データサイエンス
第８回

~クラス分類(2)~

情報理工学系研究科
創造情報学専攻
中山 英樹

レポートについてお願い

- ランキングと対応付けて欲しい人はユーザ名をレポート中に書いてください
  - ちゃんと評価できるようにするため
  - 上位の人にコンタクトできるとうれしい

  - もちろん強制ではないです

本日の内容

- クラス分類の続き

- 各手法の外観・位置づけ
  - 生成的アプローチ
  - 識別的アプローチ

クラス分類（クラス識別）

- クラスタリングとは全く違うので注意
  - <u>データから何かを発見したい</u> → 教師なし学習

説明変数	手法
量的データ（比尺度）	主成分分析、因子分析、LPP
量的データ（間隔尺度）	クラスター分析、多次元尺度構成法、数量化Ⅳ類
質的データ	数量化Ⅲ類、対応分析

  - <u>データを使って何かを予測したい</u> → 教師あり学習

目的変数	説明変数	手法
量的データ	量的データ	回帰分析
	質的データ	数量化Ⅰ類
質的データ	量的データ	判別分析、SVM、kNN…
	質的データ	数量化Ⅱ類

識別規則（の例）

- パターンの特徴ベクトル x が与えられた時のクラスCの事後確率が重要

$$P(C \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid C)P(C)}{P(\mathbf{x})}$$

- 事後確率を最大とするクラスへ識別
  - 誤識別率を最小にする
  - 誤識別のリスク（ペナルティ）がクラスによらず一定の時、最適な識別境界を与える（ベイズ識別と一致）

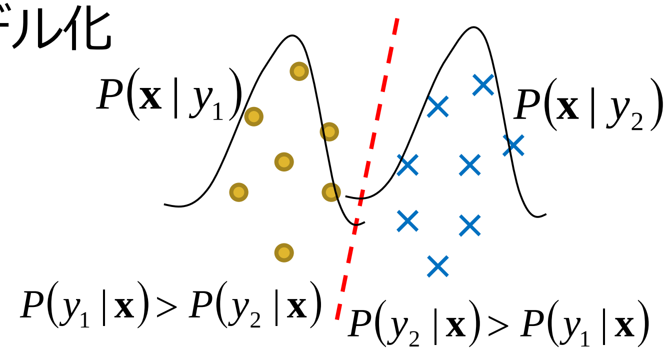
$$\boxed{\hat{C} = \arg\max_{C} P(C \mid \mathbf{x})}$$

分類のアプローチ（事後確率の推定方法）

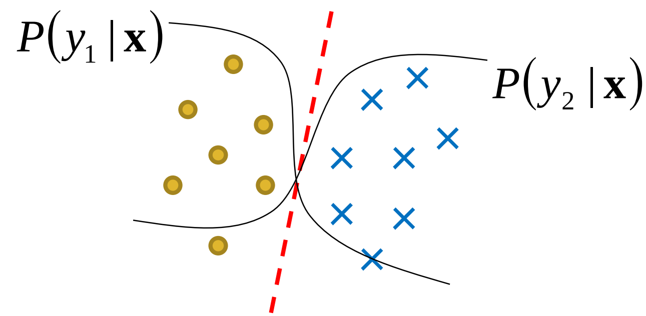
より一般的

- 1．生成モデル
  - クラスごと条件付き確率と事前確率をモデル化
  - ナイーブベイズ
  - k-最近傍法

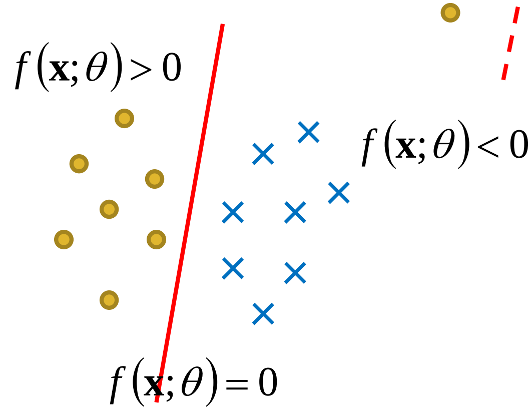
$$P(y \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid y) P(y)}{P(\mathbf{x})}$$



