第４章 パーセプトロン：分類アルゴリズムの基礎

熊谷佳奈

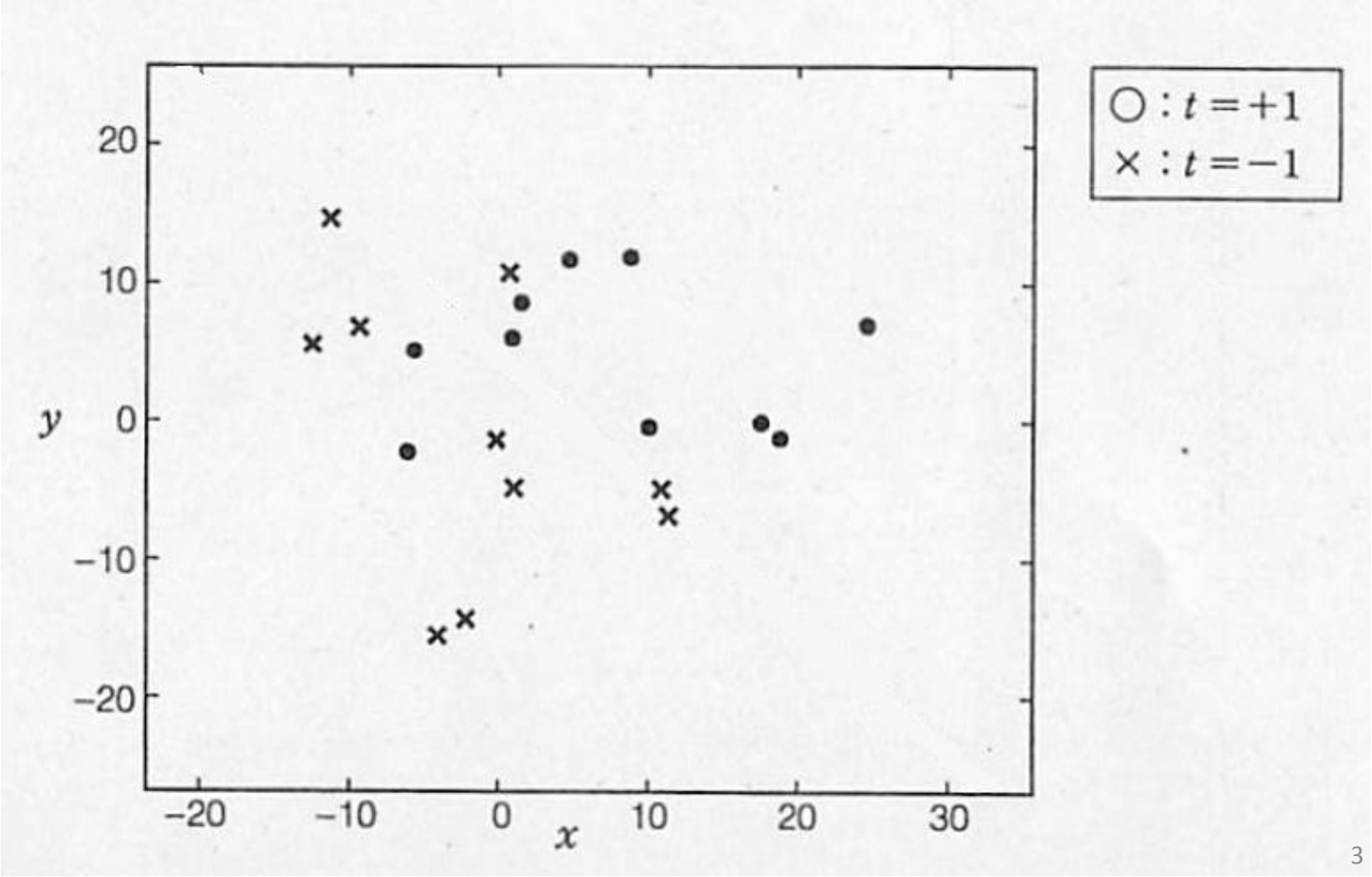
パーセプトロン

- データをいくつかのクラスに分ける
- 例題２では線形判別による新規データの分類を行うことが目的である

例題２

- (x, y)平面上にプロットされた多数のデータがある
- それぞれのデータは属性値$t = \pm 1$を持つ
- つまりN個のデータ$\{(x_n, y_n, t_n)\}_{n=1}^{N}$が与えられている状況である
- 属性値tを推測する(x, y)平面上の直線を見つけることが目的である

例題2



確率的勾配降下法のアルゴリズム

(1)パラメーターを含むモデル(数式)を設定する

(2)パラメーターを評価する基準を定める

(3)最良の評価を与えるパラメーターを決定する

平面を分割する直線の方程式(1)

