最適化と認識・学習

2019 年 11 月 13 日

1　導入

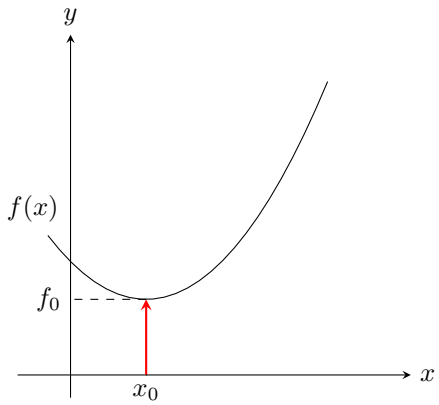
最適化とは

$$\Psi(\theta) \ : \ \text{目的関数}$$
$$\theta \ : \ \text{パラメータ}$$

としたとき, $\Psi(\theta)$ を θ で最小化 (あるいは最大化) することである. つまり

関数の最小値, 最大値を与える x をもとめる問題

ということである.



このときの

$$f_0 = \min_x f(x)$$
$$x_0 = \operatorname*{argmin}_x f(x)$$

を求める問題である. θ が関数の形状を決めるパラメータであったとき, 目的関数 $\Psi(\theta)$ は近似誤差を表す関数となり, 最適化は関数 $f(x;\theta)$ の形状を求める問題といえる.

2　回帰/分類とニュートン法, 勾配法

2.1　予測モデル

予測モデルを入力 x, パラメタ Θ として

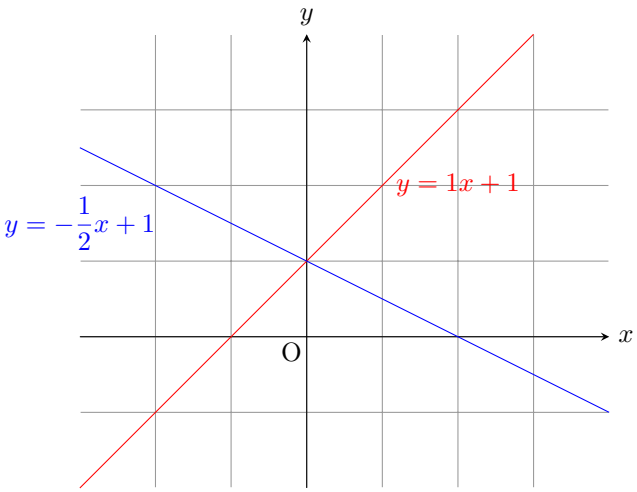
$$y = f(x;\Theta)$$

と表す.
このとき予測モデルの入力は説明変数 (explanatory variable), 独立変数 (independent variable), 予測変数 (predictor) に分類され, それに応じて出力 y は目的変数 (objective variable), 従属変数

(dependent variable), 応答変数 (response variable) となる.
たとえば, 予測モデル $y = f(x;\Theta)$ に対して,

$$y = ax + b, \quad \Theta = (a, b)$$

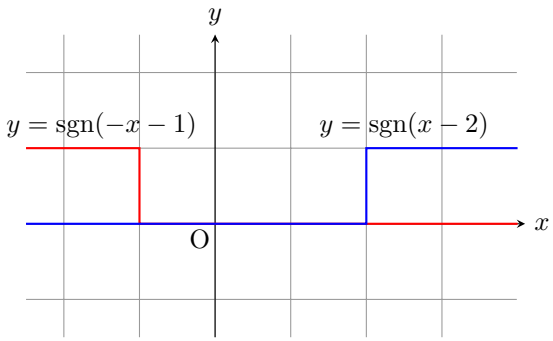
とおくと予測モデルは次のようになる.



この場合の y は量的データのときの予測問題であり回帰 (regression) と呼ぶ.
また,

$$y = \mathrm{sgn}(ax + b), \quad \Theta(a, b)$$

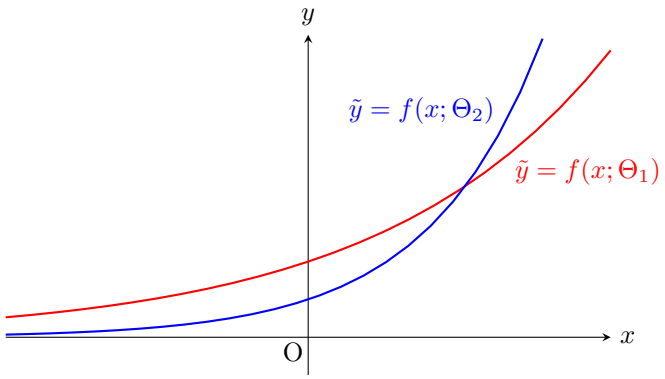
とおくと予測モデルは次のようになる.



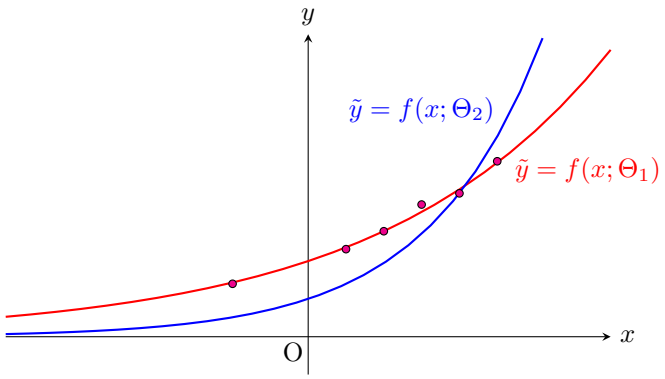
この場合の y の質的データのときの予測問題であり分類/識別 (Classification) と呼ぶ. 質的データとはカテゴリ, 離散データ, クラスといったものである. 大人を 1, 小人を 0 として人数を分類したものも質的データにあたる.

2.2 教師つき学習

出力予測モデルである y は入力 x とパラメータ Θ によって変わる. したがって, $Theta$ が異なると様々な予測モデルができる.



このようなモデルに対して学習データ (training data) を用いてパラメタ Θ を推定することを教師つき学習という.



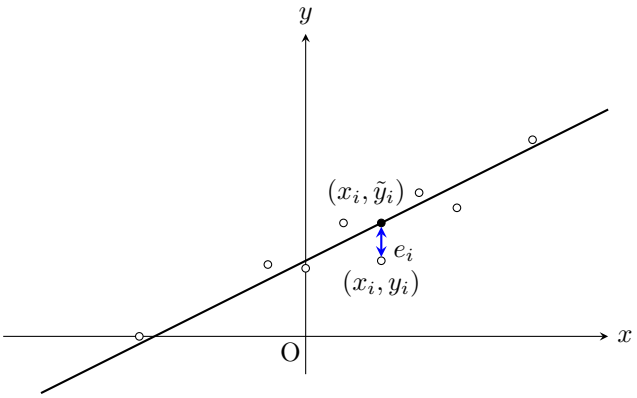
これより教師つき学習をすることによって, 今回の例ではモデルは

$$\tilde{y} = f(x;\Theta_1)$$

であることが推定される.

2.3 線形回帰

線形回帰とは与えられたデータに対し, 線形関数を当てはめる問題である.



すると推測される関数は以下のように表すことができる.

$$\tilde{y}_i = f(x_i;a,b) = ax_i + b$$

このとき, 誤差 e_i は

$$e_i = |\tilde{y}_i - y_i|$$

となるので, これの二乗したときの最小を求めれば最適化される. つまり

$$\min_{a,b} \sum_{i=1}^{N} e_i^2 = \min_{a,b} \sum_{i=1}^{N} (\tilde{y}_i - y_i)^2 \tag{2.1}$$

を求めればよい. これを求める方法を最小二乗法である.

(x_n, y_n) : n 番目の学習データ

$\boldsymbol{x}_n = \begin{bmatrix} x_n & 1 \end{bmatrix}^T$: x_n を次元拡張したベクトル

$D = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_n & \cdots \end{bmatrix}^T = \begin{bmatrix} d_{ij} \end{bmatrix}, \ (d_{i1} = x, \ d_{i2} = 1)$

$\boldsymbol{w}^T = \begin{bmatrix} a & b \end{bmatrix}$

$\tilde{y}_n = ax_n + b = \begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} x_n \\ 1 \end{bmatrix} = \boldsymbol{w}^T \boldsymbol{x}_n = \sum_{k=1}^{2} w_k \cdot d_{nk}$

$\tilde{\boldsymbol{y}} = D\boldsymbol{w}$

$$\boldsymbol{e} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \\ \vdots \end{bmatrix} = \begin{bmatrix} e_n \end{bmatrix}, \quad e_n = \tilde{y}_n - y$$

ここで誤差 E は二乗誤差の $\dfrac{1}{2}$ 倍したものと定義し,

$$\begin{aligned} E &= \frac{1}{2} \boldsymbol{e}^T \boldsymbol{e} = \frac{1}{2} (\tilde{\boldsymbol{y}} - \boldsymbol{y})^T (\tilde{\boldsymbol{y}} - \boldsymbol{y}) \\ &= \frac{1}{2} (D\boldsymbol{w} - \boldsymbol{y})^T (D\boldsymbol{w} - \boldsymbol{y}) \end{aligned}$$

ここで誤差が最小となるときは $\dfrac{\partial E}{\partial \boldsymbol{w}} = 0$ となるときである.

ここで

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{x}^T \boldsymbol{a} &= \boldsymbol{a} \\ \frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{a}\boldsymbol{x} &= \boldsymbol{a}^T \\ \frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{x}^T \boldsymbol{x} &= 2\boldsymbol{x} \end{aligned}$$

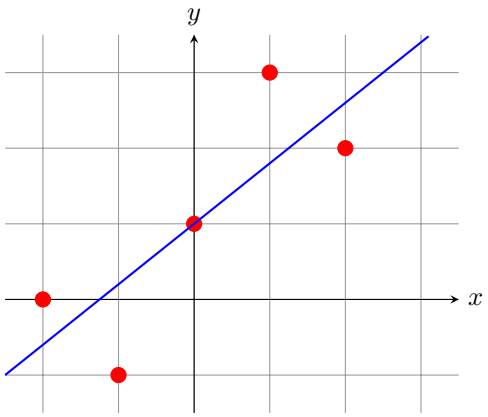
という性質を使えば微分をすることができて, 微分する項を展開して微分すると以下のようになる.

$$\begin{aligned} \frac{\partial E}{\partial \boldsymbol{w}} &= \left(\frac{1}{2} (D\boldsymbol{w} - \boldsymbol{y})^T (D\boldsymbol{w} - \boldsymbol{y}) \right)' \\ &= \frac{1}{2} \left(\boldsymbol{w}^T D^T D\boldsymbol{w} - \boldsymbol{w}^T D^T \boldsymbol{y} - \boldsymbol{y}^T D\boldsymbol{w} + \boldsymbol{y}^T \boldsymbol{y} \right)' \\ &= \frac{1}{2} \left(D^T D\boldsymbol{w} + (\boldsymbol{w}^T D^T D)^T - D^T \boldsymbol{y} - (\boldsymbol{y}^T D)^T \right) \\ &= \frac{1}{2} (D^T D\boldsymbol{w} + D^T D\boldsymbol{w}^T - D^T \boldsymbol{y} - D^T \boldsymbol{y}) \\ &= D^T D\boldsymbol{w} - D^T \boldsymbol{y} \end{aligned}$$

であるから,

$$\begin{aligned} & \frac{\partial E}{\partial \boldsymbol{w}} = D^T D\boldsymbol{w} - D^T \boldsymbol{y} = 0 \\ \iff & D^T D\boldsymbol{w} = D^T \boldsymbol{y} \\ \iff & \boldsymbol{w} = \left(D^T D \right)^{-1} D^T \boldsymbol{y} \end{aligned}$$

となる. $(-2, 0)$, $(-1, -1)$, $(0, 1)$, $(1, 3)$, $(2, 2)$ のとき, 最小二乗法を用いて線形回帰モデルを求める.



python での実装方法は以下である.

```
import numpy as np
from scipy import linalg as la

x = np.matrix([[-2,1],[-1,1],[0,1],[1,1],[2,1]]) # 本物のD
y = np.matrix([0,-1,1,3,2])
y = y.T # 本来のベクトルy
D = x.T*x #D^T*を表すD
w = la.inv(D)*x.T*y

print(w)
```

説明変数が多変数の場合も同様にして考えることができ, 以下のようになる.

$(x_{n1}, x_{n2}, \cdots, x_{np}, y_n)$: n 番目の学習データ

$\boldsymbol{x}_n = \begin{bmatrix} x_{n1} & x_{n2} & \cdots & x_{np} & 1 \end{bmatrix}^T$

$D = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_i & \cdots & \boldsymbol{x}_N \end{bmatrix}^T$: データ行列

$\boldsymbol{w} = \begin{bmatrix} a_1 & a_2 & \cdots & a_p & b \end{bmatrix}^T$: パラメタ

$\boldsymbol{y} = D\boldsymbol{w}$: 予測式

$\boldsymbol{w} = (D^T D)^{-1} D^T \boldsymbol{y}$

以下の 12 点から線形回帰モデルを最小二乗基準で求める.

(0,0,1), (0,1,3), (0,2,2), (0,3,3), (1,0,1), (1,1,4), (1,2,5), (1,3,4), (2,0,4), (2,1,5), (2,2,4), (2,3,7)

python での実行すると

$$\tilde{\boldsymbol{y}} = 1.375x_1 + 0.76666667x_2 + 1.05833333$$

となる.

一方で次の 9 点から, 線形回帰モデルを最小二乗基準で求める.

