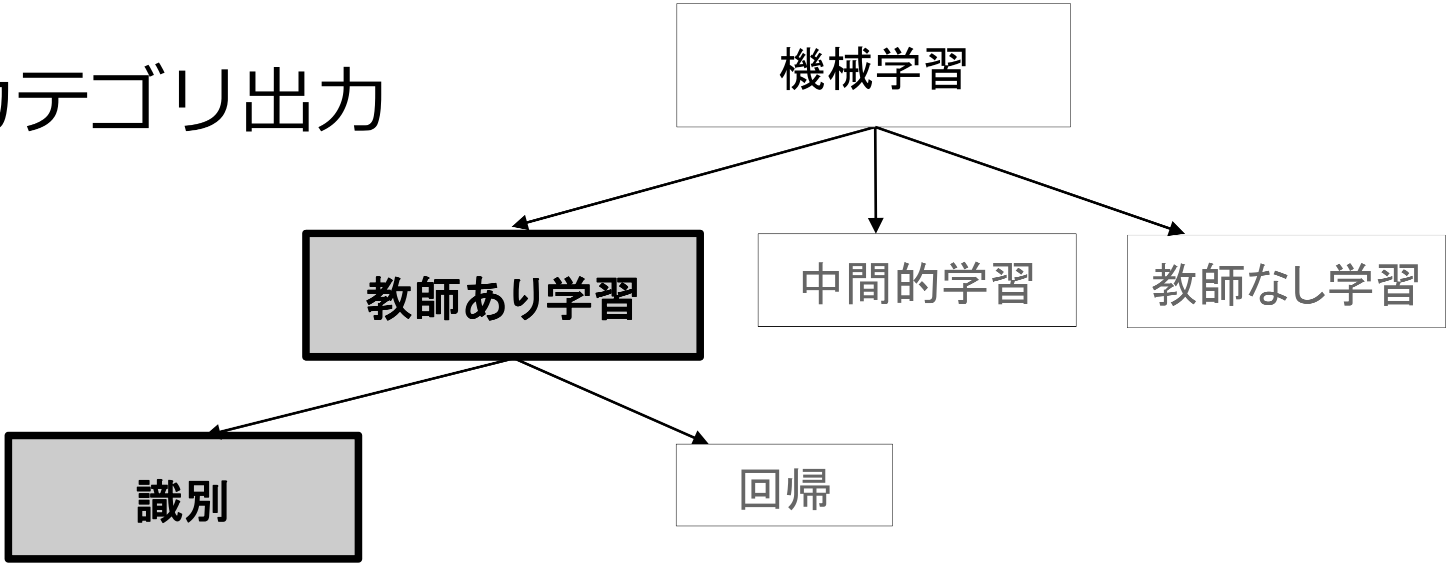
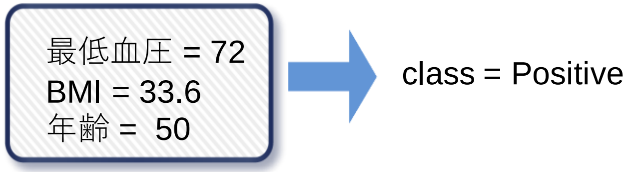
5. 識別　ー生成モデルと識別モデルー

- 問題設定
  - ◆ 教師あり学習
  - ◆ 数値入力　→　カテゴリ出力



- 数値特徴



5.1 数値特徴に対する「教師あり・識別」問題の定義

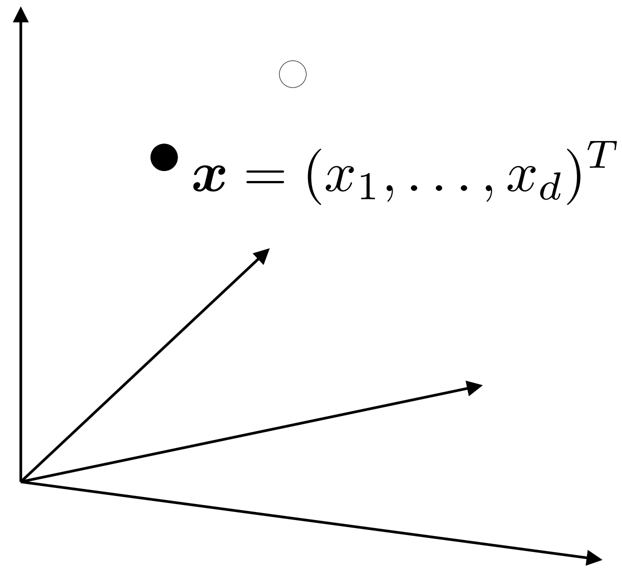
- 識別問題のデータ
  - 特徴ベクトル $\boldsymbol{x}$ と正解情報 y のペア