- 直線の式といえば$y = ax + b$である
- ここではxとyを対等に扱うために次の一次関数を用いる

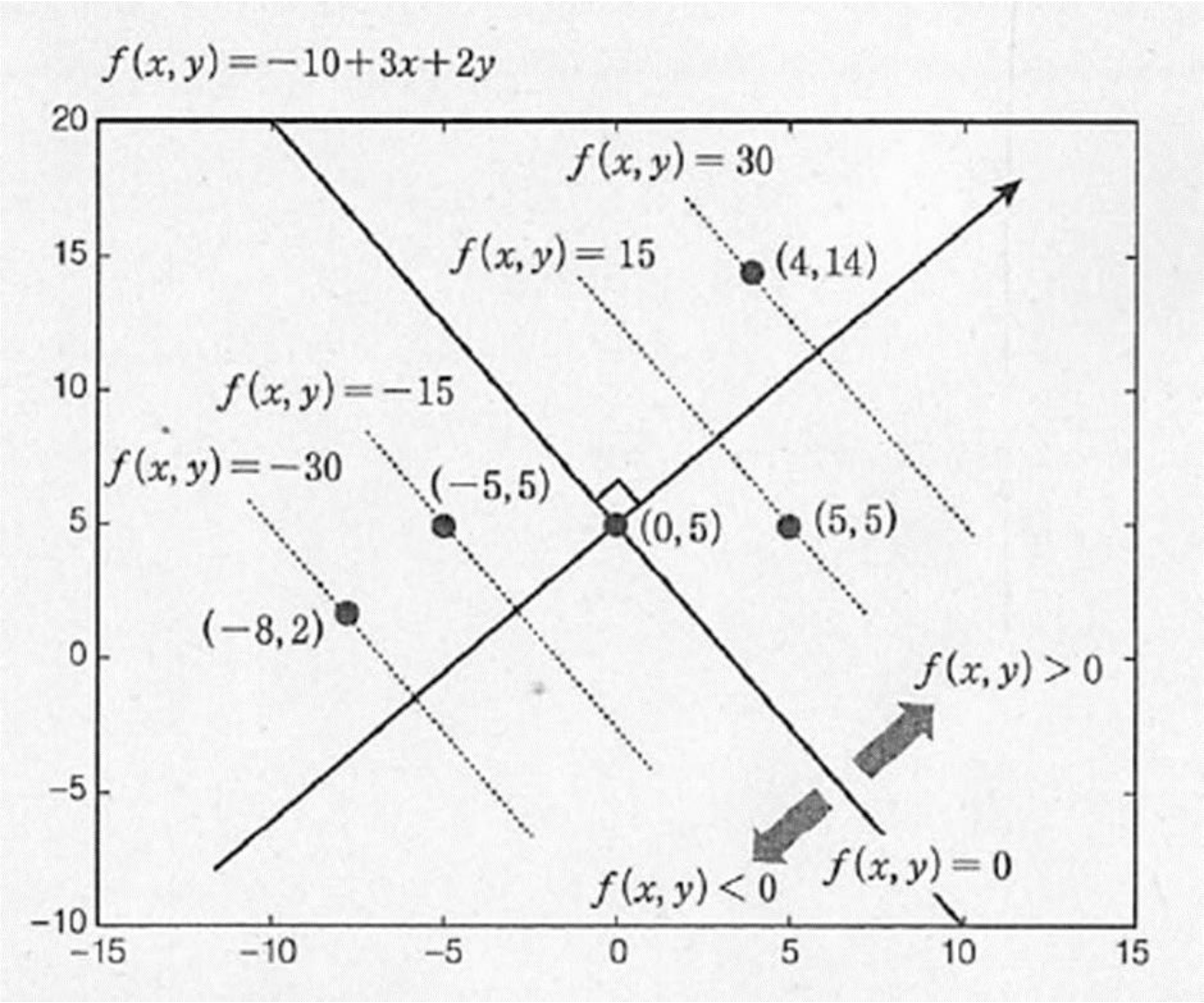
$$f(x, y) = w_0 + w_1 x + w_2 y$$

- この時、(x, y)平面を分割する直線は$f(x, y) = 0$で表される

平面を分割する直線の方程式(2)

- 分割された2つの領域は、$f(x, y)$の符号で判別できるようになる
- 境界線から離れるほど、$f(x, y)$の絶対値が大きくなる

具体例


$f(x,y) = -10+3x+2y$
$f(x,y) = 30$
$f(x,y) = 15$
$(4,14)$
$f(x,y) = -15$
$f(x,y) = -30$
$(-5,5)$
$(0,5)$
$(5,5)$
$(-8,2)$
$f(x,y) > 0$
$f(x,y) < 0$
$f(x,y) = 0$
20
15
10
5
0
−5
−10
−15
−10
−5
0
5
10
15

平面を分割する直線の方程式(3)

