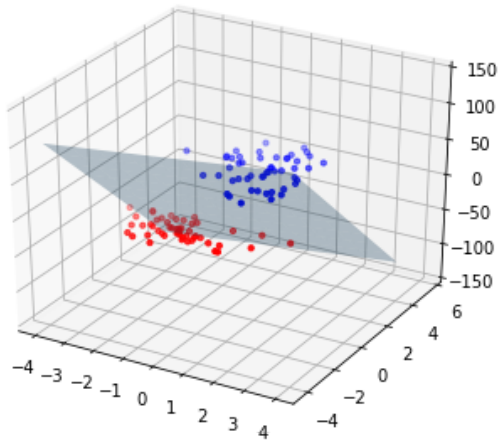
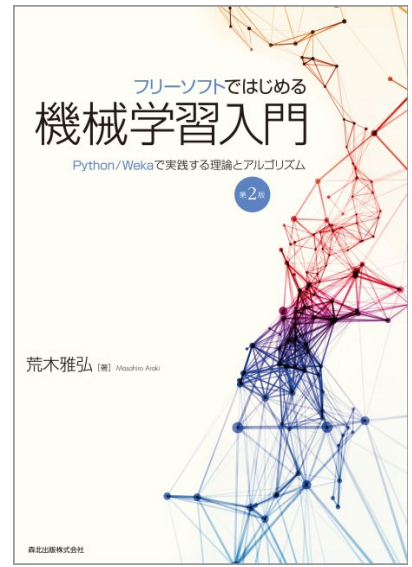
5. 識別　ー生成モデルと識別モデルー



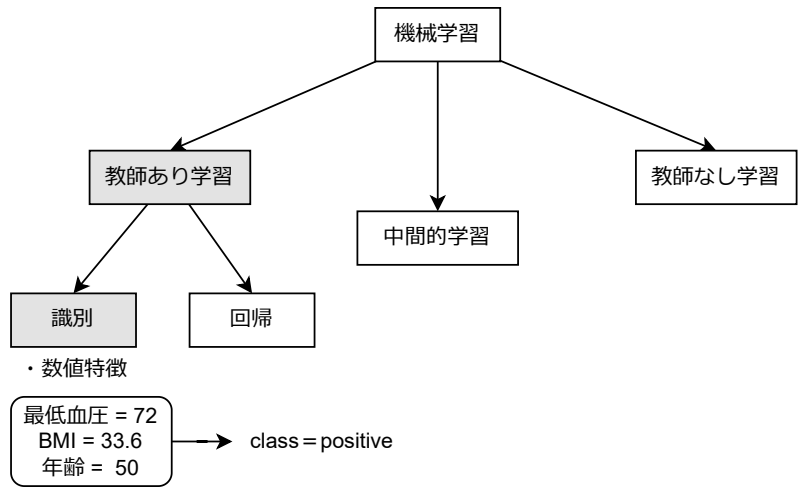
- 5.1 数値特徴に対する「教師あり・識別」問題の定義
- 5.2 生成モデル
- 5.3 識別モデル