(0,0,1), (1,2,3), (2,4,2), (3,6,2), (4,8,3), (5,10,1), (6,12,4), (7,14,5), (8,16,4)

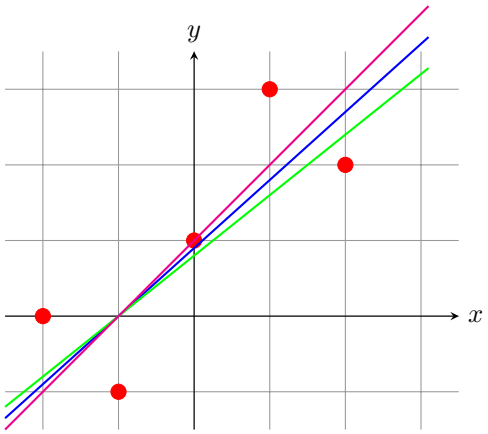
これは逆行列が存在しないので, 最小二乗法での解は存在しない.

2.4　リッジ回帰

パラメタ同士に相関がある場合, 係数は安定に定まらない. このとき, 回帰係数のノルムに応じた罰則項を加える. これをリッジ回帰といい以下の式で表される.

$$\min\left\{(D\boldsymbol{w}-\boldsymbol{y})^T(D\boldsymbol{w}-\boldsymbol{w})+\lambda\sum_{i=1}^{p}w_i^2\right\} \tag{2.2}$$

パラメタ変数に相関がある、例えば、パラメータ $\boldsymbol{w}=[a\,b]^T$ に関して $a=b$ という相関があった場合、$\boldsymbol{w}=[a\,a]^T$ のように表される。この時先の例題と同じものを用いて、教師データを $(-2,0),\ (-1,-1),\ (0,1),\ (1,3),\ (2,2)$ とするとき, 最小二乗法を用いて線形回帰モデルを求める。相関がない場合は先の回答のグラフから、$\boldsymbol{W}=[0.8\,1]^T$ であったが、この解は $a=b$ という関係を満たさない。よって、この相関関係が成り立つと仮定すると、最小の誤差を求めることができず、最小に近い近似の誤差を求めることになるため、下図のように複数の回答が存在する。



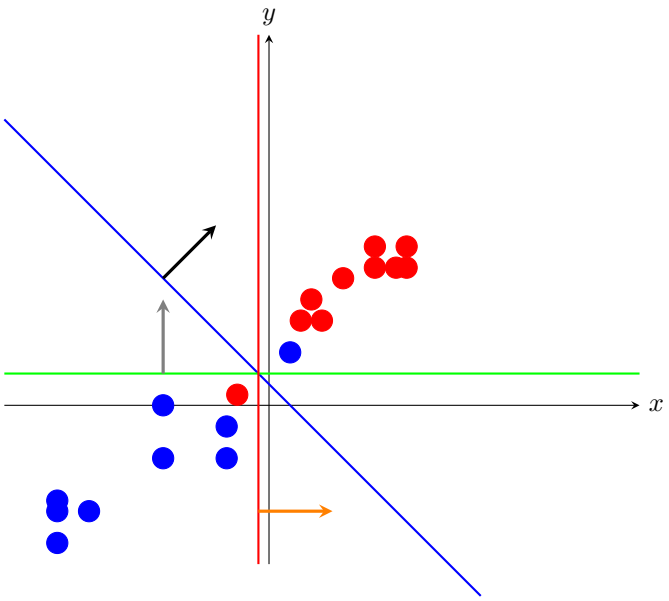
そのため、元の誤差の式だけでは、収束値が一定しない可能性があるが、一つの考え方として、パラメータベクトルの長さが一番最小となるとき、最小の誤差であると考えると、罰則項として、ノルムを加えることで収束先が一定し、最小誤差が特定できるようにするというものである。
このようにして、収束しない可能性のある目的関数を収束するような関数に変換することを**正則化**という

2.5　ロッソ回帰

この場合はデータ自体に相関が存在すると、パラメタにも相関が存在し、パラメタを減らすことにより、パラメタ間の相関関係が解決し、最適解が収束する場合がある。この時、このパラメタを減らすためにパラメタの絶対値の罰則項を付加することで、軸上に収束しやすくなり、パラメータを減らす収束値となる可能性がある正則化ができる。この回帰をロッソ回帰といい、次の式で表される。

$$\min\left\{(D\boldsymbol{w}-\boldsymbol{y})^T(D\boldsymbol{w}-\boldsymbol{w})+\lambda\sum_{i=1}^{p}|w_i|\right\} \tag{2.3}$$

また、これの場合は以下のようなデータがあり、以下のように通常なら、青色の線で識別するが、データ間に相関があるため、パラメタを一つ減らして、緑の線で識別できたり、赤色の線で識別できたりする。

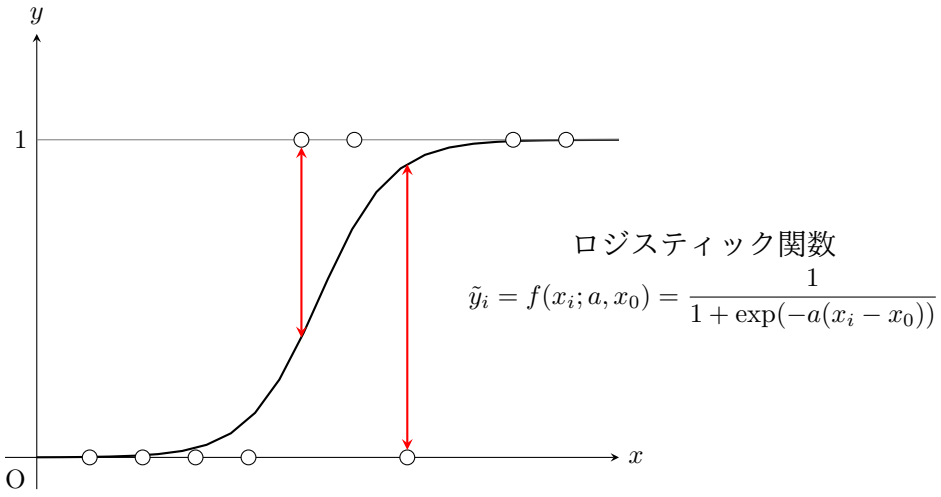


この場合はパラメタとして $\boldsymbol{x}, \boldsymbol{y}$ を使う必要はなく、$\boldsymbol{x}$ だけでも識別可能な平面のパラメタとしては十分である場合がある。そのため、このことを考慮に入れた最適化を行うために罰則項としてパラメタの絶対値を付加し、軸上に収束しやすくなる。

3　ロジスティック回帰と勾配法

3.1　ロジスティック回帰

ロジスティック回帰とは与えられたデータに対し, ロジスティック関数を当てはめる問題のことである.

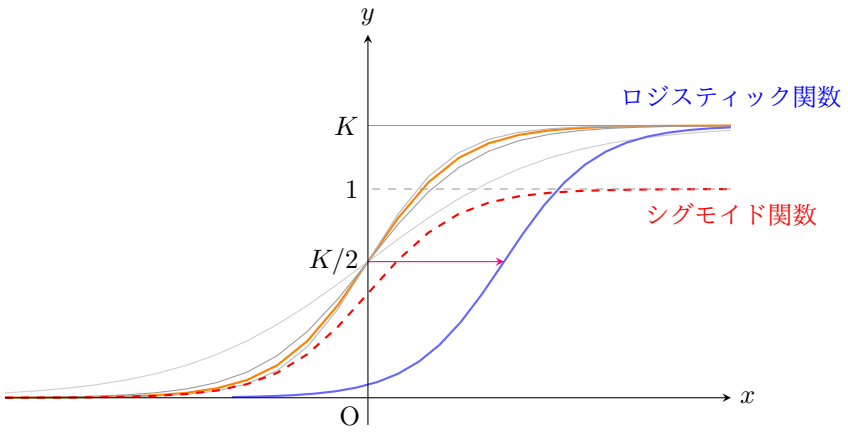


上のロジスティック関数は関数の値が $y_i = \{0, 1\}$ のときに用いられる.
以下のように識別機の学習を適用することが出来る.

$$\text{クラス } A \Leftrightarrow y_i = 0,\ \text{クラス } B \Leftrightarrow y_i = 1;$$
$$\text{if}(f(x) < 0.5) \text{ then クラス } A;\ \text{else クラス } B$$

一般にロジスティック関数は以下で定式化される.

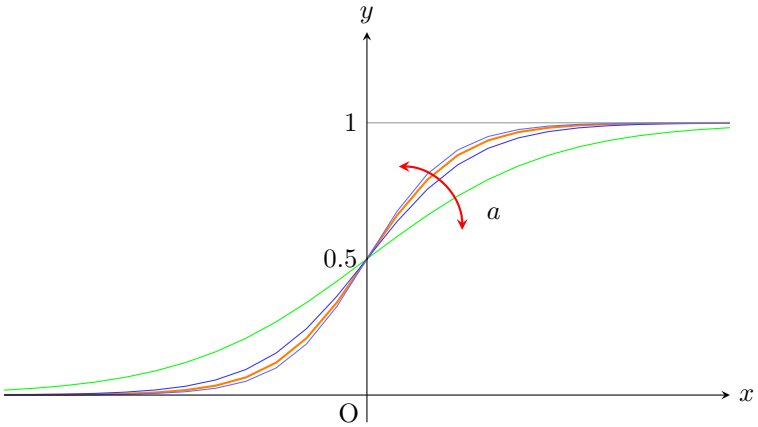
$$\frac{dN}{dt} = a\left(\frac{K-N}{K}\right)N$$
$$\Longleftrightarrow N(x) = \frac{K}{1+\exp(-aK(x-x_0))} \tag{3.1}$$



シグモイド関数は, ロジスティック関数の $K = 1$, $x_0 = 0$ の特殊な場合に相当する.
(標準) シグモイド関数は以下の式で表すことができる.

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{3.2}$$

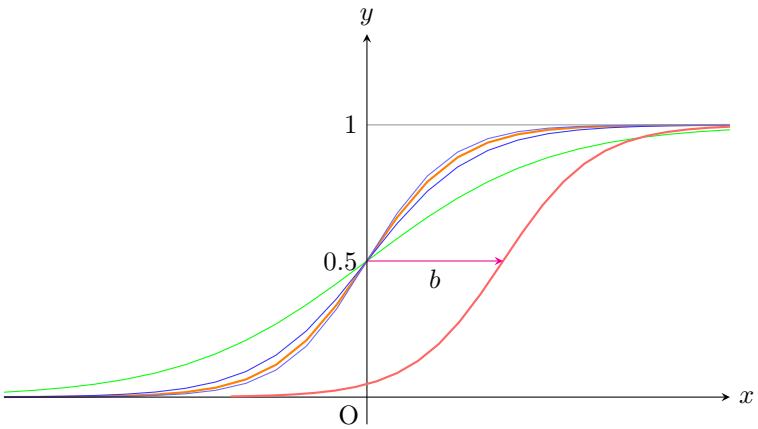
シグモイド関数において, a の値によって, 関数の傾き度合いが変わる.



シグモイド関数を x 軸に b だけシフトしたときの関数は以下のように表すことが出来る.

$$f(x) = \frac{1}{1 + \exp(-a(x - b))}$$

すると以下のようにグラフは描くことが可能となる.



シグモイド関数の重要な性質として微分が元の関数を用いて表現することができる.

$$\begin{aligned} f'(x) &= \left(\frac{1}{1+\exp(-x)}\right)' \\ &= \frac{\exp(-x)}{(1+\exp(-x))^2} \\ &= \frac{1}{1+\exp(-x)} \cdot \frac{\exp(-x)}{1+\exp(-x)} \\ &= \frac{1}{1+\exp(-x)} \cdot \frac{1+\exp(-x)-1}{1+\exp(-x)} \\ &= \frac{1}{1+\exp(-x)} \cdot \left(1 - \frac{1}{1+\exp(-x)}\right) \\ &= f(x)\left(1-f(x)\right) \end{aligned}$$

また, $f'''(x)$ については

$$\begin{aligned} f''(x) &= \left(f'(x)\right)' \\ &= \left(f(x)(1-f(x))\right)' \\ &= f'(x)(1-f(x)) + f(x)(1-f(x))' \\ &= f'(x)(1-f(x)) - f(x)f'(x) \\ &= f'(x)(1-2f(x)) \\ &= f(x)(1-f(x))(1-2f(x)) \end{aligned}$$

よって, まとめると

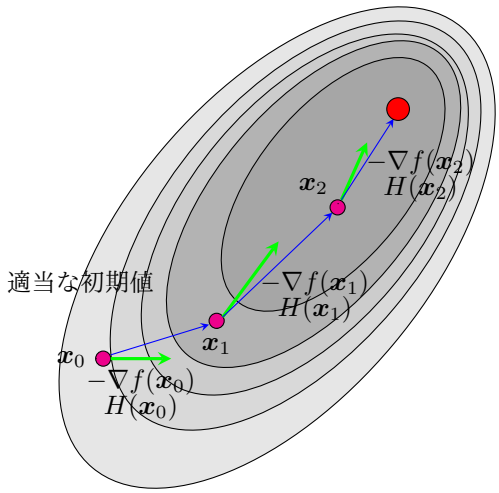
$$\begin{aligned} f(x) &= \frac{1}{1+\exp(-x)} \\ f'(x) &= f(x)(1-f(x)) \\ f''(x) &= f(x)(1-f(x))(1-2f(x)) \end{aligned} \tag{3.3}$$

(x_n, y_n) : n 番目の学習データ
$\boldsymbol{x}_n = \begin{bmatrix} x_n & 1 \end{bmatrix}^T$: x_n を次元拡張してベクトル化したもの
$D = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_n & \cdots \end{bmatrix}^T = \begin{bmatrix} d_{ij} \end{bmatrix}$, $(d_{i1} = x_i,\ d_{i2} = 1)$
$z_n = \boldsymbol{w}^T \boldsymbol{x}_n = \sum_k w_k \cdot d_{nk}$
ここで線形回帰の場合は $\tilde{y}_n = f(x_i; a, b) = ax_i + b$ としていたが, ロジスティック回帰では
$\tilde{y}_n = f(z_n) = \dfrac{1}{1+\exp(-z_n)}$
$E = \dfrac{1}{2}\sum_{n=1}^{N}(y_n - \tilde{y}_n)^2 = \dfrac{1}{2}\sum_{n=1}^{N}\left(y_n - \dfrac{1}{1+\exp(-z_n)}\right)^2$

これから誤差 E が最小となるように計算していくが, ロジスティック回帰の場合は解が解析的に求まらない (微分方程式の解が定まらない) ので, 繰り返し計算が必要となる.
ここで, **勾配法**というものを導入する.