- 次のルールで、データを分類する

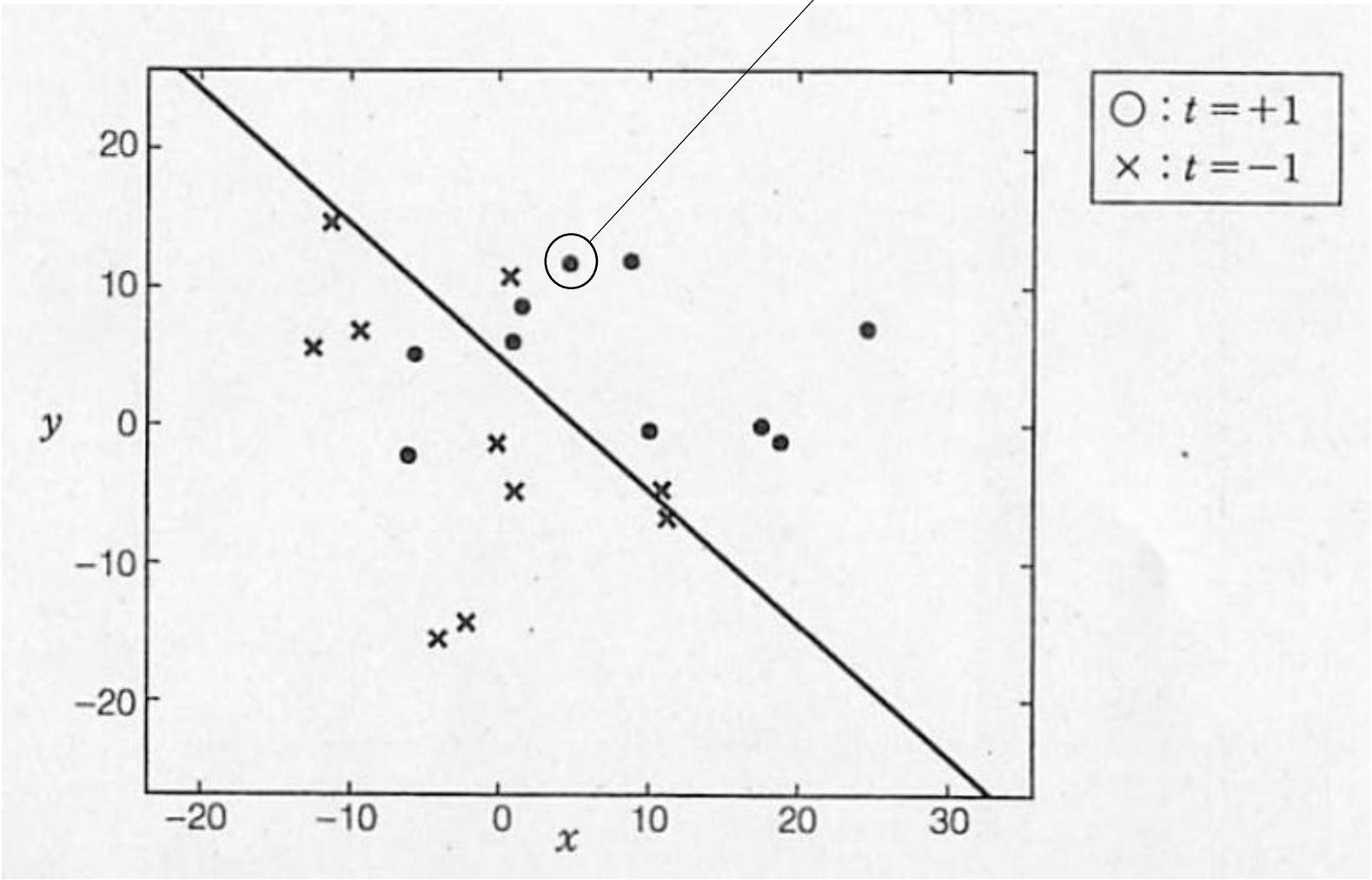
$$f(x,y) > 0 \Rightarrow t = +1$$
$$f(x,y) < 0 \Rightarrow t = -1$$

- この時、データ$\{(x_n, y_n, t_n)\}_{n=1}^{N}$について、正しく分類されているかどうかは次のルールで判定できる

$$f(x_n, y_n) \times \mathrm{t_n} > 0 : \text{正　解}$$
$$f(x_n, y_n) \times \mathrm{t_n} \leq 0 : \text{不正解}$$

例

$f(x, y) > 0, t = +1$ よって、
$f(x, y) \times \mathrm{t} > 0$: 正解 となる



平面を分割する直線の方程式(4)

- 全てのデータ(x_n, y_n, t_n)が正解となるような直線を見つけたい
- つまり$f(x, y) = w_0 + w_1 x + w_2 y$の係数$(w_0, w_1, w_2)$を見つけたい

分類結果の評価基準(1)

- 不正解の点を誤差として計算する
- この誤差の合計値が小さいほど、正しい分類に近い
- 「境界線から離れるほど、$f(x, y)$の絶対値が大きくなる」という特徴を利用する

分類結果の評価基準(2)

- 分類の誤差の合計値Eは次の式で求められる

$$E = \sum_n E_n = \sum_n |f(x_n, y_n)|$$

- また、この式は$f(x_n, y_n) \times t_n \leq 0$を満たすため、次の関係式が成り立つ

$$|f(x_n, y_n)| = -f(x_n, y_n) \times t_n$$

分類結果の評価基準(3)