- 荒木雅弘:『フリーソフトではじめる機械学習入門(第2版)』(森北出版, 2018年)
- スライドとJupyter notebook
- サポートページ

5. 識別 ー生成モデルと識別モデルー

- 問題設定
  - 教師あり学習
  - 数値入力 → カテゴリ出力



5.1 数値特徴に対する「教師あり・識別」問題の定義（1/5）

- 識別問題のデータ
  - 特徴ベクトル $\boldsymbol{x}$ と正解情報 y のペア

$$\{(\boldsymbol{x}_i, y_i)\}, \quad i = 1, \ldots, N$$

  - $\boldsymbol{x}$ は要素が数値である d 次元の固定長ベクトル、y はカテゴリ

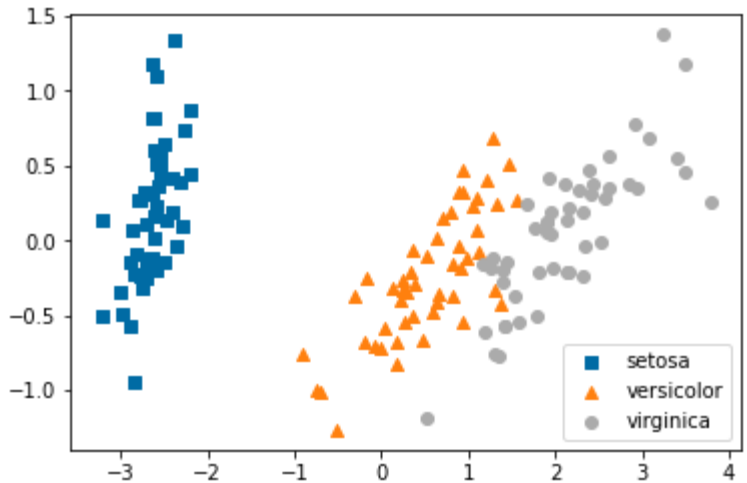
$$\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})^T$$

$$y \in \{\omega_1, \ldots, \omega_c\}$$

  - $\boldsymbol{x}$ は d 次元空間（特徴空間）上の点と見なせる
- 数値特徴に対する識別問題 ＝ 識別面の設定
  - 各クラスの確率分布を求めた結果として識別面（等確率点の集合）が定まる場合も含む

