統計的機械学習論
4. パーセプトロン

夏季集中講義(2019/09/02-13)
@九州工業大学
(飯塚キャンパス 大学院セミナー室７F)

このクラスのねらいと達成目標

ねらい

パーセプトロンの講義を通じて分類アルゴリズムの考え方を理解する。

達成目標

- 分類アルゴリズムの考えかたを理解する
- 分類アルゴリズムで使われる確率的勾配降下法を理解する
- パーセプトロンが数値解析によって最適解を求める問題であることを理解する

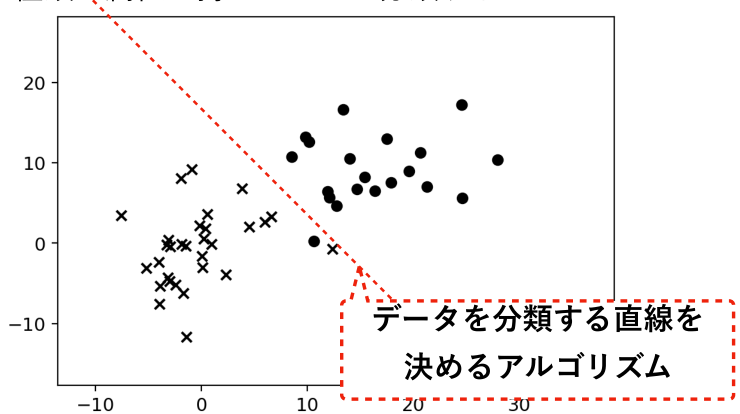
クラスの進行

4. パーセプトロン
    - 4.1.確率的勾配降下法のアルゴリズム
        - 4.1.1.平面を分割する直線の方程式
        - 4.1.2.誤差関数による分類結果の評価
        - 4.1.3.勾配ベクトルによるパラメータの修正
        - 4.1.4.サンプルコードによる確認
    - 4.2.パーセプトロンの幾何学的な解釈
        - 4.2.1.バイアス項の任意性とアルゴリズムの収束速度
        - 4.2.2.パーセプトロンの幾何学的な解釈
        - 4.2.3.バイアス項の幾何学的な意味

