのだろう？

$P(t=1|x)=w$ として、w を求めよ。

考え方（最尤推定）

与えられた入力データ x に対して、ラベルデータ T が生成される確率（尤度）が一番大きくなる w を推定値とする。

解き方

$t=0$ となる確率は $(1-w)$、$t=1$ となる確率は w
$t=0$ が3回、$t=1$ が1回出る確率（尤度）は、

$$P(\mathbf{T}=0,0,0,1|x)=(1-w)(1-w)(1-w)w$$

これが最大になる w を求めればいい。
答えは $w=0.25$

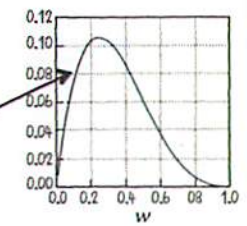


図 6.5：最尤推定の考え方

それでは、式 6-4 が最大値をとる w を求めてみましょう。まず、式 6-4 のような掛け算の連続を扱うのは大変なので、両辺の対数をとります（式 6-5）。対数をとると、掛け算が足し算になり計算が楽になります（4.7 節「指数関数と対数関数」を参照）。

$$\log P=\log\{(1-w)^3w\}=3\log(1-w)+\log w \tag{6-5}$$

対数は単調増加の関数なので、P を最大にする w と、$\log P$ を最大にする w は変わりません（4.7 節「指数関数と対数関数」を参照）。つまり、$\log P$ を最大にする w を求めれば、その w は、P も最大にすることになります。

対数をとった尤度を**対数尤度**（log likelihood）と呼び、これこそが、平均二乗誤差関数に代わる、確率を取り入れた世界での目的関数になります。平均二乗誤差関数のときは、それを最小化するパラメータを探しましたが、対数尤度の場合は最大化するパラメータを探すことになります。

最大となるパラメータを求めるときもこれまでと方法は同じです。パラメータで目的関数（対数尤度）を微分し（4.7.4 項「対数関数の微分」を参照）、イコール 0 とおいた方程式を解いていきます（式 6-6）。

$$\frac{d}{dw}\log P=\frac{d}{dw}[3\log(1-w)+\log w]=0$$
$$3\frac{-1}{1-w}+\frac{1}{w}=0 \tag{6-6}$$
$$\frac{-3w+1-w}{(1-w)w}=0$$

$0<w<1$ の範囲で解を考えれば分母は 0 にならないので、両辺に $(1-w)w$ を掛けて、式 6-7 を得ます。

$$-3w+1-w=0 \tag{6-7}$$

上記を解くと式 6-8 のようになります。

$$w=\frac{1}{4} \tag{6-8}$$

予想通りの値が得られました。つまりは、データ $\mathbf{T}=0、0、0、1$ が最も生成されうるモデルのパラメータは $w=1/4$ であり、これが w の最尤推定値となります。

これで、うまくデータからパラメータを推定することができました。しかし、まだ実践的ではありません。というのも、x が $0.8<x\leq 1.2$ の範囲にあるときに確率は一定であるという知識を使っていたからです。実際には、確率が一定となる範囲はわかりませんし、そもそも、確率が一定になるという区間は存在しないかもしれません。

6.1.4 ロジスティック回帰モデル

ここまでは、データを一様分布から生成されたものとして考えてきました。そのおかげで、$P(t=1|x)$ が理解しやすい階段状の分布になっていました。しかし、実際のデータが一様分布となることはあまりありません。例えば、体重や身長のばらつきは、ガウス分布でよく近似できることがわかっています。

そこで、人工で作った質量のデータは簡単のために一様分布から生成させていますが、あえて、ガウス分布に従っていると仮定して議論を進めることにします。この仮定のもとだと、条件付き確率 $P(t=1|x)$ は、**ロジスティック回帰モデル**で表せるこ

とがわかっています（参考文献『パターン認識と機械学習　上』、C.M. ビショップ著、元田 浩、栗田 多喜夫、樋口 知之、松本 裕治、村田 昇 監訳、丸善出版、2012 年 4 月、第 4 章を参照）。