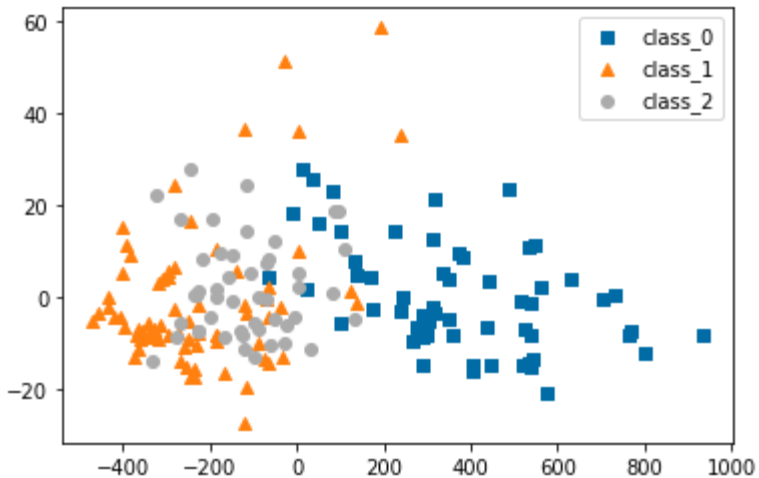
5.1 数値特徴に対する「教師あり・識別」問題の定義（2/5）

- irisデータ（4次元をPCAで2次元に削減後）の可視化



5.1 数値特徴に対する「教師あり・識別」問題の定義（3/5）

- wineデータ（13次元をPCAで2次元に削減後）の可視化

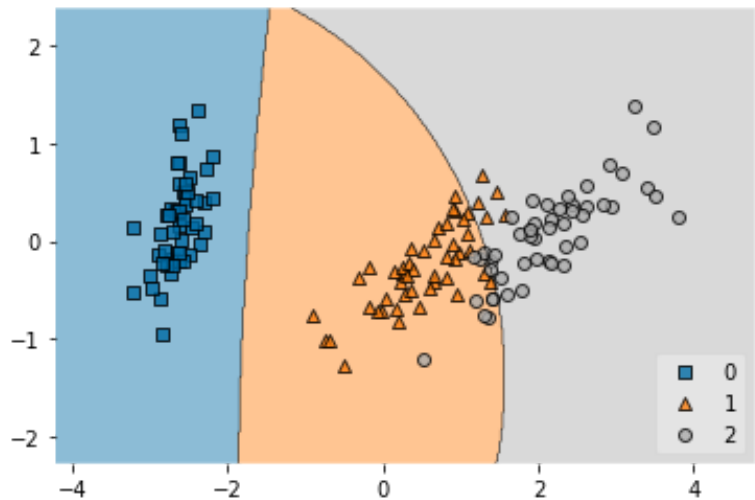


5.1 数値特徴に対する「教師あり・識別」問題の定義（4/5）

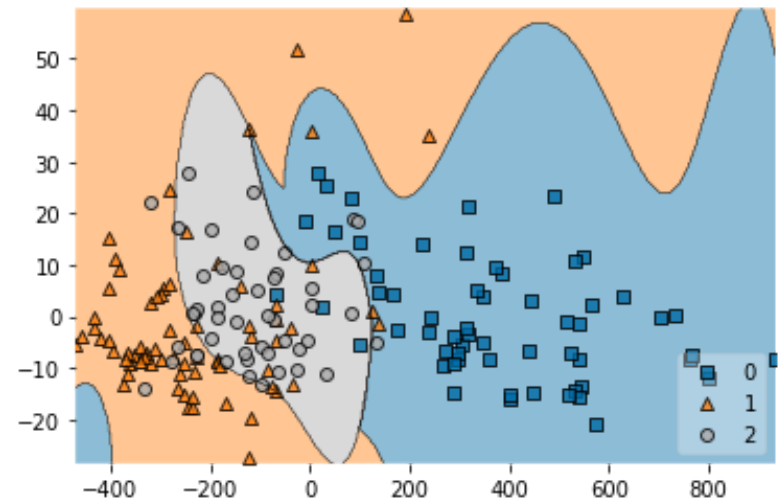
- 識別手法の選択基準
  - クラスが比較的きれいに分離している
    - 少数のパラメータで識別面を表現可能 ⇒ 統計的手法（第5章）
  - クラス境界が複雑
    - 高次元へマッピング ⇒ SVM（第7章）
    - 非線形識別 ⇒ ニューラルネット（第8章）

5.1 数値特徴に対する「教師あり・識別」問題の定義（5/5）

- 識別手法の適用例
  - 異なる識別手法で求めた識別面



irisデータに対してナイーブベイズで求めた識別面



wineデータに対してSVMで求めた識別面

5.2 生成モデル

- 統計的識別(4章)の復習
  - 最大事後確率則によるカテゴリカルデータの識別

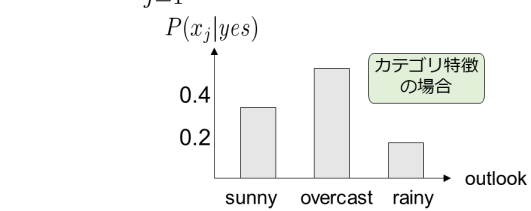
$$C_{MAP} = \arg\max_i P(\omega_i|\boldsymbol{x})$$
$$= \arg\max_i \frac{P(\boldsymbol{x}|\omega_i)P(\omega_i)}{P(\boldsymbol{x})}$$
$$= \arg\max_i P(\boldsymbol{x}|\omega_i)P(\omega_i)$$

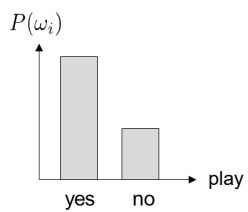
$\boldsymbol{x}$: 特徴ベクトル
$\omega_i \ (i = 1, ..., c)$: クラス

尤度　事前確率

$$P(\boldsymbol{x}|\omega_i) = P(x_1, \ldots, x_d|\omega_i)$$
$$\approx \prod_{j=1}^{d} P(x_j|\omega_i)$$

$$P(\omega_i) = \frac{n_i}{N}$$

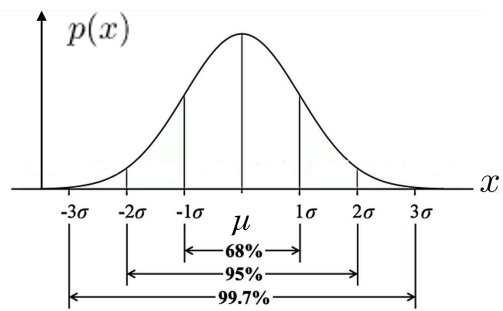




5.2.1 数値特徴に対するナイーブベイズ識別 (1/3)