- 2．識別モデル
  - 事後確率を $P(y \mid \mathbf{x})$ 直接的にモデル化（元の分布はどうでもよい）
  - ロジスティック回帰



- 3．識別関数
  - 識別の境界面だけモデル化
  - SVM



より識別に特化

生成的アプローチ

- 事後確率分布だけでなく、同時確率分布までモデル化

$$\underbrace{P(C \mid \mathbf{x})}_{\text{事後確率}} \propto \underbrace{P(\mathbf{x} \mid C)}_{\text{条件付き確率}} \underbrace{P(C)}_{\text{事前確率}}$$

  - クラスの事前確率は、何らかの先見知識がある場合を除き、単純にサンプルの割合で推定することが多い $\hat{P}(C) = \frac{N_C}{N}$

- 条件付き確率（生成モデル）の推定がポイント
例）正規分布を用い、最尤推定する（パラメトリック）

$$\hat{P}(\mathbf{x} \mid C) = N\left(\mathbf{x}; \hat{\mu}_C, \hat{\Sigma}_C\right)$$

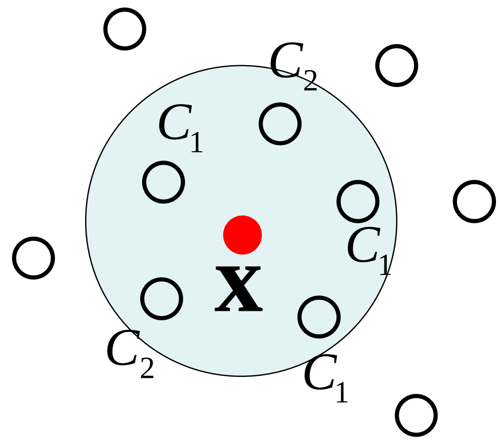
$\hat{\mu}_C, \hat{\Sigma}_C$ は最尤推定量、すなわちクラスCのサンプルの平均と分散

パラメトリックなモデルによる生成的分類

- 正規分布によるモデル化の場合
  - 各クラスの共分散パラメータを一定と仮定した場合、線形判別分析と同じ識別境界が得られる
- より複雑なモデル（GMMなど）は実際は扱いが難しい
  - パラメータが多い
  - 計算コストが膨大
  - 学習サンプルも大量に必要
  - 推定自体困難
  - …

K-最近傍法 (K-nearest neighbor, K-NN)

- 識別則は非常にシンプル
  - パターン入力 x について、 最も近い上位K個の学習データの中で、最も多い数のデータが所属するクラスへ x を識別



$k = 5$ の例

- 直感的には
  - 入力に類似している学習データの多数決
  - データが増えると精度は上がるが、識別のコストは非常に大きくなる

K-NN:確率的な解釈

- クラス C_k に属する学習データ数を N_k,
  全学習データ数を $N = \sum N_k$ とする
- 入力 x を中心とし、x の最近傍K点を含む超球の体積をVとする
- 超球に含まれる C_k のサンプル数を K_k とする