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

  - $\boldsymbol{x}$ は次元数 d の固定長ベクトル、y はカテゴリ

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

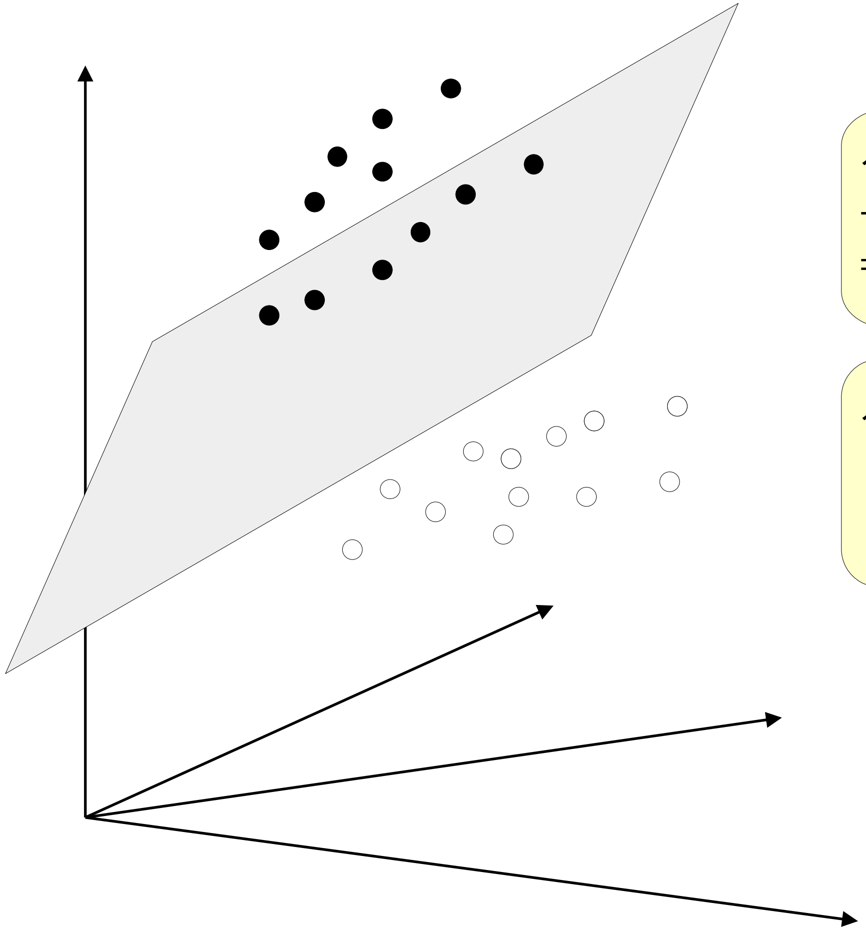
  - $\boldsymbol{x}$は d 次元空間（特徴空間）上の点と見なせる



● y = positive（正例）
○ y = negative（負例）

5.1 数値特徴に対する「教師あり・識別」問題の定義

- 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ることで、結果として識別面（等確率となる点の集合）が定まる場合も含む



クラスが比較的きれいに分離している
→ 小数のパラメータで各クラスまたは境界面を表現可能
⇒ **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）**
　非線形識別面 ⇒ **ニューラルネット（第8章）**

統計的識別の復習

- 最大事後確率則による識別

$\boldsymbol{x}$ ：特徴ベクトル

$\omega_i \quad (1 \leq i \leq c)$ ：クラス

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

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

ベイズの定理

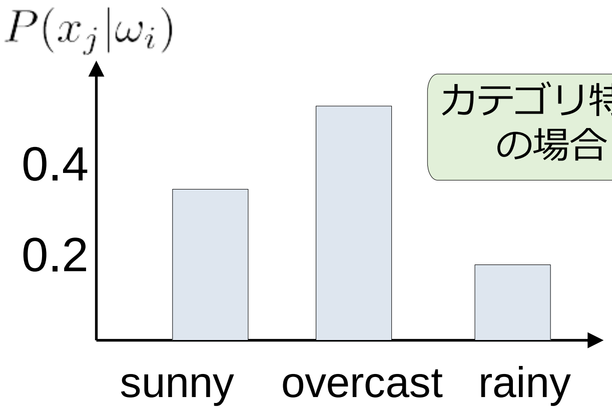
$$= \arg\max_i \underbrace{P(\boldsymbol{x}|\omega_i)}_{\text{尤度}}\underbrace{P(\omega_i)}_{\text{事前確率}}$$

上式の分母は
判定に寄与しない

ナイーブベイズの仮定

$$P(\boldsymbol{x}|\omega_i) = P(x_1, \ldots, x_d|\omega_i)$$

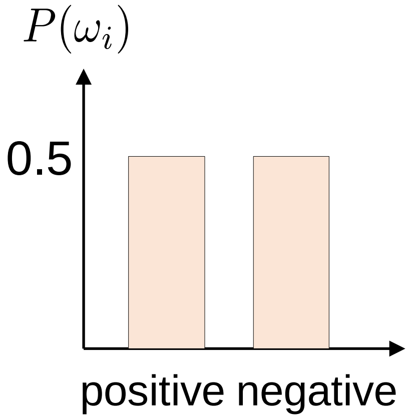
$$\approx \prod_{j=1}^{d} P(x_j|\omega_i)$$