4 パーセプトロン

分類アルゴリズムの基礎

例題：２種類の属性を持つデータを分類する



4.1 確率的勾配降下法のアルゴリズム

パラメトリックモデルの３つのステップに従い解説していく

パラメトリックモデルの３つのステップ

パラメータを含む数式モデルを設定する

↓

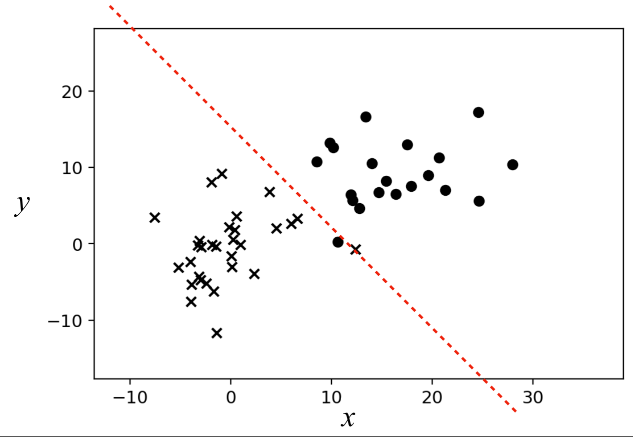
パラメータを評価する基準を定める

↓

最良の評価を与えるパラメータを決定する

4.1.1 平面を分割する直線の方程式

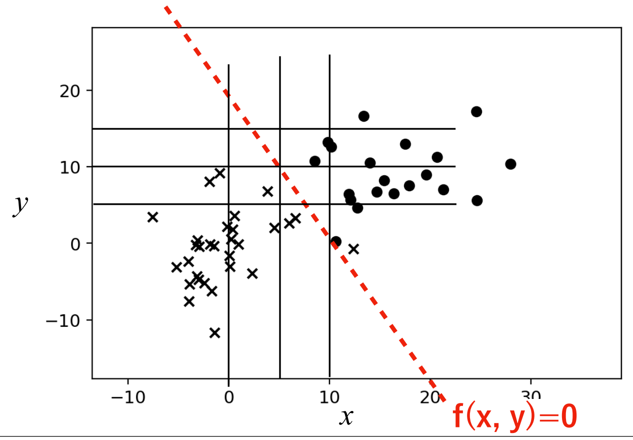
$$f(x, y) = w_0 + w_1 x + w_2 y$$



4.1.1 平面を分割する直線の方程式

xy平面を分割する直線を考える

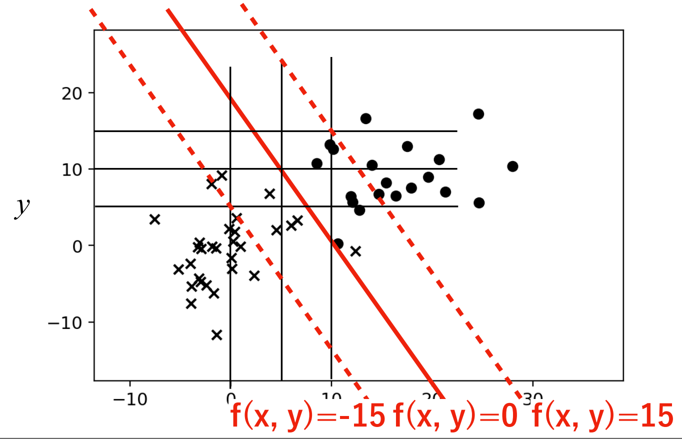
例：$f(x, y) = -20 + 2x + y$



4.1.1 平面を分割する直線の方程式

xy平面を分割する直線を考える

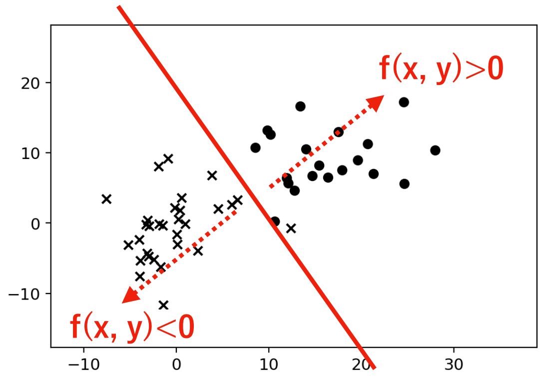
例：$f(x, y) = -20 + 2x + y$



4.1.1 平面を分割する直線の方程式

分類の仕方

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



4.1.1 平面を分割する直線の方程式

トレーニングセット

$$\{(x_n, y_n, t_n)\}_{n=1}^{N}$$

t=1である確率を正解とする

$$f(x,y) \times t_n > 0 \rightarrow true$$
$$f(x,y) \times t_n \leq 0 \rightarrow false$$

パラメータ(w_0, w_1, w_2)の評価基準を導入する

4.1.2 誤差関数による分類結果の評価

直線からの距離を以下のように表現する

$$E_n = |f(x_n, y_n)|$$

正しく分類できなかったデータについて直線からの差を加算し分類の誤差とする。

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

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

$$E = -\sum_n (w_0 + w_1 x + w_2 y) t_n$$

4.1.2 誤差関数による分類結果の評価

誤差をベクトルで表現

$$E = -\sum_n t_n \mathbf{w}^{\mathrm{T}} \phi_n$$

係数

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

データの座標

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

バイアス項

4.1.3 勾配ベクトルによるパラメータの修正

式の中に係数がないため、式変形で係数を求めるのは不可能。

誤差を係数で微分

$$\frac{\partial E}{\partial w_m} = 0 \qquad (m = 0, 1, 2)$$

誤差の微分をベクトルで表現

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

$$\nabla E(\mathbf{w}) = \begin{pmatrix} \frac{\partial E}{\partial w_0} \\ \frac{\partial E}{\partial w_1} \\ \frac{\partial E}{\partial w_2} \end{pmatrix} = \mathbf{0}$$