3.2 勾配法

勾配法は近似解を与え, 近似解の周辺の傾き等を用いて, 近似解を漸次改良し, この処理をくり返す方法である.
(例) 最急降下法, 共役勾配法, ニュートン法



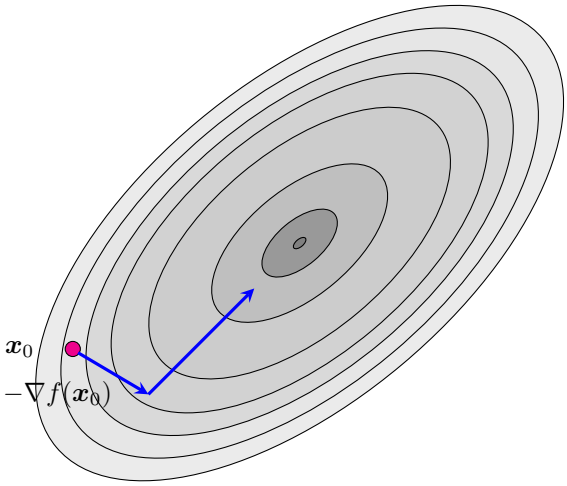
このように近似解を繰り返し修正することで最適値へ近づいていく．勾配 (, ヘシアン), つまり E の $\boldsymbol{x}_0$ における 1 次 (,2 次) の微分) 施すことで修正を行う．
問題としては近似解の精度を上げるためにどちらの方向にどれだけ動かすか？ということである．

3.2.1　最急降下法

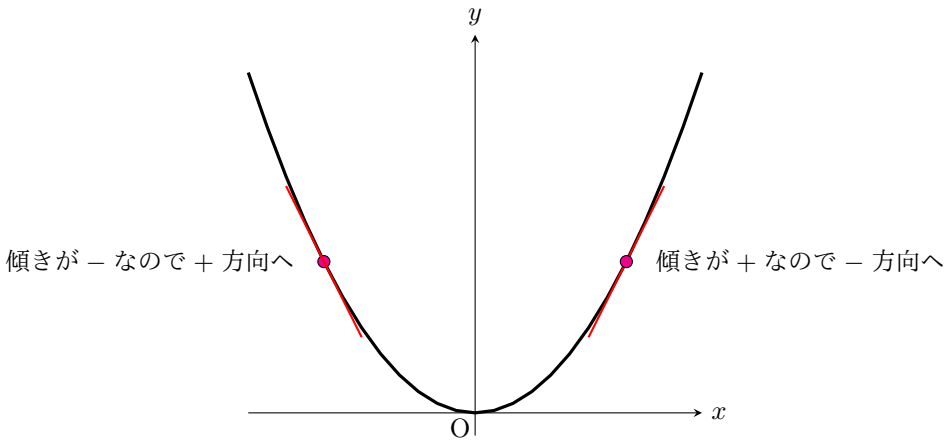
最急降下法は最も単純なもので勾配の方向に適当量移動させるという方法である．式にすると以下のように表すことができる．

$$x_{i+1} = x_i - \alpha \nabla f(\boldsymbol{x}_i) \tag{3.4}$$

イメージ図としては以下のようになる．



誤差が最小となるところに持っていく．ということは誤差関数において値が最小となるように持ってくることである．つまり誤差関数の現時点にいる場所から傾きを得る．それが負であれば + にずらし, 正なら − にずらすことで最小値に近づけることが可能となる．誤差関数が二次関数においての例は以下である．



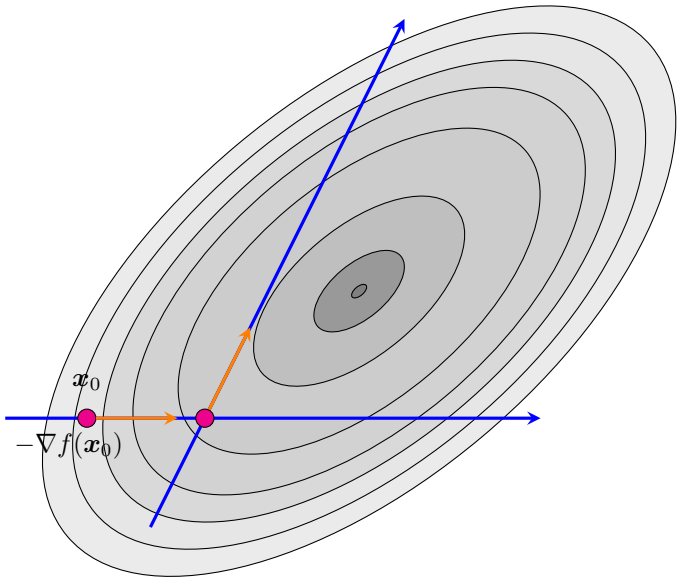
最急降下法のアルゴリズムは以下のように表すことができる.

$i_0,\ x_0$ を決める.
収束するまで繰り返し {

$$x_{i+1} = x_i - \alpha \nabla f(\boldsymbol{x}_i)$$
$$i = i + 1$$

}

α の決め方は様々であり, $-\nabla f(\boldsymbol{x})$ の最適値を解析的に求めるかラインサーチによって求めて移動する.
このときの移動量は同じであるが過去の勾配の大きさの履歴に応じて移動量を決定する.



以下の関数の最小値を最急降下法で求めてみる.

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \begin{pmatrix} 6 & 1 \\ 1 & 2 \end{pmatrix} \boldsymbol{x} - \begin{pmatrix} 2 \\ 1 \end{pmatrix}^T \boldsymbol{x}$$

これの初期値として $\boldsymbol{x} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ として python で実装した場合は以下のようになる.

```
import numpy as np
from scipy import linalg as la
TH = 0.000001
A = np.matrix([[6.0,1.0],[1.0,2.0]])
b = np.matrix([[2.0],[1.0]])
x = np.matrix([[0.0],[0.0]])
```

```
alpha = 0.1
for i in range(100):
    dx = A*x - b
    x = x - alpha * dx
    if(dx.T*dx < TH):
        break
print(i, ':', x.T)
```

実行結果として, 最小値は $\begin{pmatrix} 0.27281847 \\ 0.36325003 \end{pmatrix}$ が得られる.

α の値を移動量を最初大きくし, 最適解に近づくにつれてだんだん小さくした場合の python の実装は以下のようになる.

```
import numpy as np
from scipy import linalg as la
TH = 0.000001
A = np.matrix([[6.0,1.0],[1.0,2.0]])
b = np.matrix([[2.0],[1.0]])
x = np.matrix([[0.0],[0.0]])
alpha = 0.5
for i in range(100):
    dx = A*x - b
    x = x - alpha * dx
    alpha = alpha*0.8
    if(dx.T*dx < TH):
        break
print(i, ':', x.T)
```

実行結果として, 最小値は $\begin{pmatrix} 0.27279382 \\ 0.36320777 \end{pmatrix}$ が得られる.

関数 $f(x) = \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x}$ の最小値を与える α について解析的に求める方法はまず変数 $\boldsymbol{x}$ について

$$\boldsymbol{x} = \boldsymbol{x}_0 + \alpha\boldsymbol{d}$$

とおき, これを代入することで計算していく. まず

$$\begin{aligned} f(x) &= f(\boldsymbol{x}_0 + \alpha\boldsymbol{d}) \\ f(\alpha) &= \frac{1}{2}(\boldsymbol{x}_0 + \alpha\boldsymbol{d})^T A(\boldsymbol{x}_0 + \alpha\boldsymbol{d}) - \boldsymbol{b}^T(\boldsymbol{x}_0 + \alpha\boldsymbol{d}) \end{aligned}$$

最小値を与える α であるので極値を求めればよく,

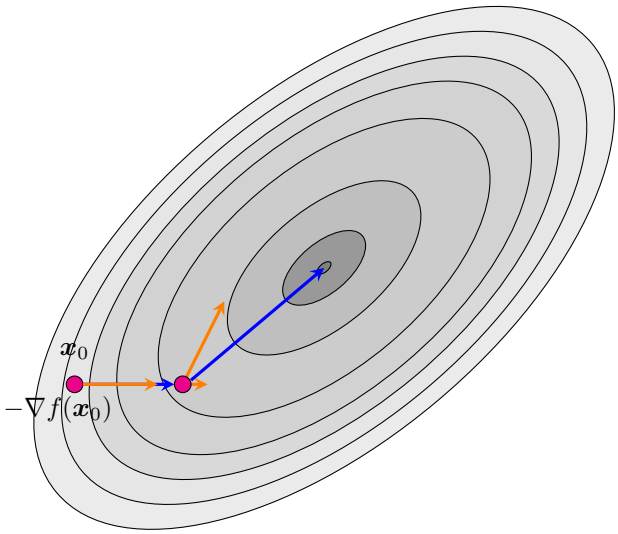
$$\frac{d}{d\alpha} f(\alpha) = 0$$

が成立すればよい. したがってこれを計算すると

$$\begin{aligned} & \frac{d}{d\alpha} f(\alpha) = 0 \\ \iff & (\boldsymbol{x}_0 + \alpha\boldsymbol{d})^T A\boldsymbol{d} - \boldsymbol{b}^T\boldsymbol{d} = 0 \\ \iff & \alpha\boldsymbol{d}^T A\boldsymbol{d} + \boldsymbol{x}_0^T A\boldsymbol{d} - \boldsymbol{b}^T\boldsymbol{d} = 0 \\ \iff & \alpha\boldsymbol{d}^T A\boldsymbol{d} + \nabla f(\boldsymbol{x}_0)^T\boldsymbol{d} = 0 \\ \iff & \alpha = -\frac{\nabla f(\boldsymbol{x}_0)^T\boldsymbol{d}}{\boldsymbol{d}^T A\boldsymbol{d}} \end{aligned}$$

3.2.2　共役勾配法

最急降下法では, 直線的に最適解に向かわないので今まで進んできた方向と共役な方向に移動方向を決める方法を共役勾配法という.



共役勾配法を式にすると以下のように表すことができる.

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{x}_{k-1} + c_k \boldsymbol{p}_k \\ \boldsymbol{p}_{k+1} &= -\nabla f(\boldsymbol{x}_k) + \alpha_k \boldsymbol{p}_k \end{aligned} \tag{3.5}$$

共役勾配法は勾配だけでなく, 今まで進んできた方向も考慮して探索の方向を決めることで効率化を図っている. 「今までの進んできた方向の考慮」にあたり, ベクトルの共役という性質を利用する.
共役とは, $\boldsymbol{x}$, $\boldsymbol{y}$ が,

$$\boldsymbol{x}^T A \boldsymbol{y} = \langle \boldsymbol{x}, \boldsymbol{y} \rangle_A = 0$$

を満たすことをいう.
固有ベクトルによる対称行列の対角化を行う. n 次対称行列 A は
固有ベクトル $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n$ を並べて作る行列

$$V = \begin{pmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{pmatrix}$$

を用いて

$$V^T A V = \Lambda$$
$$\Lambda = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{pmatrix}$$

ここで, $\lambda_1, \lambda_2, \cdots, \lambda_n$ は固有値であり, このように対角化することが可能となる.
ここで $V^T V = V V^T = I$ であるから

$$V V^T A V V^T = V \Lambda V^T, \quad A = V \Lambda V^T$$

となる. したがって, 対称行列は固有値を対角要素に持つ行列 Λ と固有ベクトルをならべて作る行列 V で表すことができる

したがって共役な $\boldsymbol{x}_1, \boldsymbol{x}_2$ ついて

$$\begin{aligned}
\boldsymbol{x}_1^T A \boldsymbol{x}_2 &= \boldsymbol{x}_1^T V \Lambda V^T \boldsymbol{x}_2 \\
&= \boldsymbol{x}_1^T V \Lambda^{\frac{1}{2}} \Lambda^{\frac{1}{2}} V^T \boldsymbol{x}_2 \\
&= \boldsymbol{x}_1^T V \Lambda^{\frac{1}{2}T} \Lambda^{\frac{1}{2}} V^T \boldsymbol{x}_2 \\
&= \boldsymbol{x}_1^T \left(\Lambda^{\frac{1}{2}} V^T \right)^T \Lambda^{\frac{1}{2}} V^T \boldsymbol{x}_2 \\
&= \left(\Lambda^{\frac{1}{2}} V^T \boldsymbol{x}_1 \right)^T \Lambda^{\frac{1}{2}} V^T \boldsymbol{x}_2
\end{aligned}$$

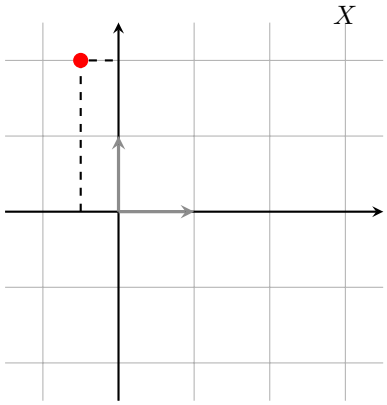
Λ は対角行列であるのでその平方根である $\Lambda^{\frac{1}{2}}$ も対角行列となる．また対角行列の転置行列は元の行列と等しいことを考慮して先のように変形することが出来る．ここでベクトル z に対して

$$\boldsymbol{z} = \Lambda^{\frac{1}{2}} V^T \boldsymbol{x}$$

なる座標変換をした後のベクトルについて考える．

$$\begin{aligned}
\boldsymbol{z} &= \Lambda^{\frac{1}{2}} V^T \boldsymbol{x} \\
\boldsymbol{z} &= \Lambda^{\frac{1}{2}} \boldsymbol{y} \\
\boldsymbol{y} &= V^T \boldsymbol{x}
\end{aligned}$$