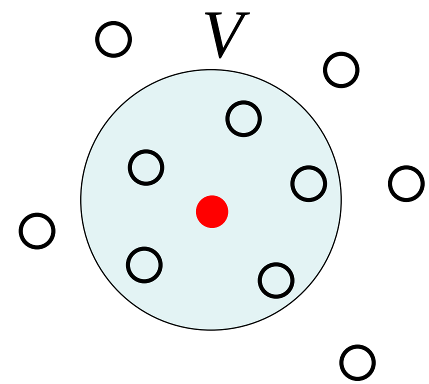
超球の内部（局所領域）
については以下のように近似できる

$$P(\mathbf{x} \mid C_k) = \frac{K_k}{N_k V}$$

$$P(\mathbf{x}) = \frac{K}{NV}$$

確率密度分布が局所的に
一定と近似

$$P(C_k) = \frac{N_k}{N}$$



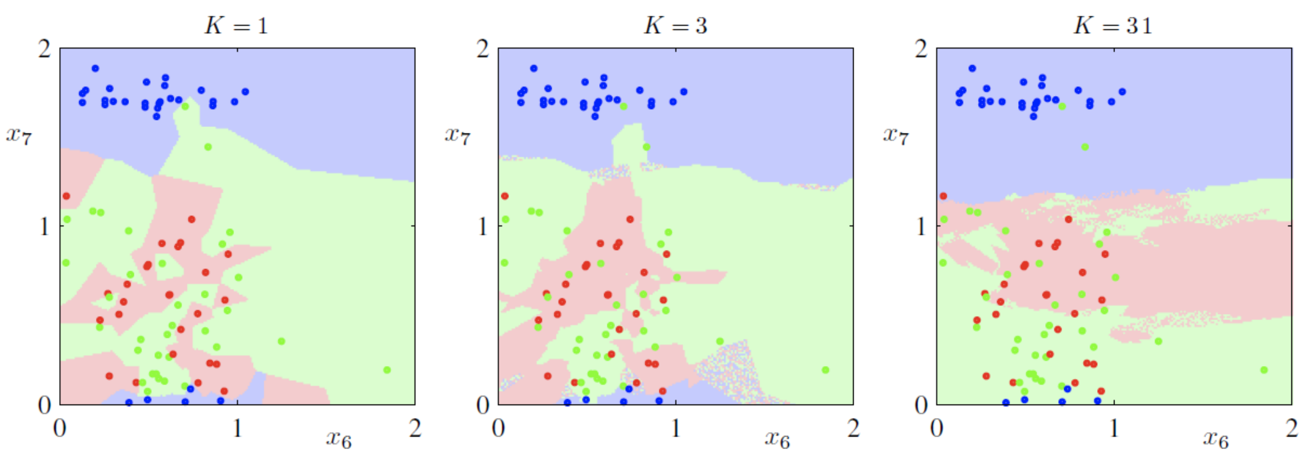
$$\therefore P(C_k \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid C_k) P(C_k)}{P(\mathbf{x})} \cong \frac{K_k}{K}$$

多クラス識別も自然に実現できる

K-NN: 確率的な解釈

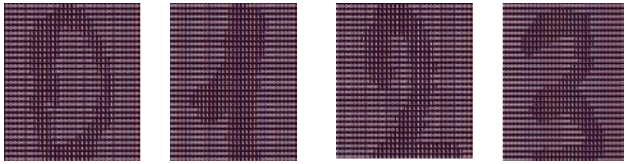
- 背後では、生成モデルの推定を行っていると解釈できる
  - 生成モデルのパラメータは置かないので、ノンパラメトリックな手法と呼ばれる
  - カーネル密度推定法と関連が深い

- Kは生成モデルの滑らかさを決定するハイパーパラメータ
  - モデルそのもののパラメータではないことがポイント
  - K→大：より大域的・単純な分布
  - K→小：より局所的・複雑な分布



手書き文字認識 (notebook)

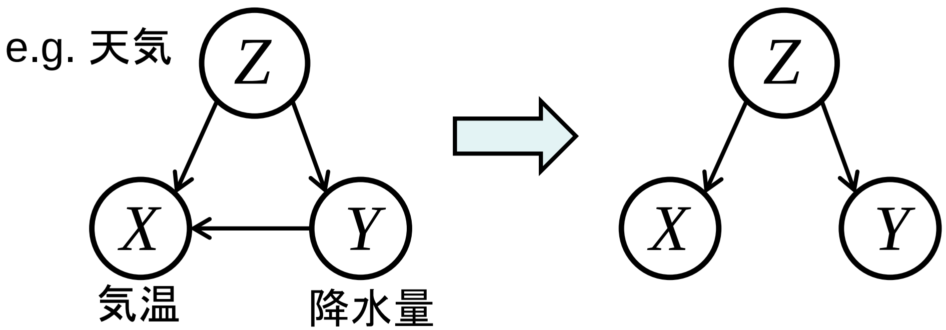
- 0~9の手書きの数字をKNNで認識してみる
  - 32x32サイズのバイナリ画像