ロジスティック回帰モデルは、以下の直線の式 6-9 を、式 6-10 のようにシグモイド関数 $\sigma(x)=1/\{1+\exp(-x)\}$ (4.7.5 項「シグモイド関数」を参照）の中に入れた形になっています。

$$y = w_0 x + w_1 \tag{6-9}$$

$$y = \sigma(w_0 x + w_1) = \frac{1}{1 + \exp\{-(w_0 x + w_1)\}} \tag{6-10}$$

こうすることで、直線モデルの大きい正の出力は 1 に近い値に、絶対値の大きい負の出力は 0 に近い値に変換され、結果、直線の関数は、0 と 1 の間に押し込められることになります（図 6.6）。

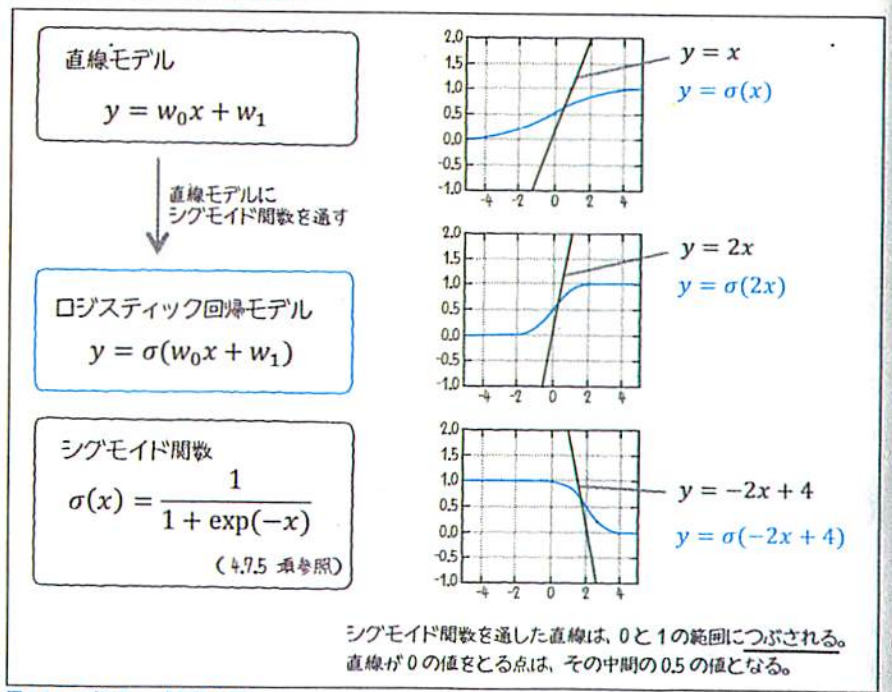


図 6.6：ロジスティック回帰モデル

それではプログラムです。リスト 6-1-(3) でロジスティック回帰モデルを定義します。

In

```
# リスト 6-1-(3)
def logistic(x, w):
    y = 1 / (1 + np.exp(-(w[0] * x + w[1])))
    return y
```

それを表示する関数を次のリスト 6-1-(4) で作っておきます。実行すると、ロジスティック回帰モデルが、決定境界とともに表示され、決定境界の値が出力されます。

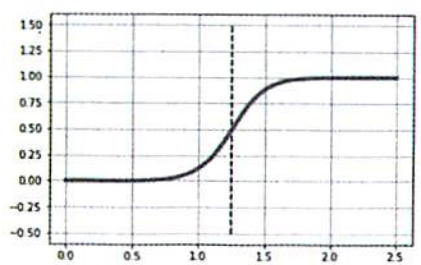
In

```
# リスト 6-1-(4)
def show_logistic(w):
    xb = np.linspace(X_min, X_max, 100)
    y = logistic(xb, w)
    plt.plot(xb, y, color='gray', linewidth=4)
    # 決定境界
    i = np.min(np.where(y > 0.5))  # (A)
    B = (xb[i - 1] + xb[i]) / 2    # (B)
    plt.plot([B, B], [-.5, 1.5], color='k', linestyle='--')
    plt.grid(True)
    return B


# test
W = [8, -10]
show_logistic(W)
```