カテゴリ特徴
の場合

学習データのクラス
分布から最尤推定

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



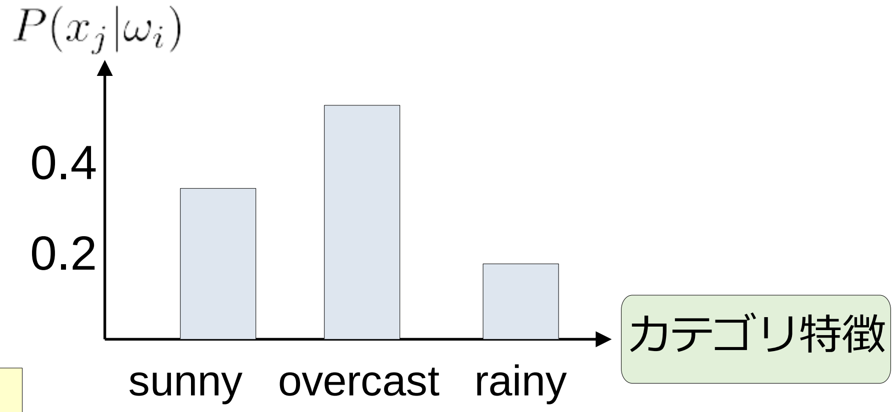
5.2 生成モデル
5.2.1 数値特徴に対するナイーブベイズ識別

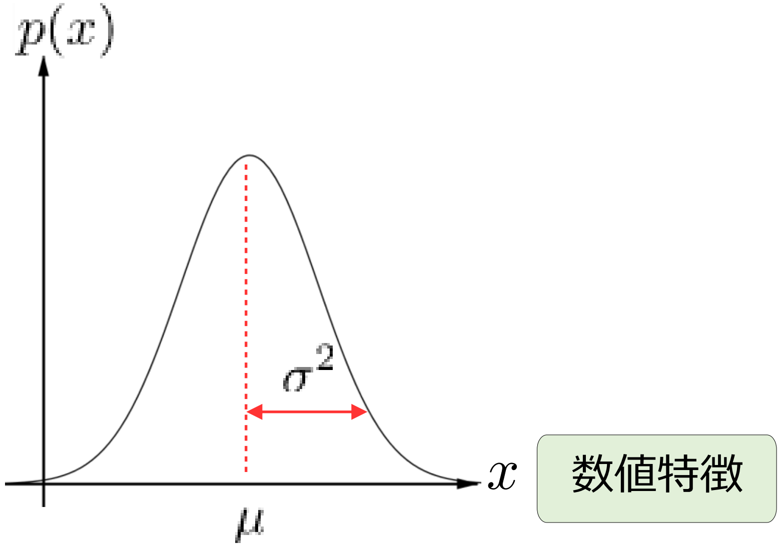
- 確率密度関数 $p(x_j \mid \omega_i)$ の推定
  - 関数をクラス毎に求めるので ω_i は省略
  - 関数の形は正規分布を仮定

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

$f(x) = e^{-x^2}$ の変形

  - 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

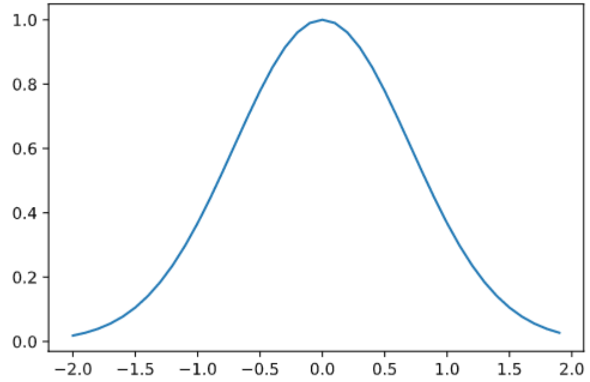




正規分布とは

- 離散型二項分布の例
  - ◆ n枚のコインを投げた時の、表の枚数の度数

n=1　1 1
n=2　1 2 1
n=3　1 3 3 1
n=4　1 4 6 4 1
n=5　1 5 10 10 5 1
…



  - ◆ $n \to \infty$ のときが正規分布
    - 独立な要因の合計によって生じた数値がこの分布にあてはまる

      例）n問からなるテストの点数

5.2.1 数値特徴に対するナイーブベイズ識別

- データの対数尤度（最大化したい）

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

- p に正規分布の式を当てはめる

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

- μ で偏微分して0とおく

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

- σ^2 で偏微分して0とおく

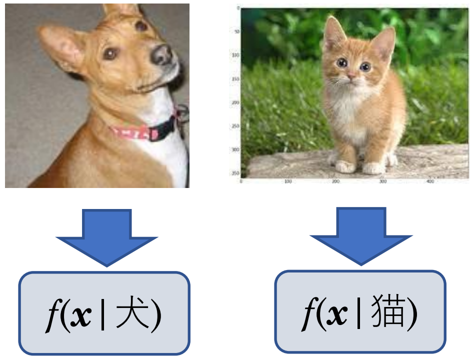
$$-\frac{N}{2}\frac{1}{\sigma^2} + \frac{1}{2(\sigma^2)^2}\sum_{i=1}^{N}(x_i - \mu)^2 = 0 \Rightarrow \sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$$