ナイーブベイズ（単純ベイズ）

- もともと識別手法の名前ではない（非常に一般的かつ重要な概念）
  - 条件付きの同時確率を個々の条件付き確率の積へばらす近似方法

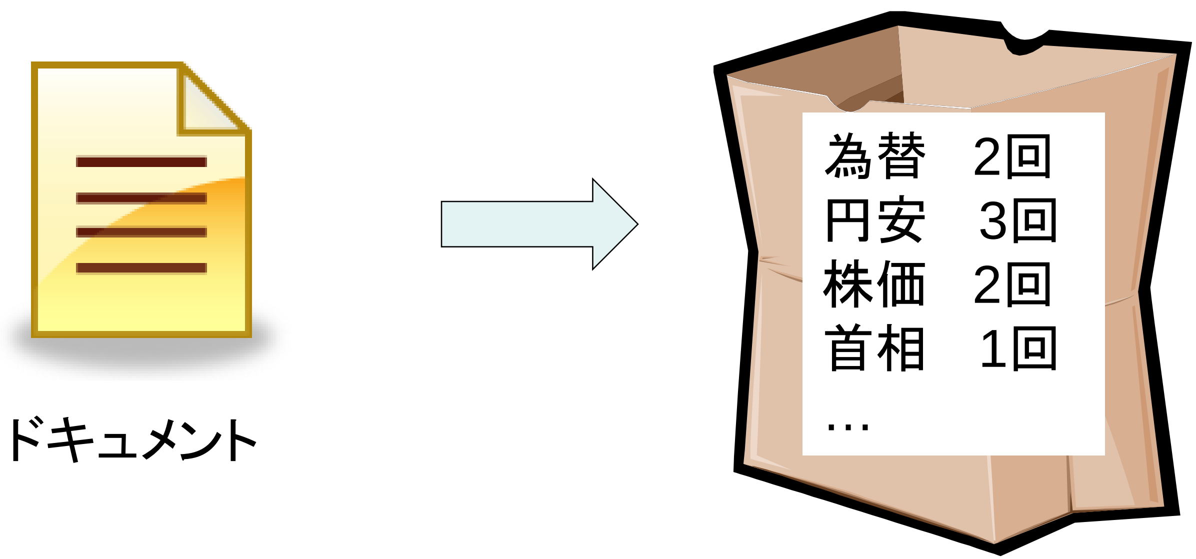
$$P(X,Y|Z) \cong P(X|Z)P(Y|Z)$$



  - Z が潜在的な構造をよく捉えていれば、比較的妥当な近似になると期待できる

ナイーブベイズ識別

- テキスト分類の基本的な手法
- ドキュメントを、出現する単語の集合として表現 =bag-of-words
  - 出現回数だけ利用
  - 各単語の位置や出現順などのコンテキストは考慮しない



ナイーブベイズ識別

ドキュメントDの事後確率

$$P(C \mid D) \propto \underline{P(D \mid C)}\,\underline{P(C)}$$

訓練サンプル中の比率で近似（あるいは単に一定）

ナイーブベイズ

$$\hat{P}(D \mid C) = P(W_1, W_2, \ldots, W_n \mid C) \propto P(W_1 \mid C)P(W_2 \mid C) \cdots P(W_n \mid C)$$

$$P(W_i \mid C) = \frac{\text{カテゴリ}C\text{に属する訓練データ中の単語}W_i\text{の数}}{\text{カテゴリ}C\text{に属する訓練データの全単語数}}$$