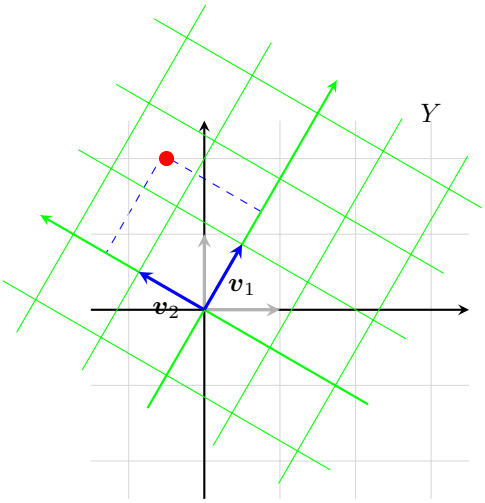
とする．そして以下の平面について考える．



これから $\boldsymbol{y}$ について変換を行うと

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{v}_1^T \boldsymbol{x} \\ \boldsymbol{v}_2^T \boldsymbol{x} \end{pmatrix}$$

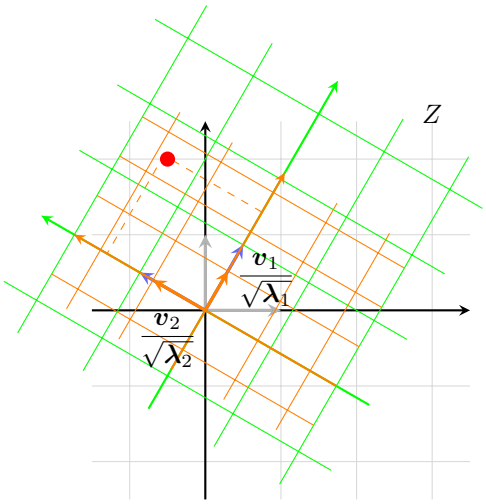
となるので Y についての平面は以下のように表すことができる．



これに加えて $\boldsymbol{z}$ に関して,

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} \sqrt{\boldsymbol{\lambda}_1} y_1 \\ \sqrt{\boldsymbol{\lambda}_1} y_1 \end{pmatrix}$$

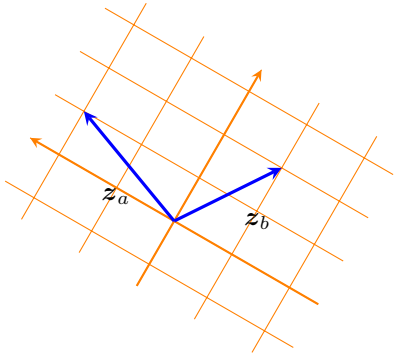
となるので固有値の平方根をとることで Z についての平面は以下のようになる.



座標変換後の直交とは

$$\boldsymbol{z}_a^T \cdot \boldsymbol{z}_b = 0$$

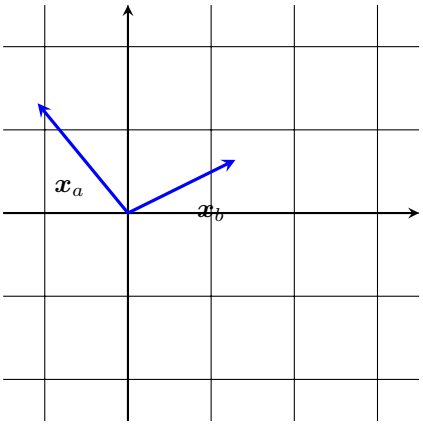
が成立することである.



これについて元の座標においては共役である. つまり

$$\boldsymbol{x}_a^T A \boldsymbol{x}_b = 0$$

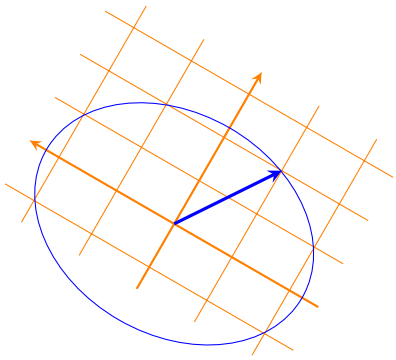
ということである.



円については座標変換後の円が

$$\boldsymbol{z}^T \cdot \boldsymbol{z} = \text{const}$$

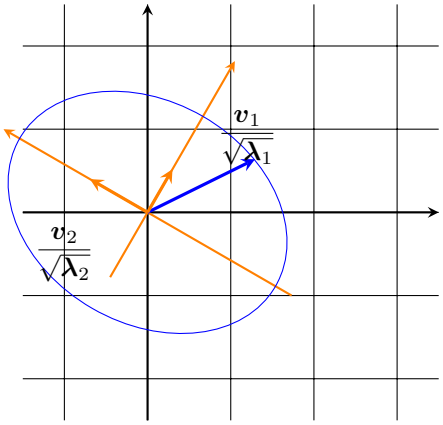
とする.



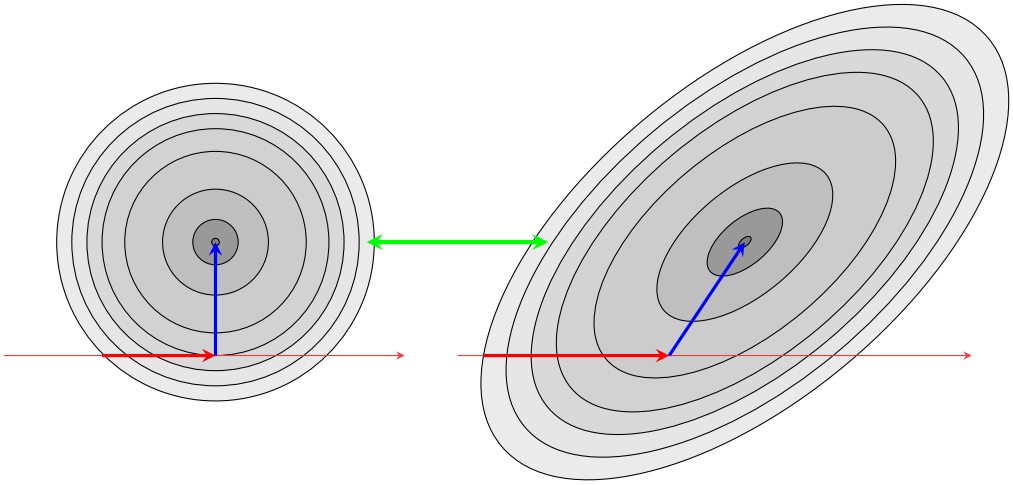
これを元の座標では楕円であり以下の式を満たす

$$\boldsymbol{x}^T A \boldsymbol{x} = \text{const}$$

このとき半径 (長半径, 短半径) は, 固有値の平方根の逆数となる.



つまり共役の方向に近似解を更新するということは, 楕円を円に変換したときに直行する方向に近似解を更新することに相当する.



A を N 次正定値対称行列として, 互いな共役なベクトル $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \cdots, \boldsymbol{p}_N$ を用いて,

$$A\boldsymbol{x} = \boldsymbol{b}$$

の解 $\boldsymbol{x}^*$ を表現することを考える.

$$\boldsymbol{x}^* = \sum_{k=1}^{N} c_k \boldsymbol{p}_k$$

とおくと,
$\boldsymbol{p}_i$ は, $\boldsymbol{p}_j$ $(j \neq i)$ は共役であるから,

$$\boldsymbol{p}_i^T A\boldsymbol{x}^* = \boldsymbol{p}_i^T \sum_{k=1}^{N} c_k A\boldsymbol{p}_k = c_i \boldsymbol{p}_i^T A\boldsymbol{p}_i = \boldsymbol{p}_i^T \boldsymbol{b}$$

$$\Longrightarrow c_i = \frac{\boldsymbol{p}_i^T \boldsymbol{b}}{\boldsymbol{p}_i^T A\boldsymbol{p}_i}$$

として c_i を定めることができる.

共役勾配法の 2 次形式の場合の解を求めてみる. つまり

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} - \boldsymbol{b}^T \boldsymbol{x}$$

に対して

$$\nabla f(\boldsymbol{x}) = A\boldsymbol{x} - \boldsymbol{b} = 0$$

すなわち

$$A\boldsymbol{x} = \boldsymbol{b}$$

の解を求める.
適当な初期値 $\boldsymbol{x}_0$ を定め, その負の勾配を $\boldsymbol{r}_0$ とする.

$$\boldsymbol{r}_0 = -\nabla f(\boldsymbol{x}_0 + \boldsymbol{b})$$

最初の基底 $\boldsymbol{p}_1$ を $\boldsymbol{r}_0$ にとる. つまり

$$\boldsymbol{p}_1 = \boldsymbol{r}_0 = -A\boldsymbol{x}_0 + \boldsymbol{b}$$

第 k 次の基底 $\boldsymbol{p}_k$ が定まった時

$$c_k = \frac{\boldsymbol{p}_k^T \boldsymbol{b}}{\boldsymbol{p}_k^T A\boldsymbol{p}_k}$$

を用いて, $\boldsymbol{x}_k$ を

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + c_k \boldsymbol{p}_k$$

と求められる. これから $\boldsymbol{x}_k$ での勾配 $\boldsymbol{r}_k$ を求めることができる.

$$\boldsymbol{r}_k = -\nabla f(\boldsymbol{x}_k) = -A\boldsymbol{x}_k + \boldsymbol{b}$$

第 $k+1$ 次の基底 $\boldsymbol{p}_{k+1}$ を, $\boldsymbol{r}_k$ と $\boldsymbol{p}_k$ の合成, つまり

$$\boldsymbol{p}_{k+1} = \boldsymbol{r}_k + \alpha_k \boldsymbol{p}_k$$

とおいて, これが $\boldsymbol{p}_k$ と共役となるように定める.
このとき, α_k は以下の等式を満たす.

$$\boldsymbol{p}_k^T A\boldsymbol{p}_{k+1} = \boldsymbol{p}_k^T A(\boldsymbol{r}_k + \alpha_k \boldsymbol{p}_k) = \boldsymbol{p}_k^T A\boldsymbol{r}_k + \alpha_k \boldsymbol{p}_k^T \boldsymbol{p}_k = 0$$

したがって

$$\alpha_k = -\frac{\boldsymbol{p}_k^T A \boldsymbol{r}_k}{\boldsymbol{p}_k^T A \boldsymbol{p}_k}$$

よって, 求める第 $k+1$ 次の基底は

$$\boldsymbol{p}_{k+1} = \boldsymbol{r}_k + \alpha_k \boldsymbol{p}_k = \boldsymbol{r}_k - \frac{\boldsymbol{p}_k^T A \boldsymbol{r}_k}{\boldsymbol{p}_k^T A \boldsymbol{p}_k} \boldsymbol{p}_k$$

と求まる. さらに, これを用いて $\boldsymbol{x}_{k+1}$ を

$$c_{k+1} = \frac{\boldsymbol{p}_{k+1}^T \boldsymbol{b}}{\boldsymbol{p}_{k+1}^T A \boldsymbol{p}_{k+1}}$$
$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + c_{k+1} \boldsymbol{p}_{k+1}$$

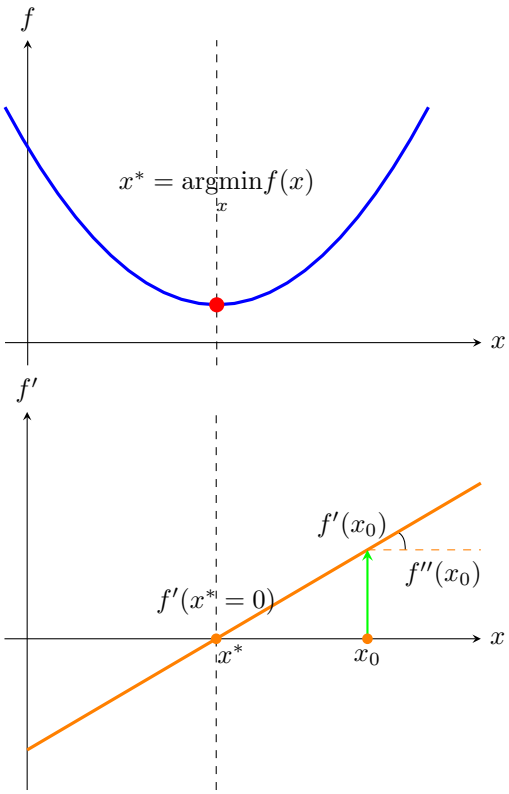
と求まる.

これより関数が 2 次形式であれば, くりかえし回数は, たかだか共役なベクトルの数 (つまり $\boldsymbol{x}$ の次元数) となる.

3.2.3 ニュートン法

ニュートン法とは勾配だけでなく, ヘシアン (スカラにおける 2 次微係数に相当する量) も使って, 近似解の変更の方向と量を決定する方法である.

2 次式の最小化問題の場合は以下のように考えることができる.



これによりニュートン法の最小化の近似値は

$$x^* = x_0 - \frac{1}{f''(x_0)} f'(x_0) \tag{3.6}$$

一般の場合のニュートン法では 2 次式を仮定して傾きを元に極値の位置を推定していく. しかし導関数によっては発散してしまう可能性がある.