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
  - ◆ クラス毎に別々に学習された尤度関数を用いて特徴ベクトルを出力する

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



これが発見できればよいのでは？

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？

5.3 識別モデル

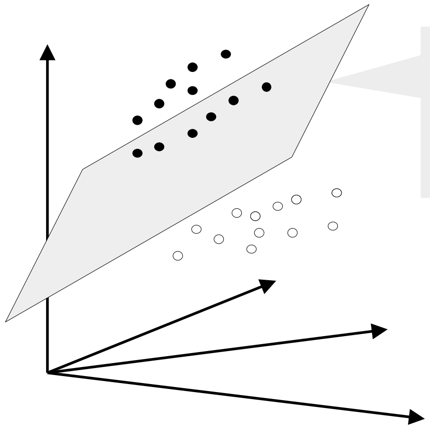
- 識別関数法
  - ◆ 確率の枠組みにはとらわれず、 $f_P(\boldsymbol{x}) > f_N(\boldsymbol{x})$ ならば $\boldsymbol{x}$ を positiveと判定する関数を推定する
  - ◆ 2クラス問題なら $f(\boldsymbol{x}) = f_P(\boldsymbol{x}) - f_N(\boldsymbol{x})$ の正負で判定すればよい（ $f(\boldsymbol{x}) = 0$ が識別面）
  - ◆ 単層パーセプトロン
    - 識別関数として1次式（＝直線・平面）を仮定

最も単純な識別関数法の実現

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$$

$$= \sum_{i=0}^{d} w_i x_i$$

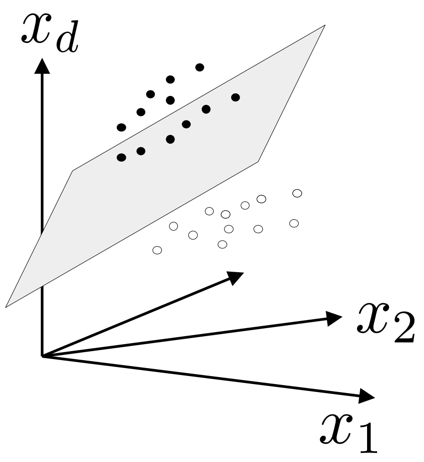
$\boldsymbol{x}$ を d+1 次元に拡張し、$x_0 \equiv 1$ とする