- あらかじめ、訓練データ中の単語を数え上げておくだけで識別ができる！

例）スパムメール識別 (notebook)

非スパム	スパム
Hi Peter, With Jose out of town, do you want to meet once in a while to keep things going and do some interesting stuff? Let me know Eugene	--- Codeine 15mg -- 30 for $203.70 -- VISA Only!!! -- -- Codeine (Methylmorphine) is a narcotic (opioid) pain reliever -- We have 15mg & 30mg pills -- 30/15mg for $203.70 - 60/15mg for $385.80 - 90/15mg for $562.50 -- VISA Only!!! ---

実行結果

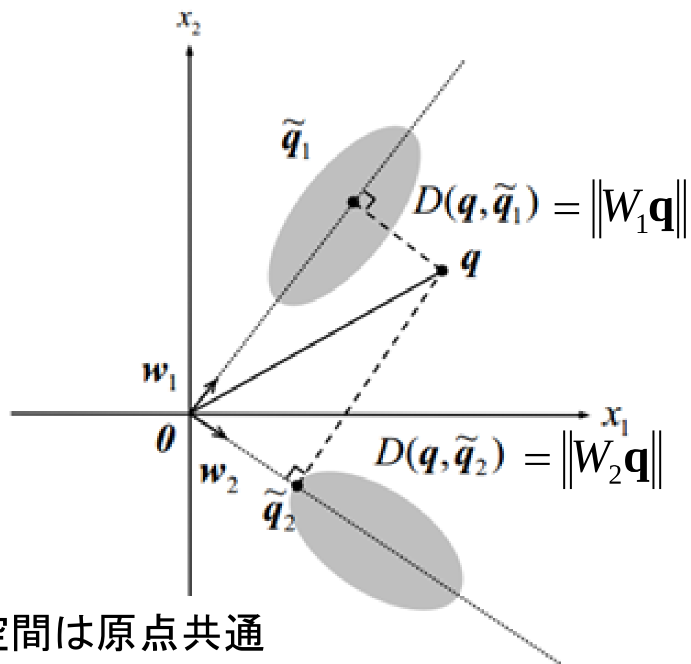
classification error ['home', 'based', 'business', 'opportunity', 'knocking', 'your', 'door', 'don', 'rude', 'and', 'let', 'this', 'chance', 'you', 'can', 'earn', 'great', 'income', 'and', 'find', 'your', 'financial', 'life', 'transformed', 'learn', 'more', 'here', 'your', 'success', 'work', 'from', 'home', 'finder', 'experts']

the error rate is: 0.1

(ランダムにサンプルを選ぶので毎回違った結果になる)

おまけ：部分空間法 (CLAFIC)

- 各クラスの成す部分空間 (PCAで学習）への近さを基準に識別
- 特徴量が線形な構造を有している場合に特に有効
- 日本発の技術



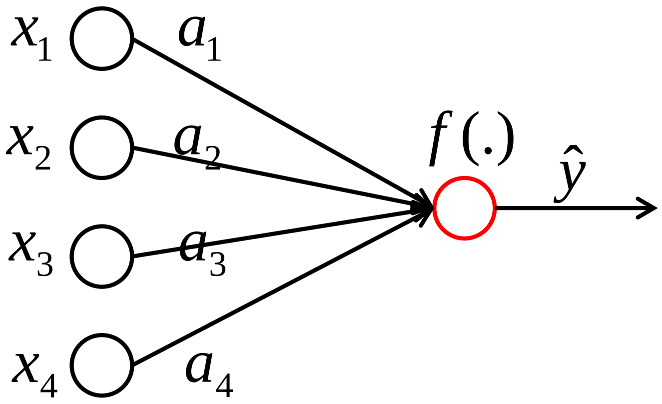
距離最小基準: $\hat{C} = \arg\min_{C_i} \|W_i\mathbf{q}\|$