多変数のニュートン法では以下のヘッセ行列 (ヘシアン, ヘシアン行列) となる.

$$H(\boldsymbol{x}_0) = \begin{pmatrix} \dfrac{\partial^2 f(\boldsymbol{x}_0)}{\partial x_1 x_1} & \dfrac{\partial^2 f(\boldsymbol{x}_0)}{\partial x_1 \partial x_2} \\ \dfrac{\partial^2 f(\boldsymbol{x}_0)}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f(\boldsymbol{x}_0)}{\partial x_2 \partial x_2} \end{pmatrix}$$

微小量の変化したときの変化量が 0 となればよいので

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{x}} f(\boldsymbol{x}_0 + \Delta \boldsymbol{x}) = \nabla f(\boldsymbol{x}_0) + H(\boldsymbol{x}_0)\Delta \boldsymbol{x} = 0 \\ \Delta \boldsymbol{x} = -H(\boldsymbol{x}_0)^{-1} \nabla f(\boldsymbol{x}_0) \\ x^* = x_0 - H(\boldsymbol{x}_0)^{-1} \nabla f(\boldsymbol{x}_0) \quad (2 \text{ 次系の場合は即最適解}) \end{aligned}$$

一般の場合は漸化式で

$$x_{i+1} = x_i - H(\boldsymbol{x}_i)^{-1} \nabla f(\boldsymbol{x}_i)$$

となる.

勾配法においては, 得られる解は初期値に依存する.

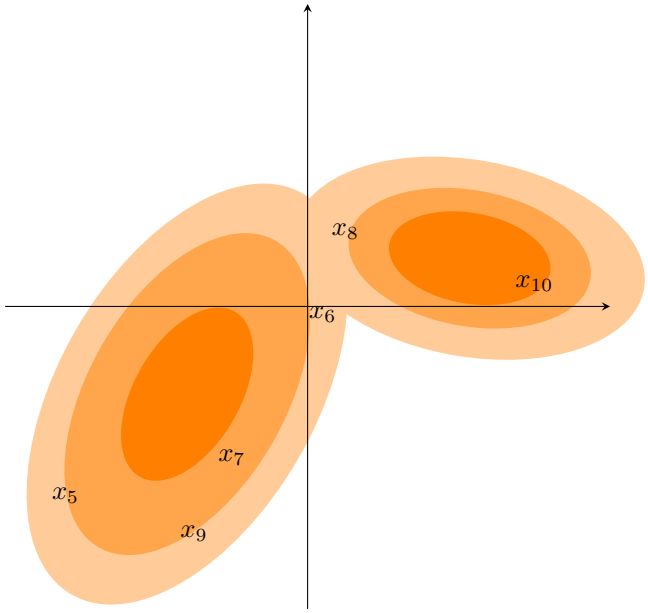
4　線形識別と勾配法

4.1　パターン認識

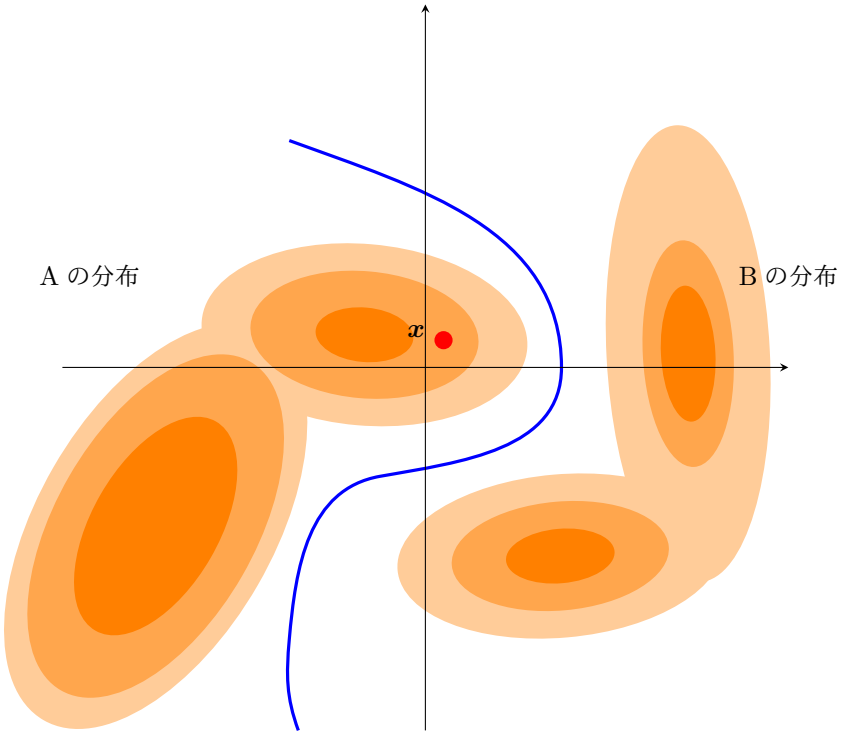
パターン認識とは予め与えられた大量のデータ (学習データ; 一般にはクラス既知) 利用して, クラス未知のデータ (テストデータ) が, どのクラスに属するか推定する問題である.
パターン認識の方法として

1. 与えられた学習データをベクトル表現に置き換える (x_1, x_2, x_3, x_4)

2. 特徴ベクトル空間上での分布を調べる

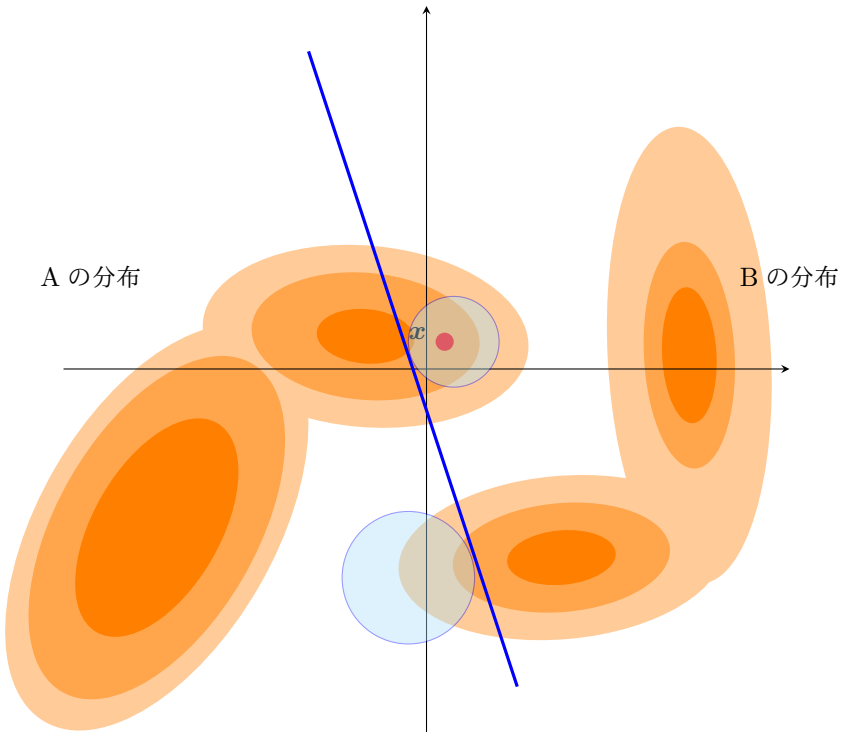


3. 分布を元にクラスの境界を定める.

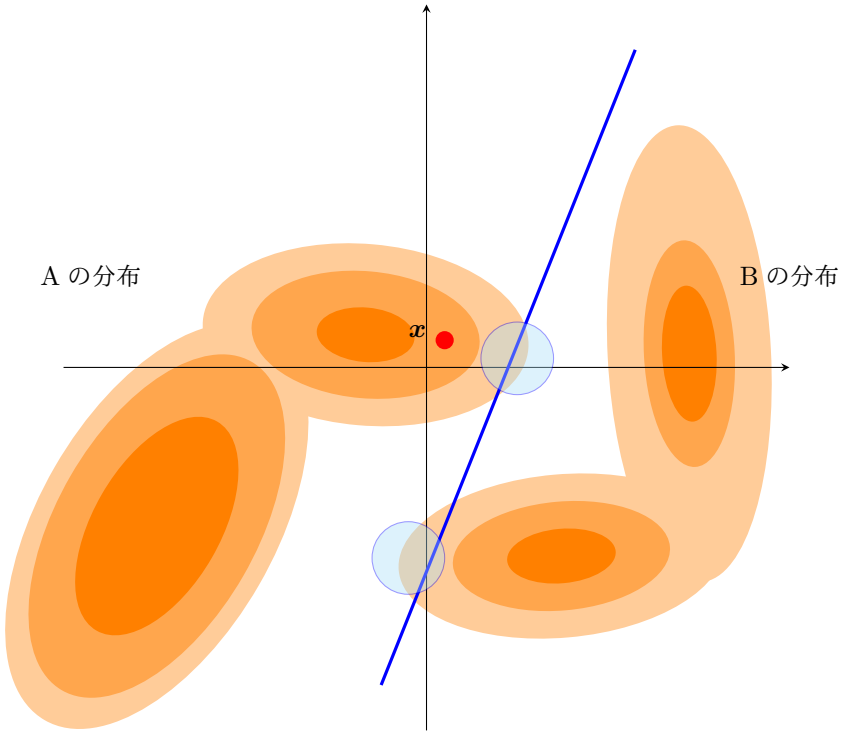


4. テストデータのベクトル $(\boldsymbol{x})$ が境界のどちら側にあるかを調べることで, テストデータ (x) のクラスを定める.

4.2 線形識別



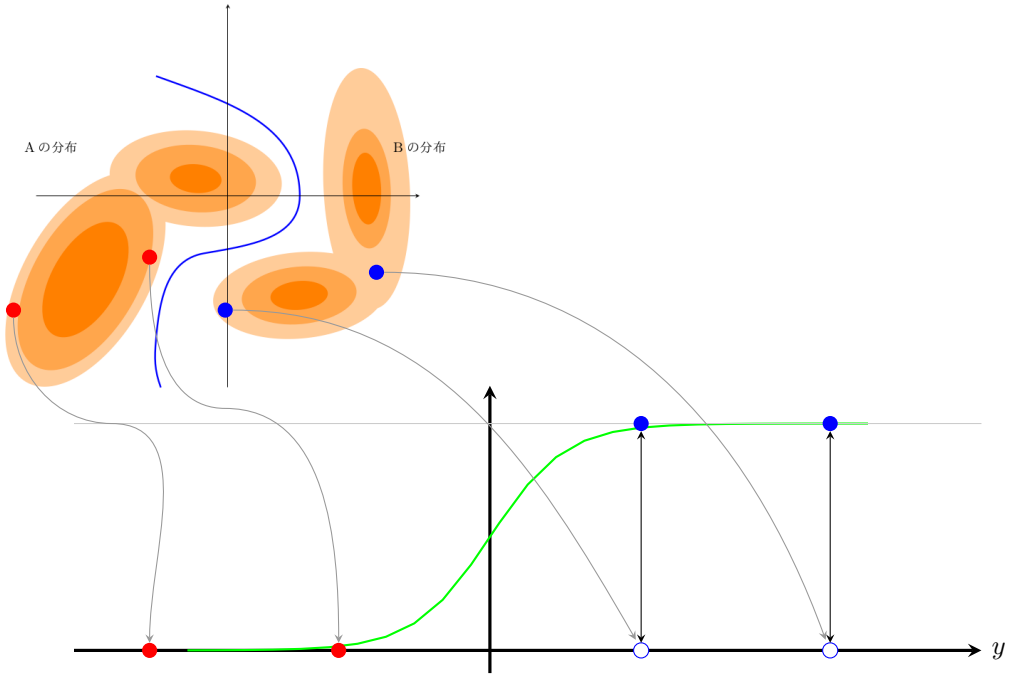
線形識別においては図の水色で囲んでいるところが誤認識となる領域となる. 識別平面の設定によって, 誤認識となる領域が変わる. 以下のように識別平面の設定によると先の場合より領域が小さくなる.



線形識別においてクラス A の分布を $f(x) = 0$, クラス B の分布を $f(x) = 1$ とする. ここで,

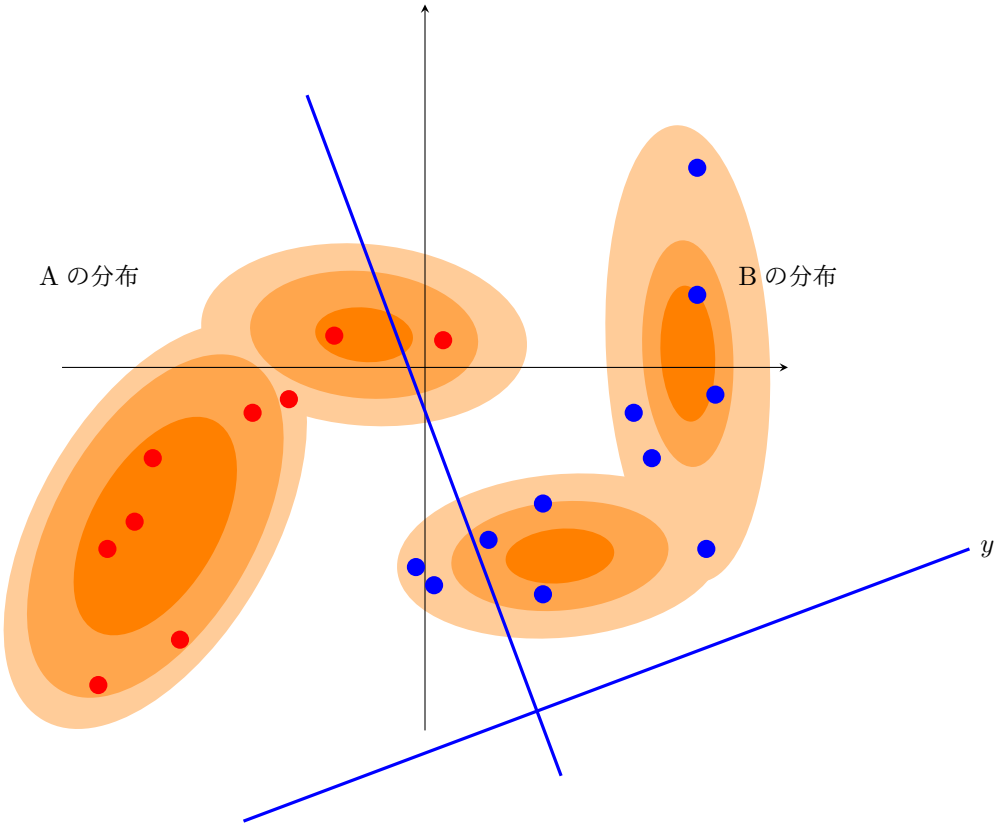
$$y = g(x; \Theta)$$

とすることでロジスティック回帰分析を行う.



$$z = f(y; \Psi) = f(g(x; \Theta); \Psi)$$

線形識別の場合は



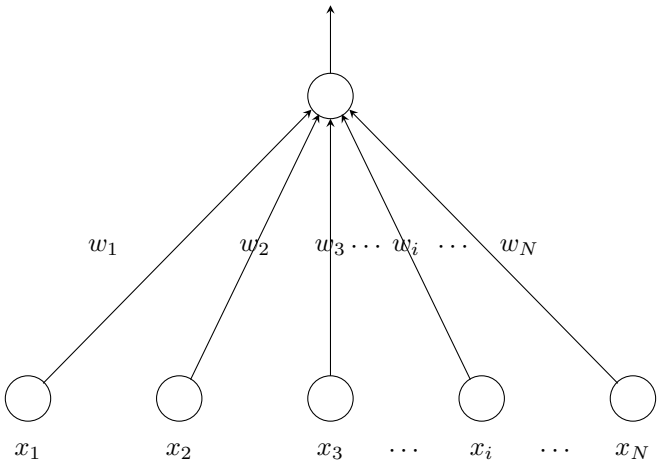
入力データ (x_i) に対して線形識別における座標変換すると

$$y_i = \boldsymbol{w}^T \boldsymbol{x}_i$$

これに対してロジスティック回帰を適用させると

$$z_i = \frac{1}{1+\exp(-y_i)} = \frac{1}{1+\exp(-\boldsymbol{w}^T \boldsymbol{x}_i)} \tag{4.1}$$

以下の 2 層のニューラルネット (パーセプトロン) について考える.