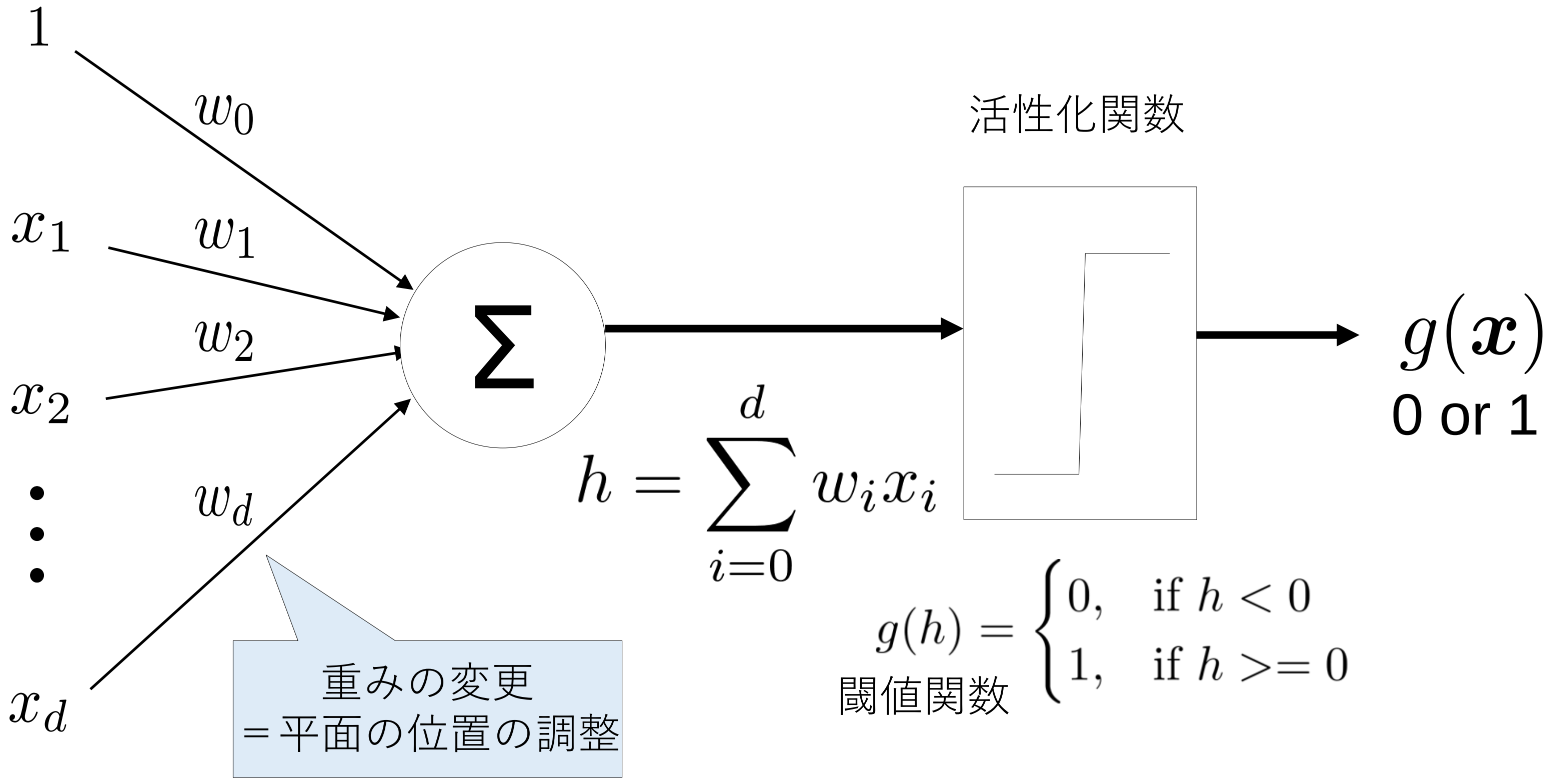


5.3.1 誤り訂正学習

- 単層パーセプトロンの定義
  - ◆ $\boldsymbol{w}^T\boldsymbol{x} = 0$ という特徴空間上の超平面を表現

以後、$\boldsymbol{w}$はw_0 を含む





$$h = \sum_{i=0}^{d} w_i x_i$$

$$g(h) = \begin{cases} 0, & \text{if } h < 0 \\ 1, & \text{if } h >= 0 \end{cases}$$

閾値関数

5.3.1 誤り訂正学習

- 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

$$\boldsymbol{w}' = \boldsymbol{w} + \rho \boldsymbol{x}$$ （positiveのデータをnegativeと誤ったとき）

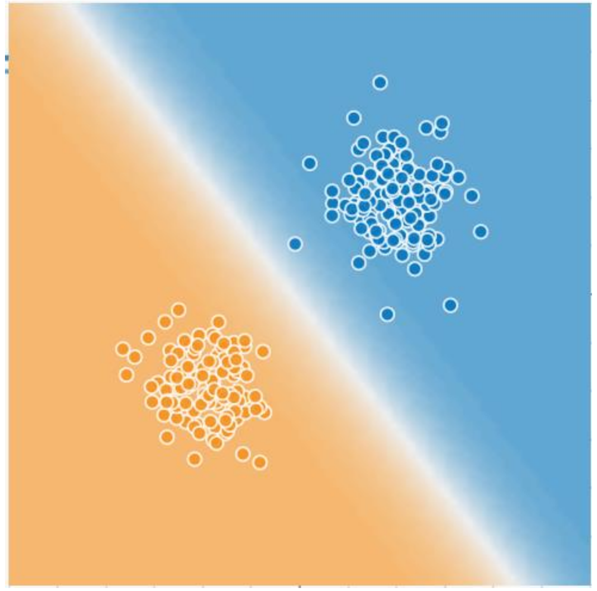
$$\boldsymbol{w}' = \boldsymbol{w} - \rho \boldsymbol{x}$$ （negativeのデータをpositiveと誤ったとき）

ρ：学習係数

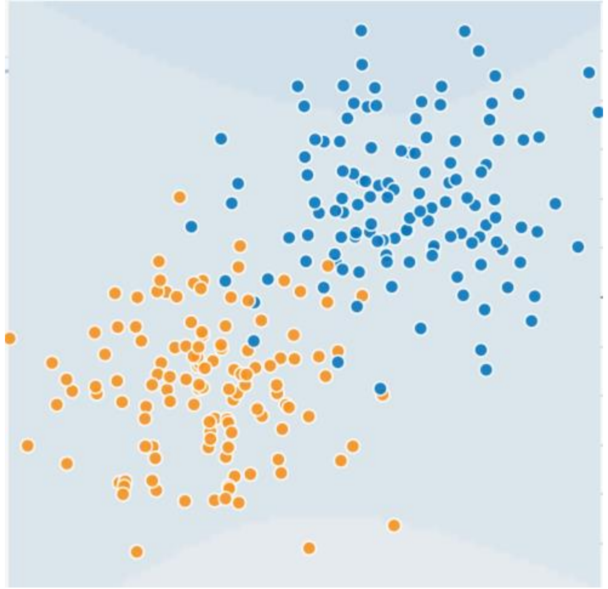
  4. 2,3 をすべての学習データについて繰り返す
  5. すべて正しく識別できたら終了。そうでなければ2へ