- したがってEは次のように表される

$$E = -\sum_{n}(w_0 + w_1 x_n + w_2 y_n)$$

- すべてのデータが正しく分類できた場合は$E = 0$になる

分類結果の評価基準(4)

- また、Eはベクトルを用いると、次のように表せる

$$E = -\sum_{n} t_n w^T \phi_n$$

$$w = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix}$$

$$\phi_n = \begin{pmatrix} 1 \\ x_n \\ y_n \end{pmatrix}$$

パラメーターの修正(1)

- 誤差Eの偏微分係数を0と置いてみる

$$\frac{\partial E}{\delta w_l} = 0 \quad (l = 0,1,2)$$

- または、勾配ベクトルが0になるとする

$$\nabla E(w) = -\sum_n t_n \phi_n = 0$$

- この式には係数wが含まれていないため、変形してもwを求めることは不可能である

パラメーターの修正(2)

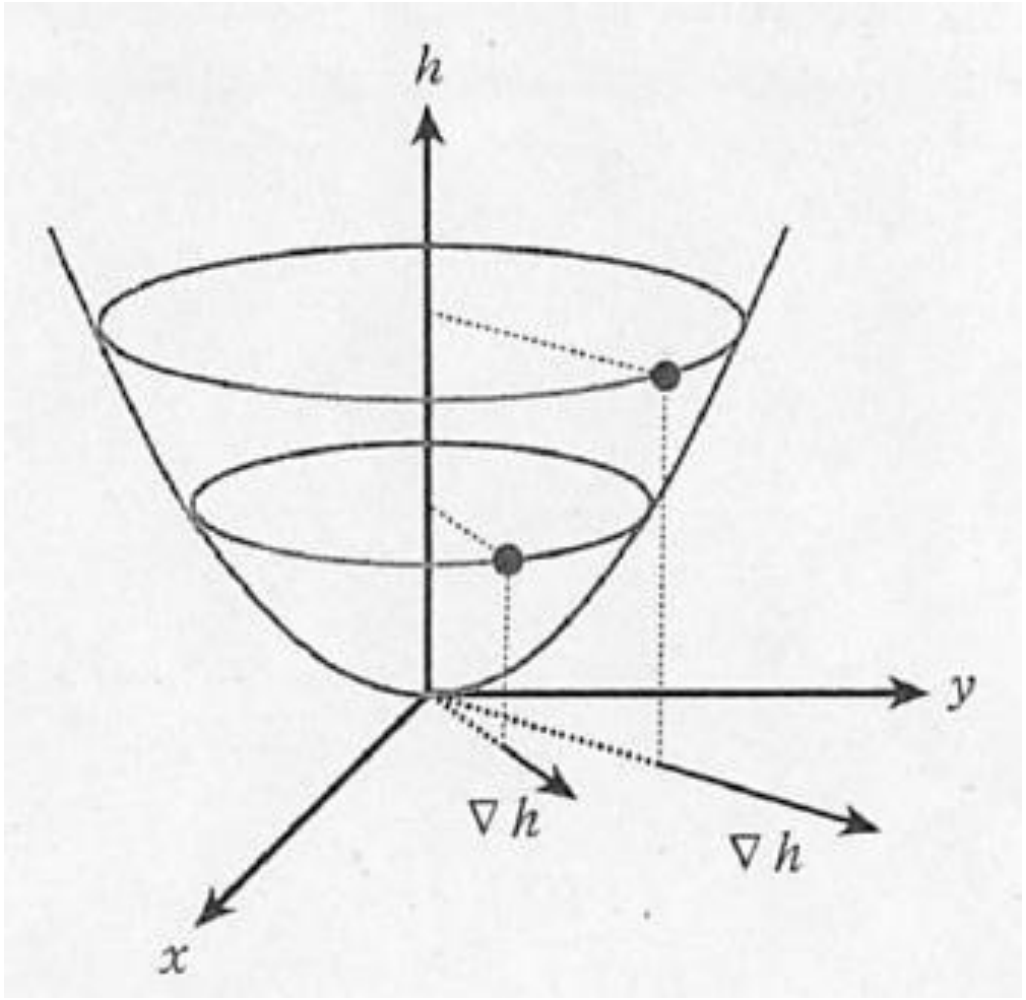
- したがって数値計算を用いて、wを修正しながら誤差Eがなるべく小さくなるものを求める
- 勾配ベクトルの幾何学的な性質を利用する

具体例

- $h(x,y) = \frac{3}{4}(x^2 + y^2)$
- この場合、勾配ベクトルは次のようになる

$$\nabla h = \begin{pmatrix} \frac{\partial h}{\delta x} \\ \frac{\partial h}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{3}{2}x \\ \frac{3}{2}y \end{pmatrix}$$

具体例



パラメーターの修正(3)

- この時、任意の点(x, y)において、どちらかの方向に少し移動すると、hの値が増減する
- 勾配ベクトル∇hはh の値が最も増加する方向を表す
- つまり「斜面をまっすぐに這い上がる方向」を表す

パラメーターの修正(4)