Out

```
1.25
```



リスト 6-1-(4) の (A) と (B) の部分で決定境界を求めていますが、補足説明です。決定境界は $y = 0.5$、となる x の値です。(A) の `np.where(y > 0.5)` は、`y > 0.5` を満たす要素番号をすべて返すという命令文です。`i = np.min(np.where(y > 0.5))` とすることで、`y > 0.5` を満たす要素番号の中で一番小さいインデックスが `i` に入ります。つまり、`i` は `y` が `0.5` を超えた直後の要素番号です。

そして、(B) の `B = (xb[i - 1] + xb[i]) / 2` で、`y` が `0.5` を超えた直後の `xb[i]` と、その直前の `xb[i - 1]` の中点が決定境界の近似値として `B` に格納されます。

6.1.5 交差エントロピー誤差

ロジスティック回帰モデルを使って、x が $t = 1$ となる確率を式 6-11 のように表します。

$$y = \sigma(w_0 x + w_1) = P(t = 1|x) \tag{6-11}$$

それでは、このパラメータ w_0 と w_1 が虫のデータに合うように最尤推定しましょう。「このモデルから虫のデータが生成されたとして、最もありえる（確率的に高い）パラメータを求める」という方針です。前節では、特定のデータ 4 つ（T = 0、0、0、1 でした）に対して最尤推定を試みましたが、ここではどんなデータにも対応できるように考えていきます。

まず、虫のデータがこのモデルから生成された確率、尤度を求めます。データが 1 つだけだとし、ある体重 x に対して $t = 1$ だったら、$t = 1$ がモデルから生成される確率は、ロジスティック回帰モデルの出力値 y そのものです。逆に、$t = 0$ だったら $1 - y$ となります。

この生成確率が t の値によって y や $1 - y$ と変わってしまうのは、一般的なデータに対して考えるとなると不便です。そこで、数学的なトリックを使って、クラスの生成確率を式 6-12 のように表します。

$$P(t|x) = y^t(1 - y)^{1-t} \tag{6-12}$$

突然、複雑になった感じがしますが、大丈夫です。$t = 1$ のときは、式 6-13 のようになります。

$$P(t = 1|x) = y^1(1 - y)^{1-1} = y \tag{6-13}$$

$t = 0$ のときは、式 6-14 のようになるので、$t = 1$ のときでも $t = 0$ のときでも、式 6-12 で $P(t|x)$ を表せることがわかります。指数をスイッチのように使っているのです。

$$P(t = 0|x) = y^0(1 - y)^{1-0} = 1 - y \tag{6-14}$$

それでは、データが N 個だったら、与えられた $\mathbf{X} = x_0, \cdots, x_{N-1}$ に対して、クラス $\mathbf{T} = t_0, \cdots, t_{N-1}$ の生成確率はどうなるでしょうか？ 1 つ 1 つのデータの生成確率をすべてのデータで掛け算すればよいので、式 6-15 のようになります。これが尤度です。

$$P(\mathbf{T}|\mathbf{X}) = \prod_{n=0}^{N-1} P(t_n|x_n) = \prod_{n=0}^{N-1} y_n^{t_n}(1 - y_n)^{1-t_n} \tag{6-15}$$

式 6-15 の対数をとって、対数尤度を得ます。パラメータ w_0、w_1 は、この対数尤度が最大になるように求めればよいことになります（式 6-16）。

$$\log P(\mathbf{T}|\mathbf{X}) = \sum_{n=0}^{N-1} \{t_n \log y_n + (1 - t_n)\log(1 - y_n)\} \tag{6-16}$$

式 6-15 から式 6-16 の変形には、公式 4-108 を使いました。第 5 章までは、平均二乗誤差が最小になるようにパラメータを求めていましたので、それと合わせるために、式 6-16 に −1 を掛けたものを考えます。これを、**交差エントロピー誤差**（cross-entropy error function）と呼びます。これなら、これまでの平均二乗誤差と同じく、誤差が最小になるようにパラメータを求めればよいことになります。そして、交差エントロピー誤差を N で割った、**平均交差エントロピー誤差** を $E(\mathbf{w})$ として定義します（式 6-17）。このほうがデータ数に誤差の値が影響されにくく、数値を調べるには都合がよいからです。

$$E(\mathbf{w}) = -\frac{1}{N}\log P(\mathbf{T}|\mathbf{X}) = -\frac{1}{N}\sum_{n=0}^{N-1} \{t_n \log y_n + (1 - t_n)\log(1 - y_n)\} \tag{6-17}$$