5.3.1 誤り訂正学習

- 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



線型分離可能なデータ



線型分離不可能なデータ

5.3.3 識別モデルの考え方

- 識別関数法を事後確率の推定に適用

正例で大きな値　負例で大きな値　あまり関係ない

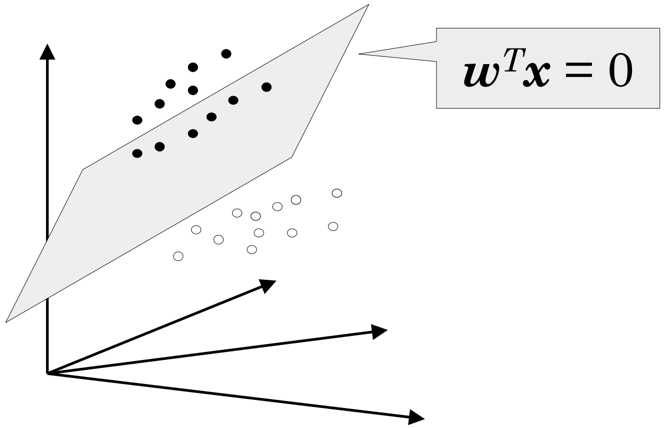
$$\boldsymbol{x} = (x_1, \quad x_2, \quad x_3, \quad \ldots, \quad x_d)$$

w_1 正の大きな値　w_2 負の大きな値　w_3 小さな値

正例／負例の閾値

$$\boldsymbol{w}^T\boldsymbol{x} = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_dx_d + w_0$$

この値が正なら正例、
負なら負例となるように
重み$\boldsymbol{w}$を学習する

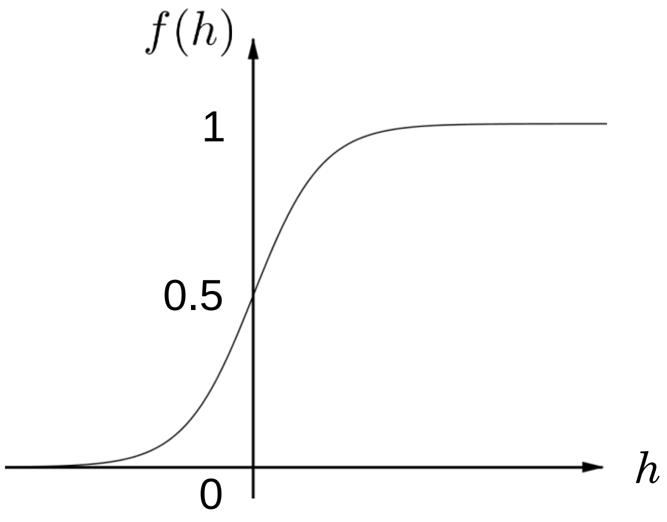


確率と対応づけるには？

5.3.4 ロジスティック識別

- ロジスティック識別
  - ◆ $\boldsymbol{w}^T\boldsymbol{x}$ の値を確率に変換（2クラスの場合は正例の確率とみなす）

$$P(Positive \mid \boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{w}^T\boldsymbol{x})}$$