4.1.3 勾配ベクトルによるパラメータの修正

例

$$h(x, y) = \frac{3}{4}(x^2 + y^2)$$

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

勾配ベクトルの反対方向に進めば関数の値は小さくなる

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

4.1.3 勾配ベクトルによるパラメータの修正

D

誤差の勾配ベクトルの反対方向

$$-\nabla E(\mathbf{w}) = \sum_n t_n \phi_n$$

勾配ベクトルの反対方向に進むことで最小値にたどり着くことができる「勾配降下」

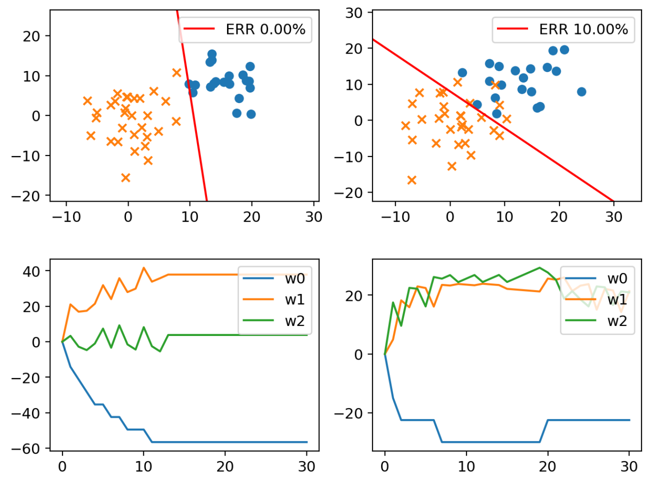
正しく分類できなかったデータのうち、1つだけサンプリングする=「確率的」

$$\mathbf{w}_{new} = \mathbf{w}_{old} + \mathbf{t}_n \boldsymbol{\phi}_n$$

確率的勾配効果法

4.1.3 サンプルコードによる確認

完全分類可能　　完全分類不可能



4.2 パーセプトロンの幾何学的な解釈

確率的勾配降下法の「収束速度」について考える。

wを何回更新すれば正解にたどりつくかわからない

正解にたどり着くまでの速さも
アルゴリズムの性能として評価される。

アルゴリズムの収束速度

4.2.1 バイアス項の任意性とアルゴリズムの収束速度

分割する直線は

直線の方程式

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

収束速度を向上させるため、方程式を以下のように変える。

直線の方程式

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

誤差

$$E = -\sum_n t_n \mathbf{w}^{\mathrm{T}} \phi_n \qquad \mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix} \qquad \phi = \begin{pmatrix} 2 \\ x \\ y \end{pmatrix}$$

4.2.1 バイアス項の任意性とアルゴリズムの収束速度

分割する直線は

直線の方程式

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

収束速度を向上させるため、方程式を以下のように変える。

直線の方程式

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

誤差

$$E = -\sum_n t_n \mathbf{w}^{\mathrm{T}} \phi_n \qquad \mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix} \phi_n = \begin{pmatrix} 2 \\ x_n \\ y_n \end{pmatrix}$$

4.2.1 バイアス項の任意性とアルゴリズムの収束速度

バイアス項が倍になる分、更新した係数の変化も大きい。

$$\mathbf{w}_{new} = \mathbf{w}_{old} + \mathbf{t}_n \boxed{\phi_n}$$

バイアス項の係数をここで引き受け

もし任意の定数をバイアス項の係数にする場合は

$$f(x,y) = cw_0 + w_1 x + w_2 y \qquad \phi_n = \begin{pmatrix} c \\ x_n \\ y_n \end{pmatrix}$$

係数の更新時、w_0は±cだけ変化する。cの値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。

4.2.1 バイアス項の任意性とアルゴリズムの収束速度

<サンプルコードのcの値>

$$c = \frac{1}{2N}\sum_{n=1}^{N}(x_n + y_n)$$

トレーニングセットのサイズが大きい場合、w_1とw_2の値の変化にバイアス項w_0の値が追いつけない。

そこで、バイアス項を(x_n, y_n)の平均的な値と同じにしてw_0の変化をw_1とw_2の変化に合わせる。

f(x,y)が原点を通る時、正解は$w_0 = 0$なので計算を始めるときw_0=1でも早く収束する。

4.2.2 パーセプトロンの幾何学的な解釈

xy平面の原点を通る直線(w_0)で考える。

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

誤差は

$$E = -\sum_n t_n \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}_n$$

$$\boldsymbol{\phi}_n = \begin{pmatrix} x_n \\ y_n \end{pmatrix} \quad \mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