- 各点における勾配ベクトルの方向に移動していけば、$h(x, y)$の値はどんどん大きくなる
- この場合、勾配ベクトルの反対方向$(-\nabla h)$に進めば、原点$(0,0)$に近づき$h(x, y)$の値は小さくなる

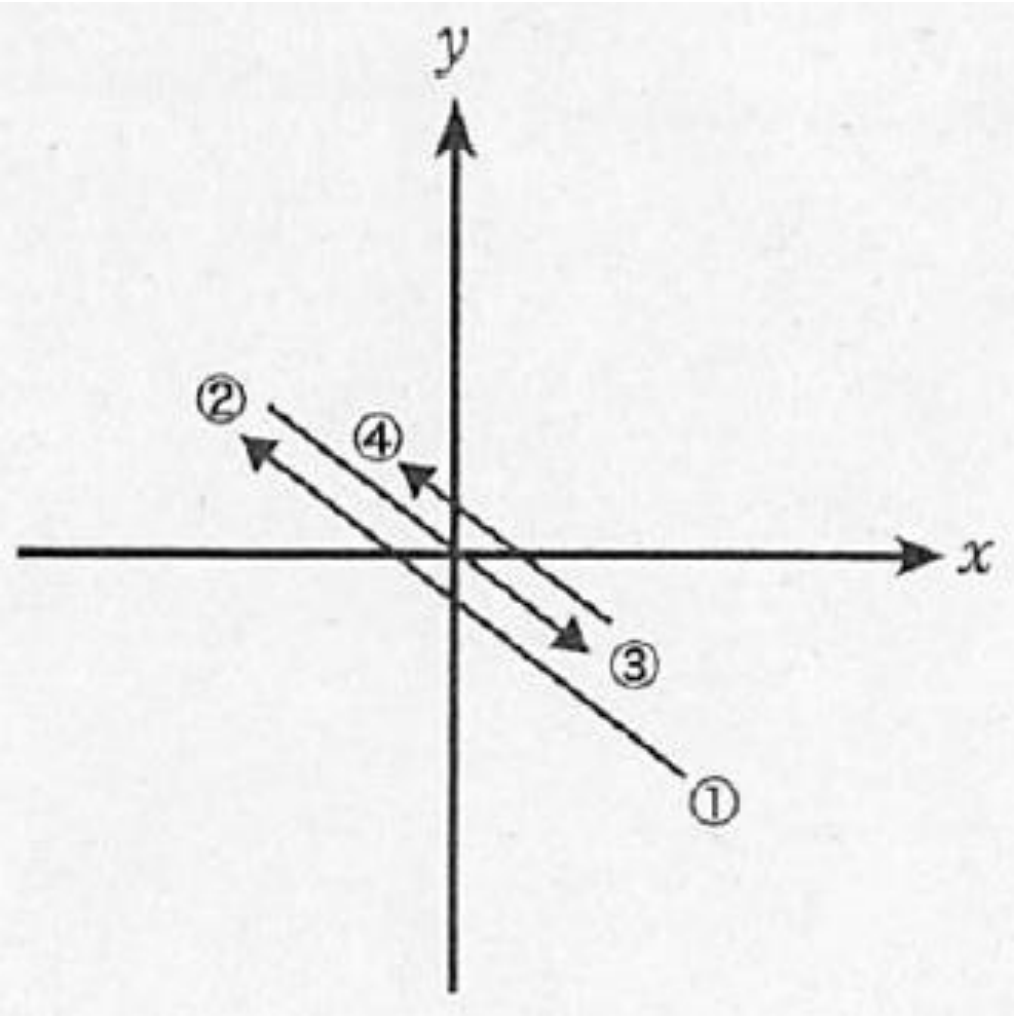
パラメーターの修正(5)