シグモイド関数　$f(h) = \dfrac{1}{1+e^{-h}}$

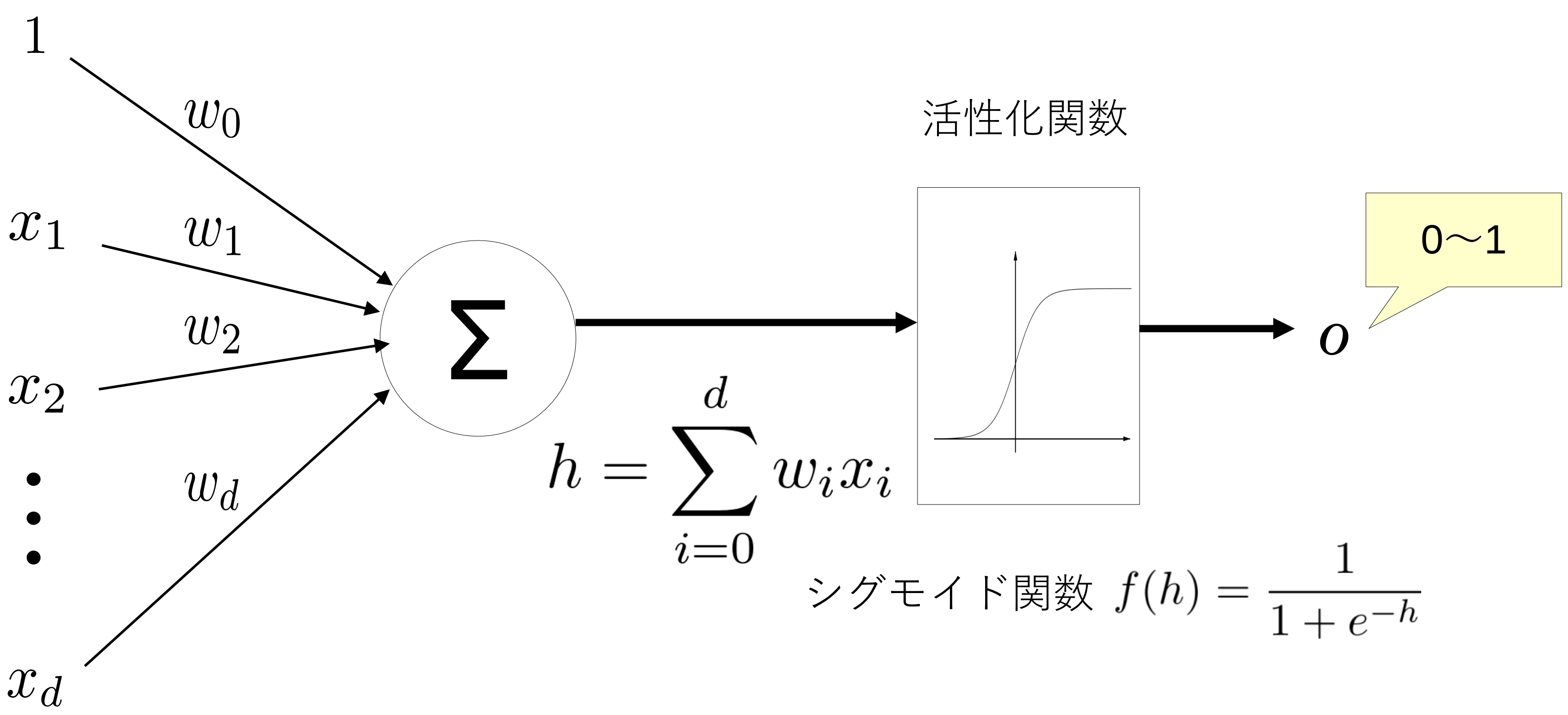
シグモイド関数の利点

- -∞～+∞の値域を持つものを、順序を変えずに0～1にマッピングできる
- 微分形が簡単な式になる

$$f'(h) = f(h)(1-f(h))$$

5.3.4 ロジスティック識別

- ロジスティック識別の計算ユニット



5.3.4 ロジスティック識別

- 最適化対象：モデルの対数尤度の反数（最小化）

正解ラベル（0 or 1）

損失関数

$$
\begin{aligned}
E(\boldsymbol{w}) &= -\log P(D|\boldsymbol{w}) \\
&= -\log \prod_{\boldsymbol{x}_i \in D} o_i^{y_i}(1-o_i)^{(1-y_i)} \\
&= -\sum_{\boldsymbol{x}_i \in D} \{y_i \log o_i + (1-y_i)\log(1-o_i)\}
\end{aligned}
$$

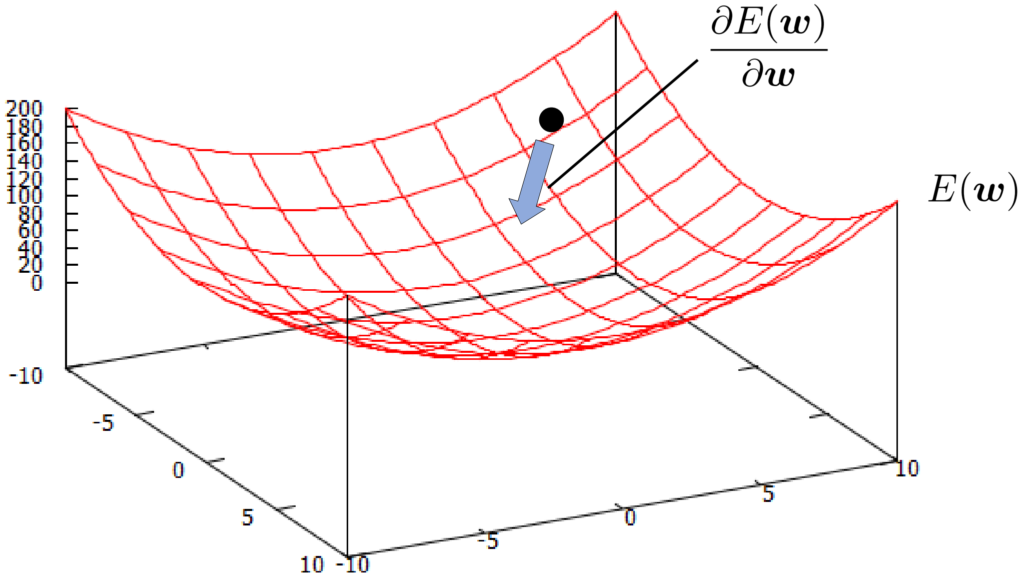
$$\boxed{o_i = P(Positive \mid \boldsymbol{x}_i) = sigmoid(\boldsymbol{w}^T \boldsymbol{x}_i)}$$

5.3.4 ロジスティック識別

- $E(\boldsymbol{w})$を最急勾配法で最小化
  1. 適当な初期値 $\boldsymbol{w}$ を選ぶ
  2. $\boldsymbol{w}$ を $E(\boldsymbol{w})$ の勾配の逆方向に少しずつ修正

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

η : 学習係数



5.3.4 ロジスティック識別

- 重みの更新量の計算

$$\frac{\partial E(\boldsymbol{w})}{\partial w_j} = \frac{\partial E(\boldsymbol{w})}{\partial o_i} \cdot \frac{\partial o_i}{\partial w_j}$$

$$= -\sum_{\boldsymbol{x}_i \in D} \left(\frac{y_i}{o_i} - \frac{1-y_i}{1-o_i}\right) o_i(1-o_i) x_{ij}$$

$$= -\sum_{\boldsymbol{x}_i \in D} (y_i - o_i) x_{ij}$$

$\boldsymbol{E(w)}$ を$\boldsymbol{o}_i$で偏微分

$o_i = P(Positive \mid \boldsymbol{x}_i) = sigmoid(\boldsymbol{w}^T \boldsymbol{x}_i)$