- 各次元独立に確率密度関数 $p(x|\omega_i)$ を推定する
  - 関数はクラス毎に求めるので ω_i は省略
  - 関数の形は正規分布を仮定

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$



  - データの対数尤度を最大とする平均 μ と分散 σ^2 を求める

5.2.1 数値特徴に対するナイーブベイズ識別（2/3）

- データの対数尤度（最大化の対象）

$$\mathcal{L}(D) = \log p(D|\mu, \sigma^2) = \sum_{i=1}^{N} \log p(x_i|\mu, \sigma^2)$$

- $p(x)$ に正規分布の式を当てはめる

$$\mathcal{L}(D) = -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i - \mu)^2$$

5.2.1 数値特徴に対するナイーブベイズ識別（3/3）

- $\mathcal{L}(D)$ を μ で偏微分して0とおき、μ について解く

$$\frac{1}{\sigma^2}\sum_{i=1}^{N}(x_i-\mu)=0 \Rightarrow \mu=\frac{1}{N}\sum_{i=1}^{N}x_i$$

- $\mathcal{L}(D)$ を σ^2 で偏微分して0とおき、σ^2 について解く

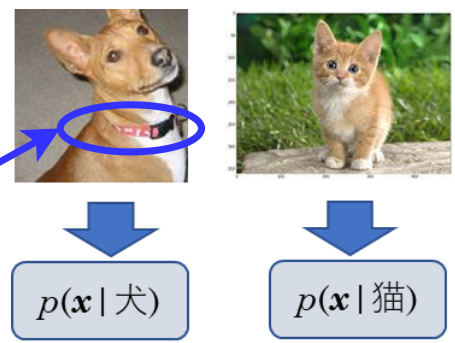
$$-\frac{N}{2}\frac{1}{\sigma^2}+\frac{1}{2(\sigma^2)^2}\sum_{i=1}^{N}(x_i-\mu)^2=0 \Rightarrow \sigma^2=\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2$$

- 求める分布の平均はデータの平均、分散はデータの分散というごく当たり前の結果

5.2.2 生成モデルの考え方

- データが生成される様子をモデル化しているとみなせる
  - 事前確率に基づいてクラスを選ぶ
  - そのクラスの尤度関数を用いて特徴ベクトルを出力する
- 事後確率を求めるより難しい問題を解いているのではないか？

$$\arg\max_i P(\omega_i|\boldsymbol{x}) = \arg\max_i P(\boldsymbol{x}|\omega_i)P(\omega_i)$$