それでは、リスト 6-1-(5) で、平均交差エントロピー誤差を計算する関数、**cee_logistic(w, x, t)** を作ります。

In

```
# リスト 6-1-(5)
# 平均交差エントロピー誤差 --------------------
def cee_logistic(w, x, t):
    y = logistic(x, w)
    cee = 0
    for n in range(len(y)):
        cee = cee - (t[n] * np.log(y[n]) + (1 - t[n]) * np.log(1 - y[n]))
    cee = cee / X_n
    return cee

# test
W=[1,1]
cee_logistic(W, X, T)
```

Out

```
1.0288191541851066
```

最後の行で $w_0 = 1$、$w_1 = 1$ として関数を実行してみると、それらしい値が返ってきました。

それでは、この平均交差エントロピー誤差がどのような形をしているのか、その形を次のリスト 6-1-(6) で確かめてみましょう。

In

```
# リスト 6-1-(6)
from mpl_toolkits.mplot3d import Axes3D

# 計算 ----------------------------------
wn = 80  # 等高線表示の解像度
w_range = np.array([[0, 15], [-15, 0]])
w0 = np.linspace(w_range[0, 0], w_range[0, 1], wn)
w1 = np.linspace(w_range[1, 0], w_range[1, 1], wn)
ww0, ww1 = np.meshgrid(w0, w1)
C = np.zeros((len(w1), len(w0)))
w = np.zeros(2)
for i0 in range(wn):
    for i1 in range(wn):
        w[0] = w0[i0]
        w[1] = w1[i1]
        C[i1, i0] = cee_logistic(w, X, T)

# 表示 ----------------------------------
plt.figure(figsize=(12, 5))
plt.subplots_adjust(wspace=0.5)
ax = plt.subplot(1, 2, 1, projection='3d')
ax.plot_surface(ww0, ww1, C, color='blue', edgecolor='black',
                rstride=10, cstride=10, alpha=0.3)
ax.set_xlabel('$w_0$', fontsize=14)
ax.set_ylabel('$w_1$', fontsize=14)
ax.set_xlim(0, 15)
ax.set_ylim(-15, 0)
ax.set_zlim(0, 8)
ax.view_init(30, -95)

plt.subplot(1, 2, 2)
cont = plt.contour(ww0, ww1, C, 20, colors='black',
                   levels=[0.26, 0.4, 0.8, 1.6, 3.2, 6.4])
cont.clabel(fmt='%.1f', fontsize=8)
plt.xlabel('$w_0$', fontsize=14)
plt.ylabel('$w_1$', fontsize=14)
plt.grid(True)
plt.show()
```