- これは次の居場所を決定するアルゴリズムとして次のように表せる

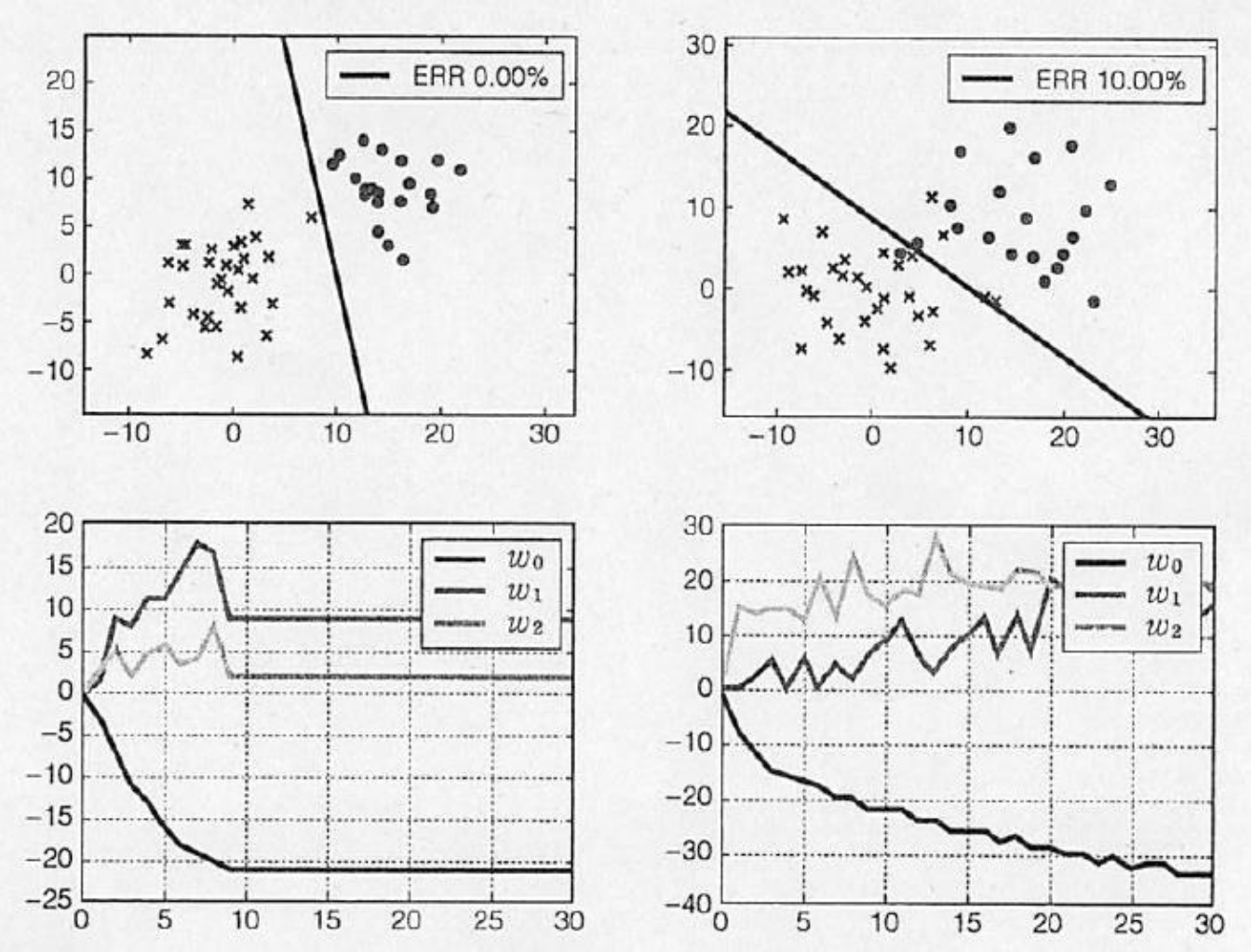
$$\mathrm{x}_{new} = \mathrm{x}_{old} - \nabla h$$

- この式に従ってxを何度も更新することで徐々に原点に近づいていく

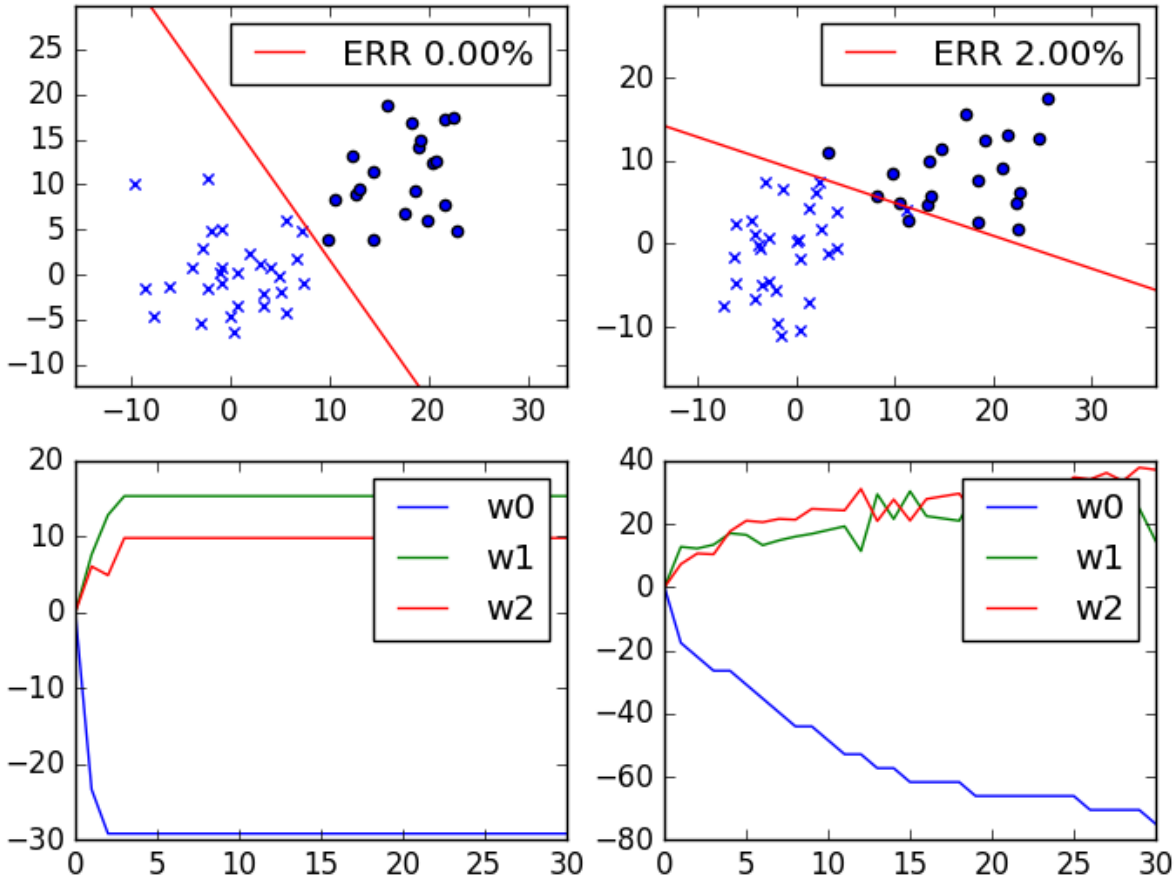
パラメーターの修正(6)