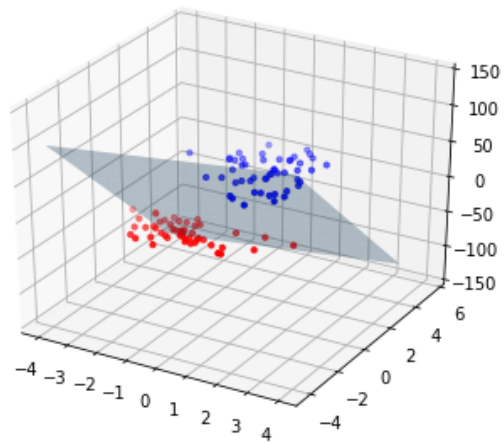


これが発見できればよいのでは？
→ **識別モデル**の考え方

生成モデルで
やっていること

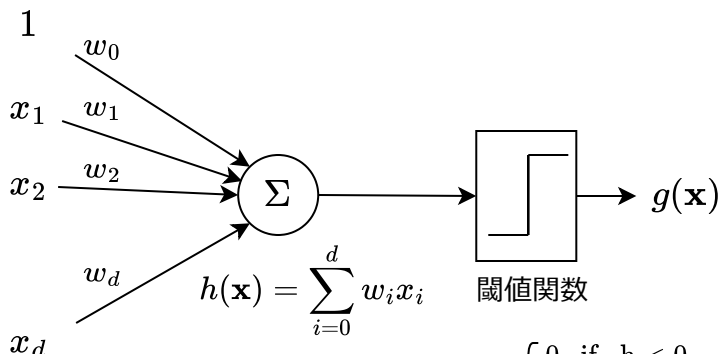
5.3 識別モデル

- 識別関数法
  - 確率の枠組みにはとらわれず、$g_i(\boldsymbol{x}) > g_j(\boldsymbol{x}), \quad i \neq j$ ならば $\boldsymbol{x}$ をクラス ω_i と判定する関数を推定する
  - 2クラス問題なら $g(\boldsymbol{x}) = g_{positive}(\boldsymbol{x}) - g_{negative}(\boldsymbol{x})$ の正負で判定する
    - $g(\boldsymbol{x}) = 0$ が識別面



5.3.1 誤り訂正学習 (1/2)

- 単層パーセプトロンの定義
  - 識別関数として1次式(＝直線・平面)を仮定
  - $\boldsymbol{w}^T\boldsymbol{x} = 0$ という特徴空間上の超平面を表現
    - 以後 $\boldsymbol{w}$ は w_0を含み、$\boldsymbol{x}$ は $x_0 \equiv 1$ と定義した $d+1$ 次元ベクトルとする

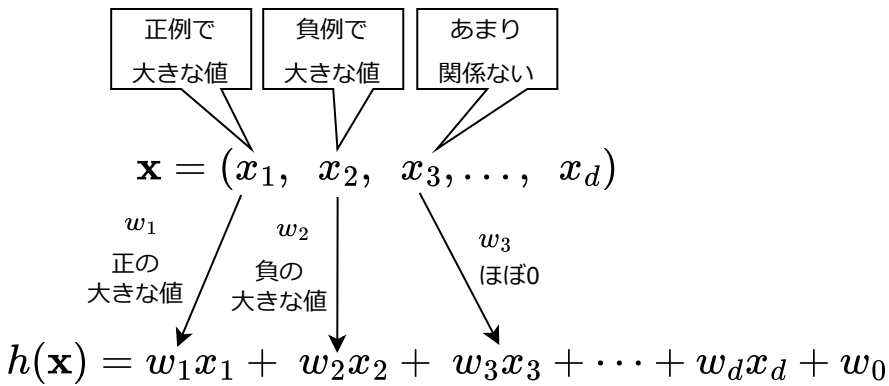


5.3.1 誤り訂正学習 (2/2)

- パーセプトロンの学習規則
  1. 重み $\boldsymbol{w}$ の初期値を適当に決める
  2. 学習データからひとつ $\boldsymbol{x}$ を選び、$g(\boldsymbol{x})$ を計算
  3. 誤識別($y \neq g(\boldsymbol{x})$)のときのみ $\boldsymbol{w}$ を修正する(ρ: 学習係数)
     - $\boldsymbol{w}' = \boldsymbol{w} + \rho\boldsymbol{x}$ (positive のデータを negative と誤ったとき)
     - $\boldsymbol{w}' = \boldsymbol{w} - \rho\boldsymbol{x}$ (negative のデータを positive と誤ったとき)
  4. 2,3 をすべての学習データについて繰り返す
  5. すべて正しく識別できたら終了。そうでなければ2へ
- パーセプトロンの学習規則の適用範囲
  - データが線形分離可能な場合は重みの学習が可能
  - 線形分離不可能な場合は学習が終了しない

5.3.3 識別モデルの考え方

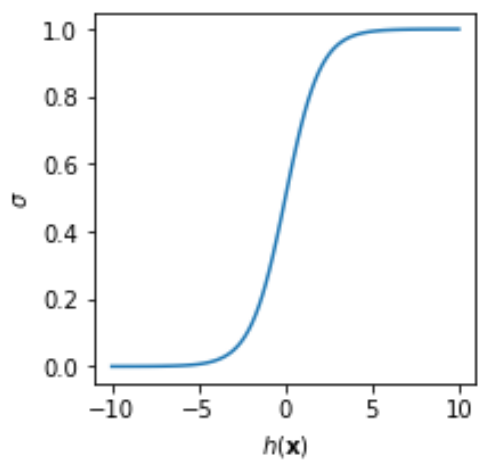
- 識別に役に立つ特徴に着目して識別関数の重みを調整
  - 正例に偏って正の値をとる特徴は、正の大きな重み
  - 負例に偏って正の値をとる特徴は、負の大きな重み
  - 特徴の値が逆転するときは重みの値も逆転
  - 識別にあまり関与しない特徴は、0に近い重み