シグモイド関数の微分

w_jの係数

- 重みの更新式

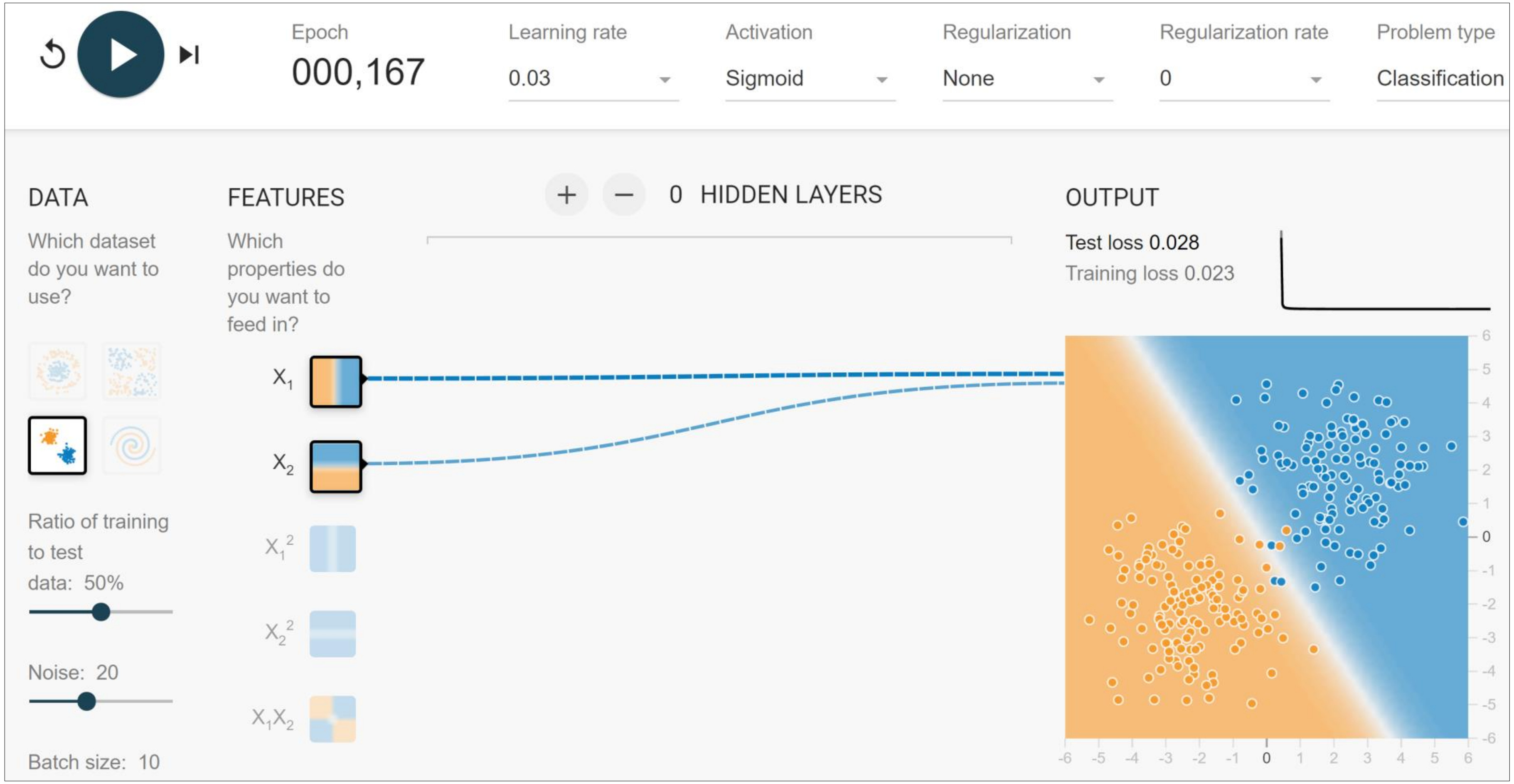
$$w_j \leftarrow w_j + \eta \sum_{\boldsymbol{x}_i \in D} (y_i - o_i) x_{ij}$$

5.3.5 確率的最急勾配法

- 最急勾配法の問題点
  - ◆ 全データに対して損失を計算するので、データ数が多い場合、重み更新に時間がかかる
- 確率的最急勾配法
  - ◆ 個々のデータに対する損失に基づき重みを更新
  - ◆ データが来る毎に学習するオンライン学習が可能
- 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

学習のシミュレーションサイト

- https://playground.tensorflow.org/



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独立に推定
      - ✓ 尤度関数：未知の特徴ベクトルを入力としてクラスらしさを出力
  - ◆ 識別モデル
    - クラス分割に寄与する特徴に重みをかける
    - シグモイド関数で確率値に変換：ロジスティック識別
    - 損失関数を定義し、最急勾配法でパラメータを学習