$\boldsymbol{x}^{(k)} = \begin{pmatrix} x_1^{(k)} & x_2^{(k)} & \cdots & x_i^{(k)} & \cdots & x_N^{(k)} \end{pmatrix}^T$: 第 k 入力ベクトル
$y^{(k)}$: 第 k に対する出力

$$y^{(k)} = f\left(\sum_{i=1}^{N} w_i x_i^{(k)}\right)$$
$$f(z) = \frac{1}{1+\exp(-z)}$$

カテゴリ A であれば 1, そうでなければ 0 とする. また
$\boldsymbol{w} = \begin{pmatrix} w_1 & w_2 & \cdots & w_i & \cdots & w_N \end{pmatrix}$: 重みベクトル (パーセプトロンのパラメータ)
$t^{(k)}$: 第 k 入力に対する正解カテゴリ (カテゴリ A であれば 1 そうでなければ 0)
M : データの個数
目的関数 $H(\boldsymbol{w})$ としたとき目的関数の中身は以下のようになる。

$$\begin{aligned} H(\boldsymbol{w}) &= \sum_{j=1}^{M} \frac{1}{2} \left(y^{(j)} - t^{(j)} \right)^2 \\ &= \sum_{j=1}^{M} \frac{1}{2} \left(f\left(\sum_{i=1}^{N} w_i {x_i}^{(j)} \right) - t^{(j)} \right)^2, f(z) = \frac{1}{1+e^{-z}} \end{aligned}$$

この時最急降下法により、$\boldsymbol{w}$ を求めると次のようになる。
まず更新式は次のようにあらわされる。

$$\begin{aligned} \boldsymbol{w}_{n+1} &= \boldsymbol{w}_n - \alpha_n \frac{\partial H\left(\boldsymbol{w}_n\right)}{\partial \boldsymbol{w}} \\ \alpha_{n+1} &= \beta \alpha_n, \alpha_0 = 0.5, \beta = 0.98 \end{aligned}$$

ここで、以下の性質が成り立つ

$$\begin{aligned} H(\boldsymbol{w}) &= \sum_{j=1}^{M} \frac{1}{2} \left(y^{(j)} - t^{(j)} \right)^2, \\ y^{(j)} &= f\left(\sum_{i=1}^{N} w_i {x_i}^{(j)} \right) \text{の時} \\ \frac{\partial}{\partial w_i} H(\boldsymbol{w}) &= \sum_{j=1}^{M} \left(y^{(j)} - t^{(j)} \right) y^{(j)} \left(1 - y^{(j)} \right) {x_i}^{(j)} \end{aligned}$$

よって、次のことが成り立つ

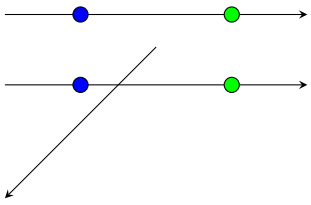
$$\begin{aligned} \nabla H(\boldsymbol{w}) &= \frac{\partial H(\boldsymbol{w})}{\partial \boldsymbol{w}} \\ &= [H(\boldsymbol{w})] \end{aligned}$$

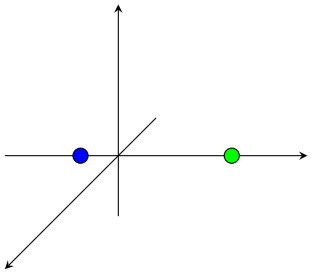
5　ニューラルネットワークと確率降下法

入力-出力の対 $(\boldsymbol{x},\ \boldsymbol{y})$ が大量に与えられば, 関数 $y = f(x)$ を精度よくもとめることができる.

6　情報圧縮と等式制約の最適化問題

情報圧縮の典型的問題は主成分分析である. 情報圧縮する必要性として, 特徴空間の次元が高くなると, 信頼できる境界を得るために必要な学習データの量が幾何級数的に増大するためである.
つまり次元を増やすたびに, 周囲にクラスを定めていない未知の空間が幾何級数的に増加するということである.

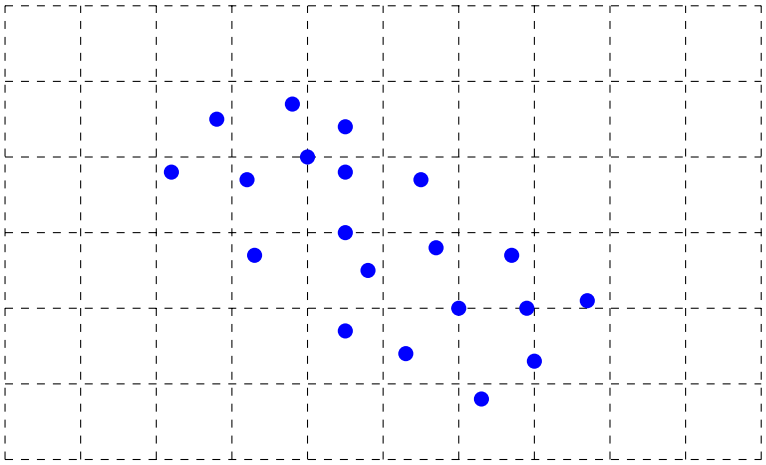




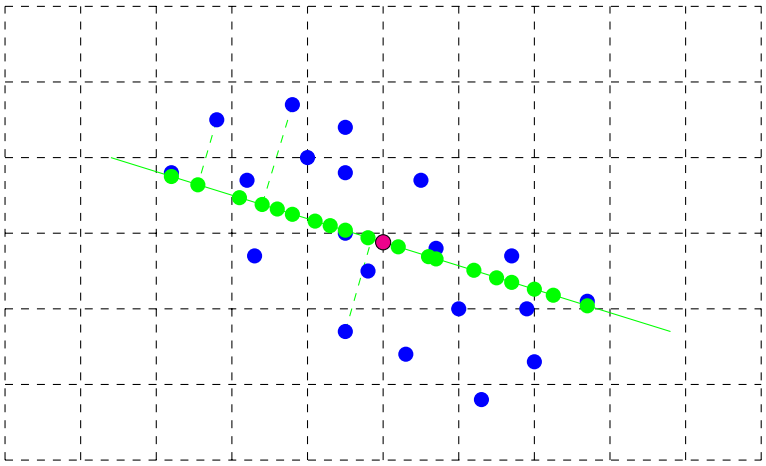
- 情報圧縮
  高次元空間上のデータを, 低次元の空間に, ある基準のもとで写像する
- 主成分分析
  低次元への写像に伴う情報のロスを最小に抑える線形写像

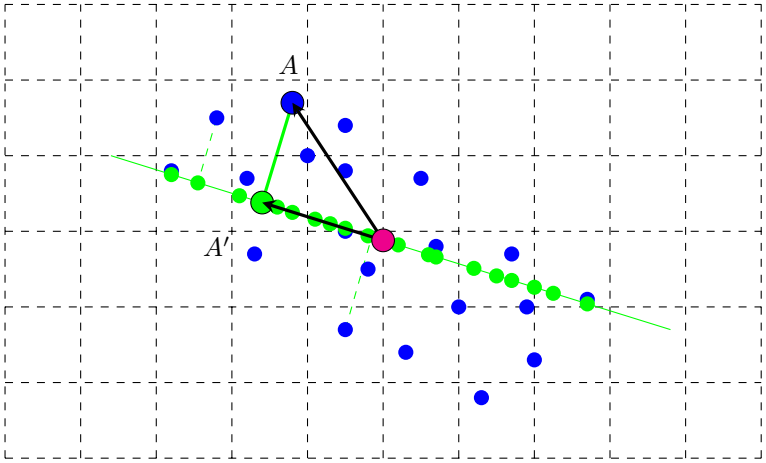
6.1 主成分分析

以下のように R^2 の空間に分布するデータが与えられたとする.



この空間上に分布するデータを, 直線状の部分空間に写像することを考える. すると以下のような写像が考えられる.





このとき $A-A'$ がロスとなる．A' において射影成分の分散が最大化されるように直線を引く時，これはロスが最小化されることとなる．これは OA の距離が一定であるので，$A-A'$ の値が最小となるときは $O-A'$ の値が最大となるためである．部分空間が変われば，分散も変わるので，最も射影成分の分散の大きくなる部分空間を求めることがよい．

次に第 1 主成分の求め方について扱う．

いま P 次元のベクトルが,N 個あるとする．

$$\boldsymbol{x}_n = \left(x_{np}\right),\ (p=1,2,3,\cdots,P)\ (n=1,2,3,\cdots,N)$$

簡単のため，各変数についてその平均値を 0 とする．0 ではない場合は平均値を引いて新たに変数を定義しなおせば一般性は失われない．

ここで，測定データ全体は次の行列 X で与えられる．

$$X = \begin{pmatrix}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_n\end{pmatrix}^T = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1P} \\ x_{21} & x_{22} & \cdots & x_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NP} \end{pmatrix}$$

求める主成分を

$$\boldsymbol{w} = \begin{pmatrix} w_1 & w_2 & \cdots & w_P \end{pmatrix}^T \quad \|\boldsymbol{w}\| = \boldsymbol{w}^T\boldsymbol{w} = 1$$

とし，今 n 番目のデータを

$$\boldsymbol{x}_n = \begin{pmatrix} x_{n1} & x_{n2} & \cdots & x_{nP} \end{pmatrix}^T$$

の主成分への射影 z_n を考えると，

$$z_n = \boldsymbol{x}_n^T\boldsymbol{w} = \sum_{p=1}^{P} x_{np}w_p$$

$\boldsymbol{z} = \begin{pmatrix} z_1 & z_2 & \cdots & z_N \end{pmatrix}^T$ とすれば，

$$\boldsymbol{z} = X\boldsymbol{w}$$

このとき z_n の分散 σ^2 は

$$\begin{aligned}\sigma^2 &= \frac{1}{N}\boldsymbol{z}^T\boldsymbol{z} = \frac{1}{N}(X\boldsymbol{w})^T(X\boldsymbol{w}) \\ &= \frac{1}{N}\boldsymbol{w}^T X^T X\boldsymbol{w} = \boldsymbol{w}^T C\boldsymbol{w}\end{aligned}$$

ただし,

$$C = \{c_{ij}\},\ c_{ij} = \frac{1}{N}\sum_{k=1}^{N} x_{ki}x_{kj}$$

C は, 共分散行列とよばれ, 平均値が 0 でない一般的な場合は

$$c_{ij} = \frac{1}{N}\sum_{k=1}^{N}(x_{ki} - x_i)(x_{kj} - \bar{x_j})$$

となる. ここで, 第 1 主成分は $\boldsymbol{w}$ で σ^2 の最大化すること. つまり

$$\max_{\boldsymbol{w}} \boldsymbol{w}^T C \boldsymbol{w}$$

を満たすことである. この中で条件として $\boldsymbol{w}^T\boldsymbol{w} = 1$ を満たさなければならない. 定式化すると

$$\begin{aligned} &\max_{\boldsymbol{w}} \boldsymbol{w}^T C \boldsymbol{w} \\ &s.t.\ \ \boldsymbol{w}^T\boldsymbol{w} = 1 \end{aligned}$$

6.2 Lagrange の未定乗数法

以下の問題を求める方法である.

$$\begin{aligned} &\max_{x} f(\boldsymbol{x}) \\ &s.t.\ g(\boldsymbol{x}) = 0 \end{aligned}$$

一般的な解法として

$$L(\boldsymbol{x}) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{x}) \tag{6.1}$$

とおき,

$$\begin{cases} \dfrac{\partial}{\partial \boldsymbol{x}} L(\boldsymbol{x}) = 0 \\ g(\boldsymbol{x}) = 0 \end{cases}$$

の連立方程式を解くことである.