Out

```
# 実行結果は図 6.7 を参照
```

リスト 6-1-(6) を実行すると図 6.7 に示したグラフが表示されます。

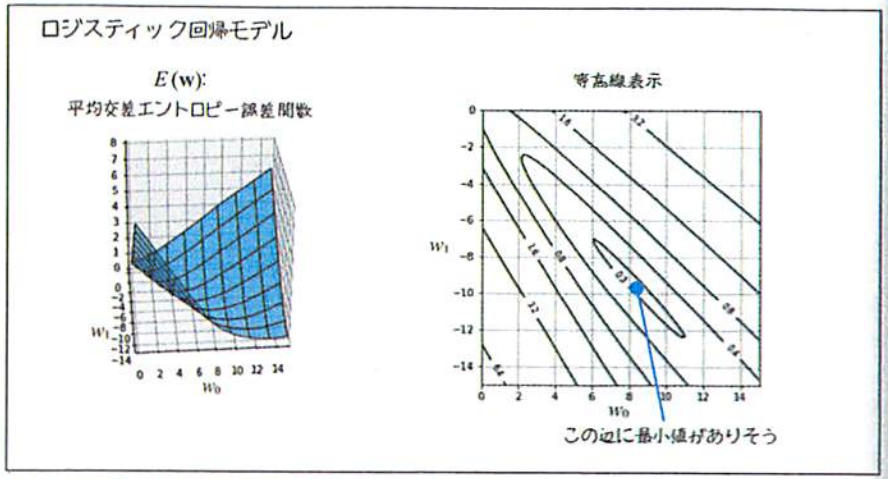


図 6.7：ロジスティック回帰モデルの平均交差エントロピー誤差関数

平均交差エントロピー誤差関数は、風呂敷の対角の隅を持って持ち上げたような形をしていました。どうやら最小値は、$w_0 = 9$、$w_1 = -9$ の付近にありそうです。

6.1.6 学習則の導出

さて、交差エントロピー誤差が最小になるパラメータの解析解は求めることができません。y_n が非線形のシグモイド関数を含んでいるからです。そこで、勾配法を使って数値的に求めることを考えます。勾配法を使うには、パラメータの偏微分が必要でした。

それでは、式 6-17 の平均交差エントロピー誤差 $E(\mathbf{w})$ を w_0 で偏微分したものを求めていきましょう。まず、式 6-17 を式 6-18 のように表します。

$$E(\mathbf{w}) = \frac{1}{N}\sum_{n=0}^{N-1} E_n(\mathbf{w}) \tag{6-18}$$

上記の和の中身は式 6-19 のように定義しました。

$$E_n(\mathbf{w}) = -t_n \log y_n - (1 - t_n)\log(1 - y_n) \tag{6-19}$$

微分と和は交換できることから（4.5 節「偏微分」を参照）、式 6-18 から式 6-20 を得ることができます。

$$\frac{\partial}{\partial w_0}E(\mathbf{w}) = \frac{1}{N}\frac{\partial}{\partial w_0}\sum_{n=0}^{N-1} E_n(\mathbf{w}) = \frac{1}{N}\sum_{n=0}^{N-1}\frac{\partial}{\partial w_0}E_n(\mathbf{w}) \tag{6-20}$$

そこで、和の記号の中身 $\frac{\partial}{\partial w_0}E_n(\mathbf{w})$ を求めてから、最後にその平均を計算し、$\frac{\partial}{\partial w_0}E(\mathbf{w})$ を求めるという作戦を考えます。

さて、$E_n(\mathbf{w})$ の中身（式 6-19）の y_n は、ロジスティック回帰モデルの出力ですが、後々の計算のために、シグモイド関数の中身 $w_0x_n + w_1$ を a_n で表すことにします（式 6-21、式 6-22）。この a_n を**入力総和**と呼ぶことにします。

$$y_n = \sigma(a_n) = \frac{1}{1 + \exp(-a_n)} \tag{6-21}$$

$$a_n = w_0x_n + w_1 \tag{6-22}$$

すると、$E_n(\mathbf{w})$ は $E_n(y_n(a_n(\mathbf{w})))$ と、入れ子の関数として解釈することができますので、w_0 で偏微分するために、4.4.4 項で解説した連鎖律の公式を使います（式 6-23）。

$$\frac{\partial E_n}{\partial w_0} = \frac{\partial E_n}{\partial y_n}\cdot\frac{\partial y_n}{\partial a_n}\cdot\frac{\partial a_n}{\partial w_0} \tag{6-23}$$

式 6-23 の右辺の 3 つのパーツのはじめのパーツは、式 6-19 を y_n で偏微分したものです（式 6-24）。

$$\frac{\partial E_n}{\partial y_n} = \frac{\partial}{\partial y_n}\{-t_n \log y_n - (1 - t_n)\log(1 - y_n)\} \tag{6-24}$$

上記の偏微分の記号を y_n に関係する部分だけに作用させます（式 6-25）。

$$= -t_n\frac{\partial}{\partial y_n}\log y_n - (1 - t_n)\frac{\partial}{\partial y_n}\log(1 - y_n) \tag{6-25}$$