5.3.4 ロジスティック回帰 (1/4)

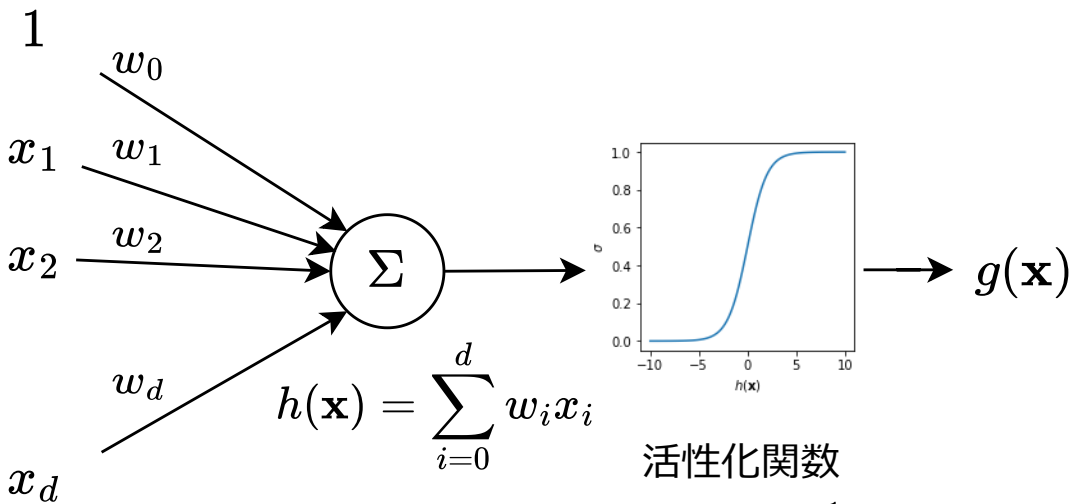
- ロジスティック回帰の考え方
  - $h(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$ の値をシグモイド関数σで$(0, 1)$の値に変換
  - 出力を正例の確率とみなす

$$g(\boldsymbol{x}) = \sigma(h(\boldsymbol{x})) = \frac{1}{1 + \exp(-h(\boldsymbol{x}))}$$



5.3.4 ロジスティック回帰（2/4）

- ロジスティク回帰の計算ユニット



活性化関数

$$\sigma(h(\mathbf{x})) = \frac{1}{1 + \exp(-h(\mathbf{x}))}$$

5.3.4 ロジスティック回帰（3/4）

- ロジスティク回帰の学習
  - 最適化対象：負の対数尤度を損失とみなして最小化する

$$E(\boldsymbol{w}) = -\log P(D|\boldsymbol{w}) = -\log \prod_{i=1}^{N} o_i^{y_i}(1-o_i)^{(1-y_i)} = -\sum_{i=1}^{N}\{y_i \log o_i + (1-y_i)\log(1-o_i)\}$$

$$o_i = g(\boldsymbol{x}_i) = \frac{1}{1+\exp(-\boldsymbol{w}^T\boldsymbol{x}_i)}$$

$$y_i \in \{0, 1\}$$

5.3.4 ロジスティック回帰 (4/4)