4.2.2 パーセプトロンの幾何学的な解釈

xy平面の原点を通る直線(w_0)で考える。

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

誤差は

$$E = -\sum_n t_n \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}_n$$

$$\boldsymbol{\phi}_n = \begin{pmatrix} x_n \\ y_n \end{pmatrix} \quad \mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

4.2.2 パーセプトロンの幾何学的な解釈

係数の更新は

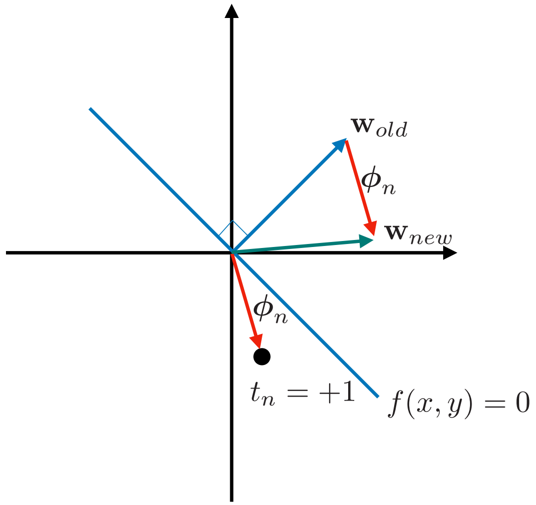
$$\mathbf{w}_{new} = \mathbf{w}_{old} + t_n \boldsymbol{\phi}$$

ここでxを原点から直線上の点(x, y)に向かうベクトルとすると、直線の式は

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

これはwとxが直交していることを示している。

4.2.2 パーセプトロンの幾何学的な解釈



4.2.3 バイアス項の幾何学的な意味

f(x, y)が原点を通らないときにはどうやってアルゴリズムの収束時間を短縮すれば良いだろうか

3次元空間上にちらばったデータ群を原点を通る平面で考える

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

このときの誤差関数は

$$E = -\sum_n t_n \mathbf{w}^{\mathrm{T}} \phi_n \qquad \mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix} \quad \phi_n = \begin{pmatrix} z_n \\ x_n \\ y_n \end{pmatrix}$$

4.2.3 バイアス項の幾何学的な意味

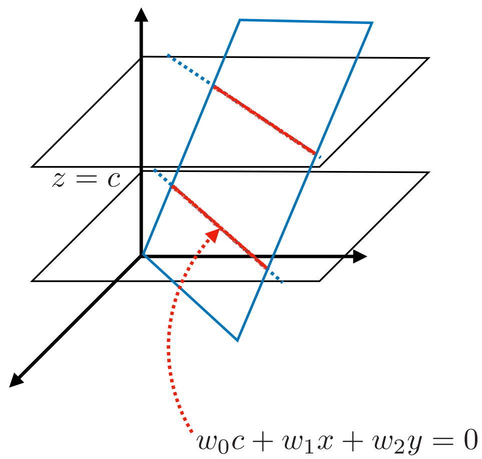
パラメータを修正する手続きは、

$$\mathbf{w}_{new} = \mathbf{w}_{old} + t_n \phi$$

となり係数も平面f(x, y, z)=0に直交する法線ベクトルとなる

ここで、トレーニングセットとして与えられているデータが全てz_n=cであるような特別な場合について考える(z=cの平面に全てのデータが乗っている)

4.2.3 バイアス項の幾何学的な意味



まとめ

- 分類の機械学習アルゴリズムの基礎を理解した
- パーセプトロンにおける誤差の評価の仕方を把握した
- 誤差を最小にするための数値計算の手続きを理解した
- アルゴリズムの収束時間を短縮するためにバイアス項を調整することを理解した

ミニッツペーパー

- パーセプトロンにおける誤差（正しく分類できなかった時のペナルティ）をどう計算しているか説明してください。
- 確率的勾配降下法について
  - なぜ確率的と呼ばれているのか説明してください。
  - なぜ勾配ベクトルとは逆の向きに更新するのか説明してください。