サンプルコード(1)



サンプルコード(2)



確率的勾配降下法の「収束速度」

- データを完全に分割する直線が存在する場合、パラメーターwを更新していくとその直線にたどり着く
- しかし、たどり着くまでの更新回数は不明である
- どの程度すばやく正解にたどり着けるかという速さを「アルゴリズムの収束速度」という

バイアス項と収束速度(1)

- (x, y)平面を直線で分割するための関数$f(x, y)$を次のように定義

$$f(x, y) = 2w_0 + w_1 x + w_2 y$$

- w_0の定義を$\frac{1}{2}$倍だけ変更したと考える

バイアス項と収束速度(2)

- この時、誤差関数Eは次のように表される

$$E = -\sum_n t_n w^T \phi_n$$

$$w = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix}$$

$$\phi_n = \begin{pmatrix} 2 \\ x_n \\ y_n \end{pmatrix}$$

バイアス項と収束速度(3)

- 誤差関数の形は変わりないため、先の手順と同様にして次の確率的勾配降下法の手続きが得られる

$$w_{new} = w_{old} + t_n \phi_n$$

- ただしベクトルϕ_nはバイアス項を2にしたものを用いる
- したがってt_nは± 1の値をとり、更新するとw_0は± 2だけ変化する

バイアス項と収束速度(4)

- 先の手順を一般化する
- cを任意の定数として、次のように定義できる

$$f(x, y) = w_0 c + w_1 x + w_2 y$$

$$\phi_n = \begin{pmatrix} c \\ x_n \\ y_n \end{pmatrix}$$

- このcを適切に選ぶことでアルゴリズムの収束速度を改善できる(後述)

パラメーターの更新(1)

- (x, y)平面の原点を通る直線でデータを分類する場合について考える
- その直線でデータを完全に分類できると、最初から分かっているものとする
- この場合、次のように仮定できる

$$f(x, y) = w_1 x + w_2 y$$

パラメーターの更新(2)

- したがって、誤差関数Eは次のように表せる

$$E = -\sum_n t_n w^T \phi_n$$

$$w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

$$\phi_n = \begin{pmatrix} x_n \\ y_n \end{pmatrix}$$

パラメーターの更新(3)

- すると確率的勾配降下法の手続きは、次のようになる

$$w_{new} = w_{old} + t_n \phi_n$$

- wとϕ_nが2次元のベクトルになっている

パラメーターの更新(4)