- $E(\boldsymbol{w})$ を勾配降下法で最小化
  - 適当な初期値 $\boldsymbol{w}$ から始め、$E(\boldsymbol{w})$ の勾配の逆方向に少しずつ修正

$$w_j \leftarrow w_j - \eta \frac{\partial E(\boldsymbol{w})}{\partial w_j}$$

  - 重みの更新量の計算

$$\frac{\partial E(\boldsymbol{w})}{\partial w_j} = \sum_{i=1}^{N} \frac{\partial E(\boldsymbol{w})}{\partial o_i} \cdot \frac{\partial o_i}{\partial w_j} = \sum_{i=1}^{N} (\frac{y_i}{o_i} - \frac{1-y_i}{1-o_i}) o_i (1-o_i) x_{ij} = \sum_{i=1}^{N} (y_i - o_i) x_{ij}$$

  - 重みの更新式

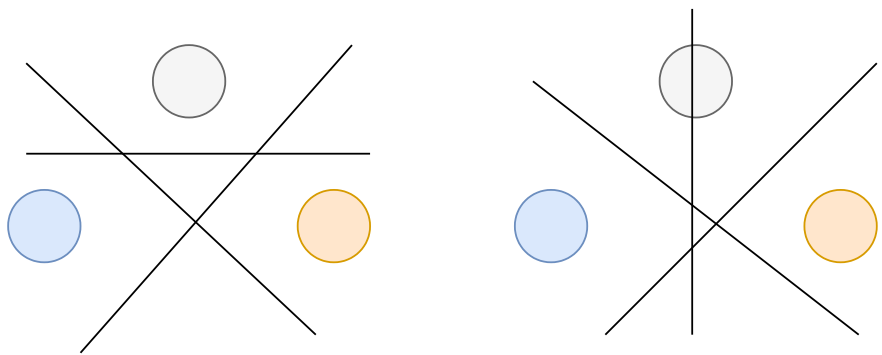
$$w_j \leftarrow w_j - \eta \sum_{i=1}^{N} (y_i - o_i) x_{ij}$$

5.3.5 確率的勾配降下法

- 勾配降下法の問題点
  - 全データに対して損失を計算するので、データが多いと重み更新に時間がかかる
- 確率的勾配降下法
  - ランダムに一つのデータを選択し、その損失に基づき重みを更新する
  - 更新方向が安定しないが、十分な回数繰り返せば最適解に至る
  - データが来る毎に学習するオンライン学習として適用可能
- ミニバッチ法
  - 数十～数百程度のデータで損失を計算し、修正方向を決める
  - 更新方向が比較的安定し、最適解への収束も早い
  - GPUを用いて高速な学習が可能
- 学習のシミュレーションサイト https://playground.tensorflow.org/

多クラスの識別 (1/2)

- 2クラス分類器を用いた多クラス分類
  - one-versus-rest法
    - 各クラスについて、そのクラスに属するかどうかを識別する識別器を作る
    - 2つ以上のクラスに属すると判定された場合は識別面からの距離が大きいものに分類する
  - ペアワイズ法
    - クラス対ごとに識別器を作り、判定は多数決を取る



one-versus-rest法　　ペアワイズ法

多クラスの識別（2/2）

- ロジスティク回帰の活性化関数をsoftmaxとする

$$g_j(\boldsymbol{x}) = \frac{\exp(\boldsymbol{w}_j^T \boldsymbol{x})}{\sum_{k=1}^{c} \exp(\boldsymbol{w}_k^T \boldsymbol{x})}$$

  - 最適化対象

$$E(\boldsymbol{w}) = -\log P(D|\boldsymbol{w}) = -\sum_{i=1}^{N}\sum_{j=1}^{c}[y_i == j] \log g_j(\boldsymbol{x}_i)$$

    - []（Iverson bracket）：内部の命題が真ならば1、偽ならば0を返す

5.4 まとめ

- 数値特徴の「教師あり・識別」問題へのアプローチ
- 生成モデル
  - 学習データを各クラスに分割
  - それぞれのクラスの尤度関数を最尤推定
  - 別途、事前確率が得られているような場合に有効
- 識別モデル
  - 損失関数を定義し、勾配降下法でパラメータを学習
  - クラス分割に寄与する特徴を見つけていると解釈できる
  - 一般的に生成モデルよりも性能が高い