周囲の長さが X の長方形の中で, 面積の一番大きいものの辺の長さを求める.
各辺を a, b とおくと問題は

$$\begin{aligned} &\max ab \\ &\quad s.t.\ 2(a+b) = X \end{aligned}$$

という問題となる.
つまり,

$$\begin{aligned} f(a,b) &= ab \\ g(a,b) &= 2(a+b) - X \end{aligned}$$

となるので,

$$\begin{aligned} L(a,b,\lambda) &= f(a,b) - \lambda g(a,b) \\ &= ab - \lambda(2a + 2b - X) \end{aligned}$$

について連立方程式を立てて解けばよい.

$$\begin{aligned} \frac{\partial}{\partial a} L(a,b,\lambda) &= b - 2\lambda = 0 \\ \frac{\partial}{\partial b} L(a,b,\lambda) &= a - 2\lambda = 0 \\ \frac{\partial}{\partial \lambda} L(a,b,\lambda) &= -2a - 2b + X = 0 \end{aligned}$$

となるので, 整理すると

$$\begin{aligned} &a = b = \lambda \\ &-4\lambda - 4\lambda + X = 0 \\ &\lambda = \frac{X}{8} \\ &a = b = 2\lambda = \frac{X}{4} \end{aligned}$$

となる.

Lagrange の未定乗数法を用いて $\boldsymbol{w}$ による σ^2 の最大化問題は $\|\boldsymbol{w}\| = \boldsymbol{w}^T\boldsymbol{w} = 1$ の条件の下で解くことを考える.
Lagrange の未定乗数法より以下の方程式を立てる.

$$L(\boldsymbol{w}, \lambda) = \boldsymbol{w}^T C \boldsymbol{w} - \lambda(\boldsymbol{w}^T\boldsymbol{w} - 1)$$

とおいて, $\dfrac{\partial L}{\partial \boldsymbol{w}} = 0$ を解くことを考える. すると

$$\begin{aligned} \frac{\partial L}{\partial \boldsymbol{w}} &= C\boldsymbol{w} + (\boldsymbol{w}^T C)^T - \lambda(\boldsymbol{w} + \boldsymbol{w}) \\ &= C\boldsymbol{w} + C^T\boldsymbol{w} - 2\lambda\boldsymbol{w} \\ &= C\boldsymbol{w} + C\boldsymbol{w} - 2\lambda\boldsymbol{w} \\ &= 2(C - \lambda I)\boldsymbol{w} = 0 \end{aligned}$$

となる. 共分散行列に関して

$$c_{ij} = \frac{1}{N}\sum_{k=1}^{N} x_{ki}x_{kj} = \frac{1}{N}\sum_{k=1}^{N} x_{kj}x_{ki} = c_{ji}$$

であるから, 対称行列であることがいえる. よって,

$$(C - \lambda I)\boldsymbol{w} = 0 \iff C\boldsymbol{w} = \lambda\boldsymbol{w}$$

となり, 問題は固有値問題に帰着される.
また

$$(C-\lambda I)\boldsymbol{w}=0 \iff \det|C-\lambda I|=0$$

となるので, λ が満たすべき条件は, 共分散行列 C の固有方程式となる.
したがって, 射影の分散の最大値を与える主成分 $\boldsymbol{w}$ は分散共分散行列の固有ベクトルのひとつであり, λ はその固有ベクトルに対応した固有値であることがわかる.
データ数が P 個あることから $(C-\lambda I)\boldsymbol{w}=0$ を満たす固有ベクトル $\boldsymbol{w}$, 固有値 λ はそれぞれ P 個ある.
このうち, どれが射影の分散の最大値を与えるかを考える.

$$C\boldsymbol{w}=\lambda\boldsymbol{w}$$

という関係の下,

$$\sigma^2=\boldsymbol{w}^TC\boldsymbol{w}$$

であるから

$$\sigma^2=\boldsymbol{w}^TC\boldsymbol{w}=\lambda\boldsymbol{w}^T\boldsymbol{w}$$

となり, $\boldsymbol{w}^T\boldsymbol{w}=1$ であることを考慮すれば

$$\sigma^2=\lambda$$

となる. つまり射影の分散の値は固有値に等しい.
よって, 求める主成分は最大固有値に対応した固有ベクトルとして与えられることが分かる.

第 2 以降の主成分の求め方について,
帰納的に第 m 主成分を求める. 分散共分散行列 C の固有値のうち大きいものから順に $m-1$ 個に対応する固有ベクトルとして, $m-1$ の主成分 $\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_k,\cdots,\boldsymbol{w}_{m-1}$ が求まっているとする. これらのベクトルは

$$\begin{aligned}(C-\lambda I)\boldsymbol{w}_k&=0\\ \boldsymbol{w}_k^T\boldsymbol{w}_k&=1\end{aligned}$$

を満たし, 互いに直交しているものとする[*1].

$$\boldsymbol{w}_k^T\boldsymbol{w}_{k'(\neq k)}=0$$

つまり

$$\begin{aligned}f(\boldsymbol{w}_m)&=\boldsymbol{w}_m^TC\boldsymbol{w}_m\\ s.t.\ \boldsymbol{w}_m^T\boldsymbol{w}_m&=1,\ \boldsymbol{w}_k^T\boldsymbol{w}_{k'(k'\neq k)}=0\end{aligned}$$

という関係式が得られる.
Lagrange の未定乗数法より

$$J_m=\boldsymbol{w}_m^TC\boldsymbol{w}_m-\lambda_m(\boldsymbol{w}_m^T\boldsymbol{w}_m-1)-\sum_{k=1}^{m-1}\mu_k\boldsymbol{w}_m^T\boldsymbol{w}_k$$

[*1] $\boldsymbol{w}$ は固有ベクトルなので自明

とおいて, $\dfrac{\partial J_m}{\partial \boldsymbol{w}_m} = 0$ とおくと,

$$
\begin{aligned}
& 2C_{\boldsymbol{w}_m} - 2\lambda_m \boldsymbol{w} - \sum_{k=1}^{m-1} \mu_k \boldsymbol{w}_k = 0 \\
\Longleftrightarrow\ & C_{\boldsymbol{w}} - \lambda_m \boldsymbol{w}_m - \sum_{k=1}^{m-1} \mu_k \boldsymbol{w}_k = 0
\end{aligned}
$$

ここで μ_k について $\mu_k = \frac{1}{2}\mu_k$ として再定義.
左から $\boldsymbol{w}_j^T$ $(j = 1, 2, 3, \cdots, m-1)$ をかけると,$(\boldsymbol{w}_k^T \boldsymbol{w}_{k'(k' \neq k)} = 0$ を考慮して)

$$
\begin{aligned}
& \boldsymbol{w}_j^T C \boldsymbol{w}_m - \lambda_m \boldsymbol{w}_j^T \boldsymbol{w}_m - \sum_{k=1}^{m-1} \mu_k \boldsymbol{w}_j \boldsymbol{w}_k = 0 \\
\Longleftrightarrow\ & \boldsymbol{w}_j^T C \boldsymbol{w}_m - \mu_j = 0
\end{aligned}
$$

となる. また

$$
\begin{aligned}
\boldsymbol{w}_j^T C \boldsymbol{w}_m &= \boldsymbol{w}_m^T C \boldsymbol{w}_j \\
&= \lambda_j \boldsymbol{w}_m^T \boldsymbol{w}_j = 0 \ (j = 1, 2, \cdots, m-1)
\end{aligned}
$$

であることから

$$
\mu_j = 0 \ (j = 1, 2, \cdots, m-1)
$$

となる. これから最適化のための評価関数として以下の関係式が導かれる.

$$
J_m = \boldsymbol{w}_m^T C \boldsymbol{w}_m \lambda_m (\boldsymbol{w}_m^T \boldsymbol{w}_m - 1) \tag{6.2}
$$

よって, 第 1 主成分を求めるための評価関数と同じであるから, 求める第 m 主成分 $\boldsymbol{w}_m$ は分散共分散行列の C の m 番目に大きい固有値に対応した固有ベクトルとして与えられる.

- 寄与率：部分空間で表現出来る分布の広がり程度
  -元の空間における分散：σ_O^2
  -第 m 時の部分空間における分散：σ_{Sm}^2
  とすると

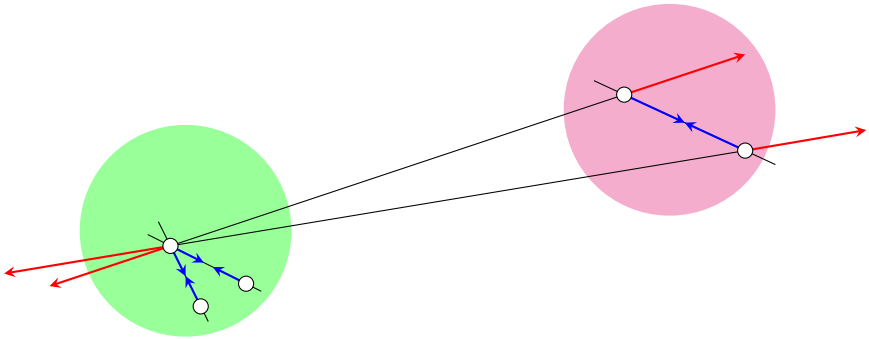
$$
\text{寄与率} = \frac{\sigma_{Sm}^2}{\sigma_O^2} \tag{6.3}
$$

  となる.
- 累積寄与率

$$
\text{累積寄与率} = \frac{\sum_{m=1}^{M} \sigma_{Sm}^2}{\sigma_O^2} \tag{6.4}
$$

6.3 線形判別分析 (LDA:Linear Discriminant Analysis

級内分散 C_W と級間分散 C_B の比を最大にする座標空間を求める.



カテゴリの同じデータは近くに, 違うデータは遠くに配置する.
2 クラスの場合の LDA について考えてみる. 2 クラスについてそれぞれクラス A, B として以下の関数について考えてみる.
$y = \boldsymbol{w}^T \boldsymbol{x}$
$\mu' = E[y],\ \sigma' = E[(y-\mu)^2]$
として

$$J(\boldsymbol{w}) = \frac{(\mu'_A - \mu'_B)^2}{\sigma'^2_A + \sigma'^2_B}$$

射影ベクトル $\boldsymbol{w}$ を変えると J の値が変わると, J が大きい時, 違うデータの集まりが遠くに配置されていることになるので, 識別が容易になる. したがって, J を最大化するベクトル $\boldsymbol{w}$ を求めることでクラスを分割することができる.
$\boldsymbol{x}$: 特徴ベクトル
n_i : クラス i のデータ数 $\left(\text{総データ数 } n = \sum_i n_i\right)$
$\boldsymbol{\mu}_i$: クラス i の平均ベクトル
X_i : クラス i に対する元空間における特徴ベクトルの集合
とおくと, C_i : クラス i の共分散行列
S_i : クラス i の変動行列
は次式で定義される.

$$C_i = \frac{1}{n_i} \sum_{\boldsymbol{x} \in X_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T$$

$$S_i = \sum_{\boldsymbol{x} \in X_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T = n_i C_i$$

また, 射影先について考える. ベクトル $\boldsymbol{w}$ に $\boldsymbol{x}$ を射影して作った 1 次元の空間において
y : 射影して作られた $\boldsymbol{x}$ の像 $y = \boldsymbol{w}^T \boldsymbol{x}$
$\tilde{\mu}_i$: 射影先の空間におけるクラス i のデータの平均値
Y_i : 射影先の空間におけるクラス i のデータの集合
$\tilde{\sigma}_i{}^2$: 射影先の空間におけるクラス i のデータの分散
$\tilde{S}_i$: 射影先の空間におけるクラス i の変動 (行列)

とすると

$$\begin{aligned}
\tilde{\sigma}_i^2 &= \frac{1}{n_i}\sum_{y\in Y_i}(y-\tilde{\mu}_i)^2 \\
&= \frac{1}{n_i}\sum_{x\in X_i}\|\boldsymbol{x}-\boldsymbol{\mu}_i\|^2 \\
&= \frac{1}{n_i}\sum_{x\in X_i}(\boldsymbol{w}^T\boldsymbol{x}-\boldsymbol{w}^T\boldsymbol{\mu}_i)(\boldsymbol{w}^T\boldsymbol{x}-\boldsymbol{w}^T\boldsymbol{\mu}_i)^T \\
&= \frac{1}{n_i}\sum_{x\in X_i}\boldsymbol{w}(\boldsymbol{x}-\boldsymbol{\mu}_i)(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\boldsymbol{w} \\
&= \boldsymbol{w}^T C_i \boldsymbol{w}
\end{aligned}$$

また, クラス i の変動については

$$\tilde{S}_i = \sum_{y\in Y_i}(y-\tilde{\mu}_i)^2 = n_i\sigma_i^2 = n_i\boldsymbol{w}^T C_i\boldsymbol{w}$$

となる. クラス内変動 S_w およびクラス間変動 S_B を以下のように定義.

$$\begin{aligned}
S_W &= S_1 + S_2 = \sum_{i=1,2}\sum x\in X_i(\boldsymbol{x}-\boldsymbol{\mu}_i)(\boldsymbol{x}-\boldsymbol{\mu}_i)^T \\
S_B &= \sum_{i=1,2} n_i(\boldsymbol{\mu}_i-\boldsymbol{\mu})(\boldsymbol{\mu}_i-\boldsymbol{\mu})^T
\end{aligned}$$

ただし, μ は全データの平均値であり, このときの射影先におけるクラス内変動 $\tilde{S}_W$, クラス間変動 $\tilde{S}_B$ は以下のようになる.

$$\begin{aligned}
\tilde{S}_W &= \tilde{S}_1 + \tilde{S}_2 = \boldsymbol{w}^T S_W \boldsymbol{w} \\
\tilde{S}_B &= \sum_{i=1,2} n_i(\tilde{\mu}_i-\tilde{\mu})^2 \\
&= n_1\left(\tilde{\mu}_1 - \frac{n_1\tilde{\mu}_1+n_2\tilde{\mu}_2}{n}\right)^2 + n_2\left(\tilde{\mu}_2 - \frac{n_1\tilde{\mu}_1+n_2\tilde{\mu}_2}{n}\right)^2 \\
&= n_1\left(\frac{n_2(\tilde{\mu}_1-\tilde{\mu}_2)}{n}\right)^2 + n_2\left(\frac{n_1(\tilde{\mu}_1-\tilde{\mu}_2)}{n}\right)^2 \\
&= \left(\frac{n_1n_2^2}{n^2}+\frac{n_1^2n_2}{n^2}\right)(\tilde{\mu}_1-\tilde{\mu}_2)^2 \\
&= \frac{n_1n_2}{n}(\tilde{\mu}_1-\tilde{\mu}_2)^2 \\
&= \boldsymbol{w}^T S_B \boldsymbol{w}
\end{aligned}$$