- さらに、直線$f(x, y) = 0$は次のように表せる

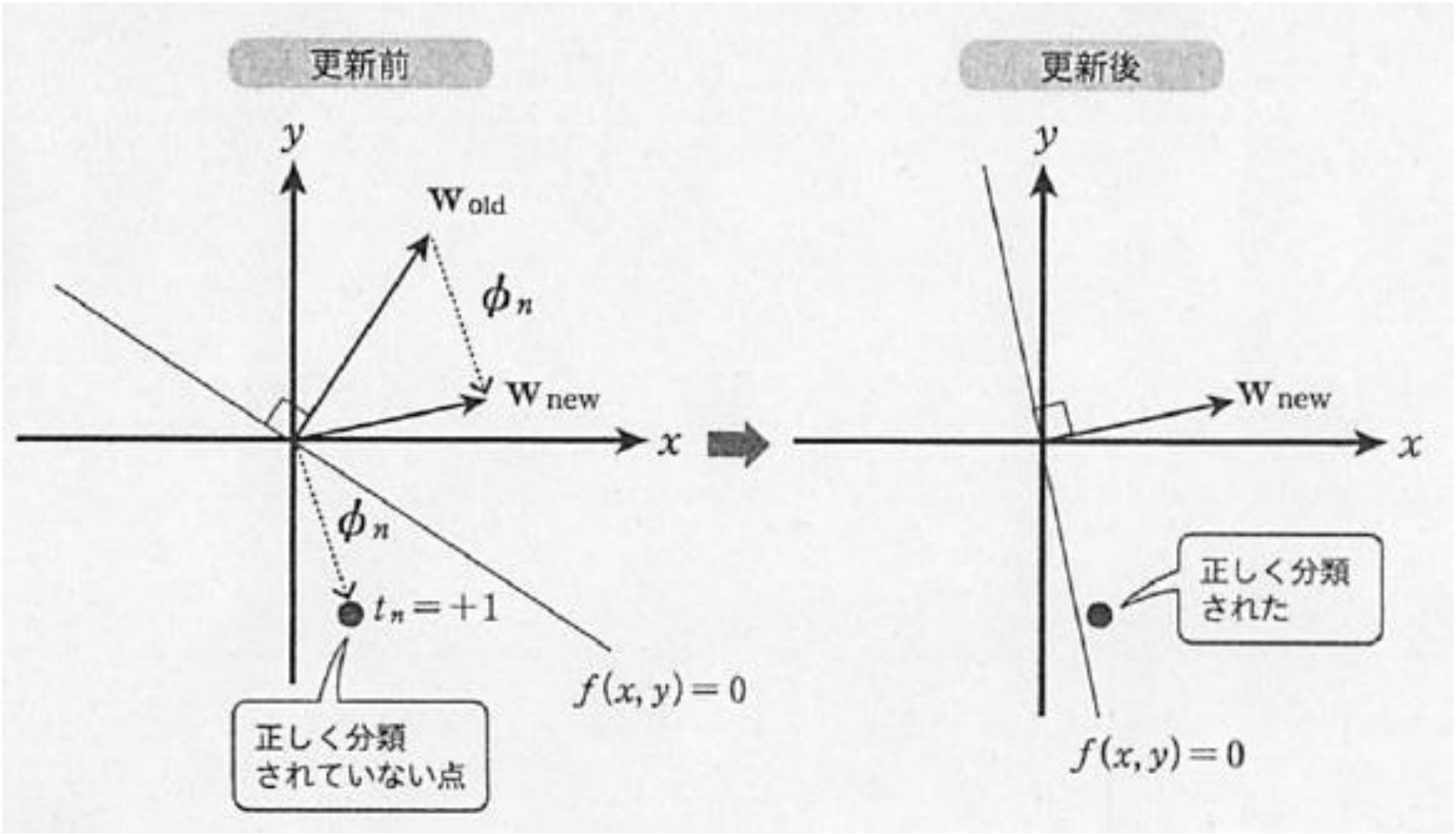
$$w^T \mathrm{x} = 0$$

- これは、原点から直線上の点(x, y)に向かうベクトルxとベクトルwが直交していることを示す
- つまりwは直線$f(x, y) = 0$に直交する法線ベクトルになっている

例

- $f(x, y) < 0$の領域に$t_n = +1$の点が存在している
- つまり、正しく分類できていない点(x_n, y_n)が存在している
- ϕ_nは原点から点(x_n, y_n)に向かうベクトルになっている
- 法線ベクトルwをϕ_nの方向に修正する

例


更新前
更新後
y
x
$\mathbf{w}_{\text{old}}$
ϕ_n
$\mathbf{w}_{\text{new}}$
$t_n = +1$
$f(x, y) = 0$
正しく分類
されていない点
正しく分類
された

３次元への拡張(1)

- データを分類する直線が原点を通るとは限らない場合について考える
- 先の手順を3次元に拡張する
- つまり3次元の(x, y, z)空間に散らばった、$t_n = \pm 1$のデータ群を「原点を通る平面」で分類する

３次元への拡張(2)

- 平面の式は次の式で表せる

$$f(x, y, z) = w_0 z + w_1 x + w_2 y$$

- 原点を通る平面は$f(x, y, z) = 0$と表せる

３次元への拡張(3)

- 誤差関数Eを次のように定義する

$$E = -\sum_n t_n w^T \phi_n$$

$$w = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix}$$

$$\phi_n = \begin{pmatrix} z_n \\ x_n \\ y_n \end{pmatrix}$$

３次元への拡張(4)

- するとこれまでの手順と同様に、パラメーターを修正する手続きは次のようになる

$$w_{new} = w_{old} + t_n \phi_n$$

- この場合、ベクトルwは平面$f(x, y, z) = 0$に直交する法線ベクトルなる
- 法線ベクトルの方向を修正することで、データを分類する平面の向きを修正していくものと考えられる

バイアス項の修正(1)

- さらに、データがすべて$z_n = c$になっている場合について考える
- つまり$z = c$の平面上にすべてのデータが乗っているような状況である
- この時、先の手続きは、バイアス項をcとおいたパーセプトロンのアルゴリズムと同じものになる

バイアス項の修正(2)

- この時、データが乗っている平面を分割する直線は次式で表される

$$f(x, y) = w_0 c + w_1 x + w_2 y = 0$$

- 正しくデータを分類する直線が原点の遠くを通る場合、収束速度の改善にはバイアス項の修正が必要となる

バイアス項の修正(3)