J を以下のように定義.

$$J = \frac{\tilde{S}_B}{\tilde{S}_W} = \frac{n_1n_2}{n}\cdot\frac{(\tilde{\mu}_1-\tilde{\mu}_2)^2}{n_1\tilde{\sigma}_1^2+n_2\tilde{\sigma}_2^2} \tag{6.5}$$

これより先の計算結果から以下の式が得られる.

$$J = \frac{\boldsymbol{w}^T S_B\boldsymbol{w}}{\boldsymbol{w}^T S_W\boldsymbol{w}}$$

この J を最大にする $\boldsymbol{w}$ を求める問題を解く. また制約条件として

$$\boldsymbol{w}^T S_W\boldsymbol{w} = 1$$

と定めるとき,

$$\boldsymbol{w}^T S_B\boldsymbol{w}$$

を最大にする問題と等価である．ここで Lagrange の未定乗数法と同様にして以下のように E を定義する．

$$E = \boldsymbol{w}^T S_B \boldsymbol{w} - \lambda(\boldsymbol{w}^T S_W \boldsymbol{w} - 1)$$

これに $\boldsymbol{w}$ で偏微分してゼロとおくと，

$$\begin{aligned} & S_B \boldsymbol{w} - \lambda S_W \boldsymbol{w} = 0 \\ \Longleftrightarrow\ & S_W^{-1} S_B \boldsymbol{w} - \lambda S_W^{-1} S_W \boldsymbol{w} = 0 \\ \Longleftrightarrow\ & (S_W^{-1} S_B - \lambda I)\boldsymbol{w} = 0 \end{aligned}$$

となる．
したがって，$S_W^{-1} S_B$ の固有値を求めればよいということがわかる．つまり $S_W^{-1} S_B$ の最大固有値を λ_1 とすれば，J の最大値を与える $\boldsymbol{w}$ は λ_1 に対する固有ベクトルとして求めることができる．

多クラスの場合の LDA について N クラスの問題は $N-1$ 軸に射影することを考える．
ベクトル $\boldsymbol{w}_i$ $(i = 1, \cdots, N-1)$ に $\boldsymbol{x}$ を射影して作った $N-1$ 次元の空間において
$\boldsymbol{y}$：射影して作られた $\boldsymbol{x}$ の像 $\boldsymbol{y} = W^T \boldsymbol{x}$, $W = \begin{pmatrix} \boldsymbol{w}_1 & \boldsymbol{w}_2 & \cdots & \boldsymbol{w}_{N-1} \end{pmatrix}$
$\tilde{\mu}_i$：射影先の空間におけるクラス i のデータの平均値
Y_i：射影先の空間におけるクラス i のデータの集合
$\tilde{C}_i$：射影先の空間におけるクラス i のデータの共分散行列
$\tilde{S}_i$：射影先の空間におけるクラス i の変動行列
とすると

$$\begin{aligned} \tilde{C}_i &= \frac{1}{n_i} \sum_{y \in Y_i} (\boldsymbol{y} - \tilde{\boldsymbol{\mu}}_i)(\boldsymbol{y} - \tilde{\boldsymbol{\mu}}_i)^T \\ &= \frac{1}{n_i} \sum_{x \in X_i} \boldsymbol{w}^T (\boldsymbol{x} - \boldsymbol{m}\boldsymbol{u}_i)(\boldsymbol{x} - \boldsymbol{m}\boldsymbol{u}_i)^T \boldsymbol{w} = \boldsymbol{w}^T C_i \boldsymbol{w} \end{aligned}$$

2 クラスのときと同様に

$$\begin{aligned} S_W &= \sum_{i=1}^{N} S_i = \sum_{i=1}^{N} \sum_{\boldsymbol{x} \in X_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T = \sum_{i=1}^{N} n_i C_i \\ S_B &= \sum_{i=1}^{N} n_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T \end{aligned}$$

また，

$$C_W = \frac{1}{n} S_W,\ C_B = \frac{1}{n} S_B$$
$$\tilde{C}_W = W^T C_W W,\ \tilde{C}_B = W^T C_B W$$

これの $\tilde{C_B}$ と $\tilde{C}_W$ の比が大きくなるように W を設定すればよい．
2 クラスのときと異なり，$\tilde{C}_B$, $\tilde{C}_W$ は行列であるから，単純な比を取る代わりに，(例えば) 次の J を評価関数と取る．

$$J = \frac{\mathrm{tr}\tilde{C}_B}{\mathrm{tr}\tilde{C}_W}$$

ここで，tr(トレース) とは，行列の対角成分の和のことである．
この関数の W による最大化は

$$W^T C_W W = I$$

となる条件で, 分子を最大化することと等価である. つまり以下の評価関数の最大化問題である.

$$E = \sum_{i=1}^{N-1} \boldsymbol{w}_i^T C_B \boldsymbol{w}_i - \sum_{i=1}^{N-1} \lambda_i (\boldsymbol{w}_i^T C_W \boldsymbol{w}_i - 1)$$

$\boldsymbol{w_i}$ で偏微分してゼロとしておくと

$$\begin{aligned} & C_B \boldsymbol{w}_i - \lambda_i C_W \boldsymbol{w}_i = 0 \\ \Longleftrightarrow & C_W^{-1} C_B \boldsymbol{w}_i - \lambda_i C_W^{-1} C_W \boldsymbol{w}_i = 0 \\ \Longleftrightarrow & (C_W^{-1} C_B - \lambda_i I) \boldsymbol{w}_i = 0 \end{aligned}$$

をえるので, 問題は $C_W^{-1} C_B$ の固有値を求める問題に帰着する.
つまり, $C_W^{-1} C_B$ の固有値の大きい方から $N-1$ 個とって, これらに対応する固有ベクトル $\boldsymbol{w}_i$, $(i = 1, 2, \cdots, N-1)$ を求めれば, これが W の各列ベクトルとなる.
データ間の距離と分散の関係は以下のようになる.

$$\begin{aligned} \sum_{i=1}^{N} \sum_{j=i+1}^{N} (x_i - x_j)^2 &= \sum_{i=1}^{N} \sum_{j=i+1}^{N} (x_i^2 - 2x_i x_j + x_j^2) \\ &= (N-1) \sum_{i=1}^{N} x_i^2 - 2 \sum_{i=1}^{N} \sum_{j=i+1}^{N} x_i x_j \\ &= N \sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i^2 + 2 \sum_{i=1}^{N} \sum_{j=i+1}^{N} x_i x_j \right) \\ &= N \sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i \right)^2 \\ &= N^2 \left(\frac{1}{N} \sum_{i=1}^{N} x_i^2 - \left(\frac{1}{N} \sum_{i=1}^{N} x_i \right)^2 \right) \\ &= N \sum_{i=1}^{N} (x_i - \bar{x})^2 \end{aligned}$$

Pair-wise の距離について
$z_i = \boldsymbol{x}_i^T \boldsymbol{a}$
として

$$\begin{aligned} E &= \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (z_i - z_j)^2 = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{a}^T (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{a} \\ &= \frac{1}{2} \boldsymbol{a}^T \left(\sum_{i=1}^{N} \sum_{j=1}^{N} (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \right) \boldsymbol{a} \\ &= \boldsymbol{a}^T \left(N \sum_{i=1}^{N} \boldsymbol{x}_i \boldsymbol{x}_i^T \right) \boldsymbol{a} - \boldsymbol{a}^T \left(\sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{x}_j \boldsymbol{x}_i^T \right) \boldsymbol{a} \\ &= \boldsymbol{a}^T \left(N \sum_{i=1}^{N} \boldsymbol{x}_i \boldsymbol{x}_i^T \right) \boldsymbol{a} - \boldsymbol{a}^T \left(\sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{x}_j \boldsymbol{x}_i^T \right) \boldsymbol{a} \\ &= \boldsymbol{a}^T X B X^T \boldsymbol{a} - \boldsymbol{a}^T X X^T \boldsymbol{a} \\ &= \boldsymbol{a}^T X C X^T \boldsymbol{a} \end{aligned}$$

であり

$$X = \begin{pmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_N \end{pmatrix}, \; B = \begin{pmatrix} b_{ij} \end{pmatrix} \; b_{ij} = \begin{cases} N & i = j \\ 0 & i \neq j \end{cases} \quad C = B - I$$

つまり, 主成分分析は, 上記 XCX^T の最大固有値に対する, 固有ベクトルを求める問題である.

i, j で決まる重み w_{ij} を導入する.

$$
\begin{aligned}
E &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}(z_i - z_j)^2 = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{a}^T(\boldsymbol{x}_i - \boldsymbol{x}_j)w_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j)^T\boldsymbol{a} \\
&= \boldsymbol{a}^T\left(\sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{x}_i w_{ij}\boldsymbol{x}_i^T\right)\boldsymbol{a} - \boldsymbol{a}^T\left(\sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{x}_i w_{ij}\boldsymbol{x}_j^T\right)\boldsymbol{a} \\
&= \boldsymbol{a}^T X D X^T\boldsymbol{a} - \boldsymbol{a}^T X W X^T\boldsymbol{a} = \boldsymbol{a}^T X L X^T\boldsymbol{a} \\
D &= \left(d_{ij}\right),\ d_{ij} = \begin{cases} \sum_{k=1}^{N} w_{ik} & i = j \\ 0 & i \neq j \end{cases} \quad W = \left(w_{ij}\right)\ L = D - W \\
F &= \frac{\boldsymbol{a}^T X L X^T\boldsymbol{a}}{\boldsymbol{a}^T X D X^T\boldsymbol{a}}
\end{aligned}
$$

ここで

$$
w_{ij} = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{t}\right)
$$

のように, w_{ij} を $\boldsymbol{x}_i$ と $\boldsymbol{x}_j$ との距離が大きいときに小さく, 距離が小さいときに 1 をとって, F を最小化することで, つまり $(XDX^T)^{-1}XLX^T$ の最小固有値に対する固有ベクトルとして $\boldsymbol{a}$ を定めれば, 変換前の座標系で近くに分布しているデータ同士の写像後の距離を小さくすることが出来る.

$\Rightarrow$ Locality Preserving Projection

重みつき分散の比の導入について

$$
\begin{aligned}
E &= \frac{\sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}(z_i - z_j)^2}{\sum_{i=1}^{N}\sum_{j=i+1}^{N} v_{ij}(z_i - z_j)^2} \\
&= \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{a}^T(\boldsymbol{x}_i - \boldsymbol{x}_j)w_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j)^T\boldsymbol{a}}{\sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{a}^T(\boldsymbol{x}_i - \boldsymbol{x}_j)v_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j)^T\boldsymbol{a}} \\
&= \frac{\boldsymbol{a}^T X D X^T\boldsymbol{a} - \boldsymbol{a}^T X W X^T\boldsymbol{a}}{\boldsymbol{a}^T X G X^T\boldsymbol{a} - \boldsymbol{a}^T - \boldsymbol{a}^T X V X^T\boldsymbol{a}} \\
&= \frac{\boldsymbol{a}^T X L X^T\boldsymbol{a}}{\boldsymbol{a}^T X M X^T\boldsymbol{a}}
\end{aligned}
$$

とし,

$$
\begin{aligned}
&G_{ij} = \left(g_{ij}\right) \quad g_{ij} = \begin{cases} \sum_{k=1}^{N} v_{ik} & i = j \\ 0 & i \neq j \end{cases} \\
&V_{ij} = \left(v_{ij}\right) \\
&M = G - V
\end{aligned}
$$

ここで,

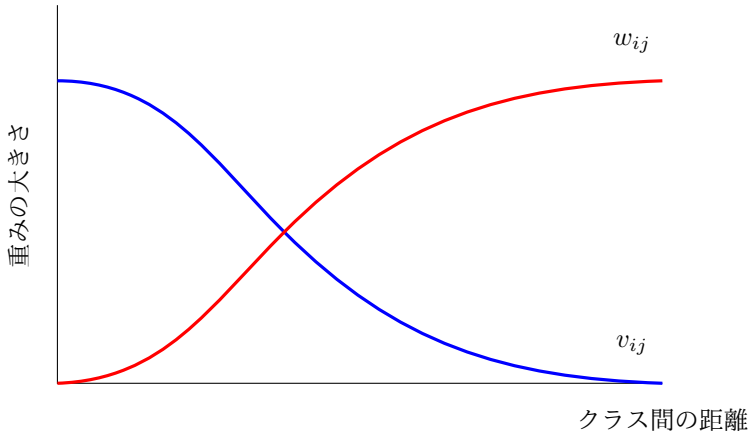
$$w_{ij} = \begin{cases} 1 & \boldsymbol{x}_i と \boldsymbol{x}_j が違うクラス \\ 0 & \boldsymbol{x}_i と \boldsymbol{x}_j が同じクラス \end{cases}$$

$$v_{ij} = \begin{cases} 1 & \boldsymbol{x}_i と \boldsymbol{x}_j が違うクラス \\ 0 & \boldsymbol{x}_i と \boldsymbol{x}_j が同じクラス \end{cases}$$

として, E を最大化すれば, LDA と等価である.

6.4 重み付き分散比と CDDA(クラス間距離を考慮した判別分析)

いくつかの多クラスの問題では, クラス間の距離に応じて, データを分布させたいことがある
例えば, 年齢分布の 30 歳のカテゴリの横に 31 歳のカテゴリを分布させたいなどである.



CDDA : Class Distance-based Discriminant Analysis
クラスが遠いデータは離れようとする. ただし, 十分離れていれば差はつけないというものである.
クラスが近いデータは近づこうとする. ただし, 十分近いデータは差をつけないというものである.