角度最小基準: $\hat{C} = \arg\min_{C_i} \dfrac{\|W_i\mathbf{q}\|}{\|\mathbf{q}\|}$

※部分空間は原点共通
（自己相関行列によるPCA）

識別的アプローチ：線型識別関数＆識別モデル

- 要は単純パーセプトロン
  - 活性化関数 fと、誤差をどう考えるかで異なってくる



$$\hat{y} = f(\eta)$$

$$\eta = \mathbf{a}^T \mathbf{x}$$

$f(\eta) = \eta$ $\rightarrow$ 線形回帰と等価（二乗誤差）

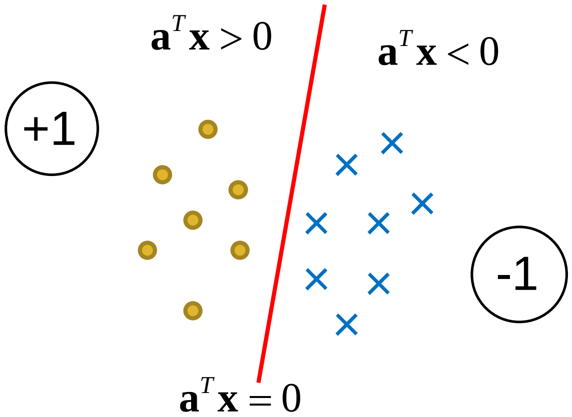
$f(\eta) = \dfrac{1}{1+\exp(-\eta)}$ $\rightarrow$ ロジスティック回帰と等価（ロジスティック損失）

線型識別関数

- 以下、2カテゴリの分類問題を扱う
  - 出力 (目的変数)は $y = \pm 1$ の二値とする

$$\hat{y} = \begin{cases} 1 & \mathbf{a}^T\mathbf{x} > 0 \\ -1 & \mathbf{a}^T\mathbf{x} < 0 \end{cases}$$

最終的には符号しか使わない



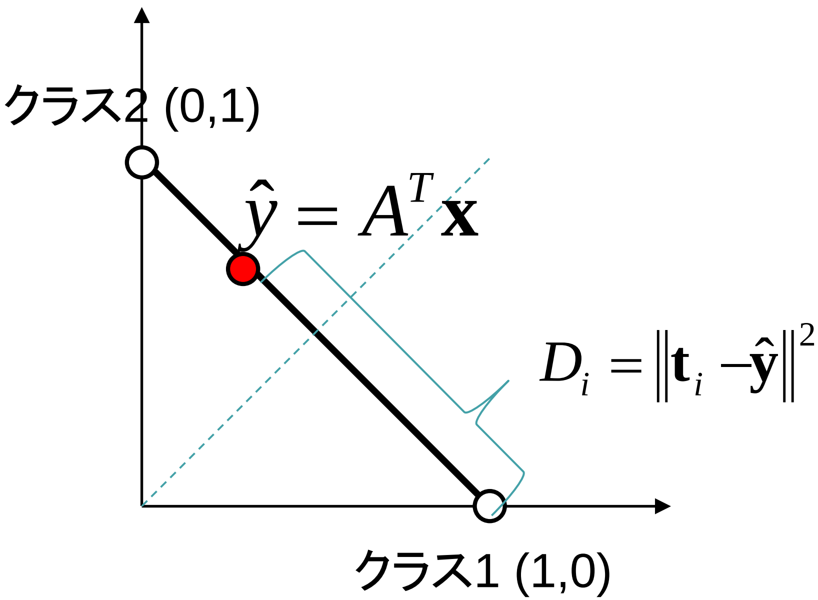
- 多クラス(Nクラス)の分類は、２クラス分類の線形識別関数の組み合わせで実現可能（詳しくは次回）
  - One-versus-all : あるクラスと、それ以外のクラス全てを識別する関数N個のうち、最も高い出力を出した識別器の結果をとる
  - One-versus-one : 全ての２クラス識別器 ${}_N C_2$ 個の多数決結果

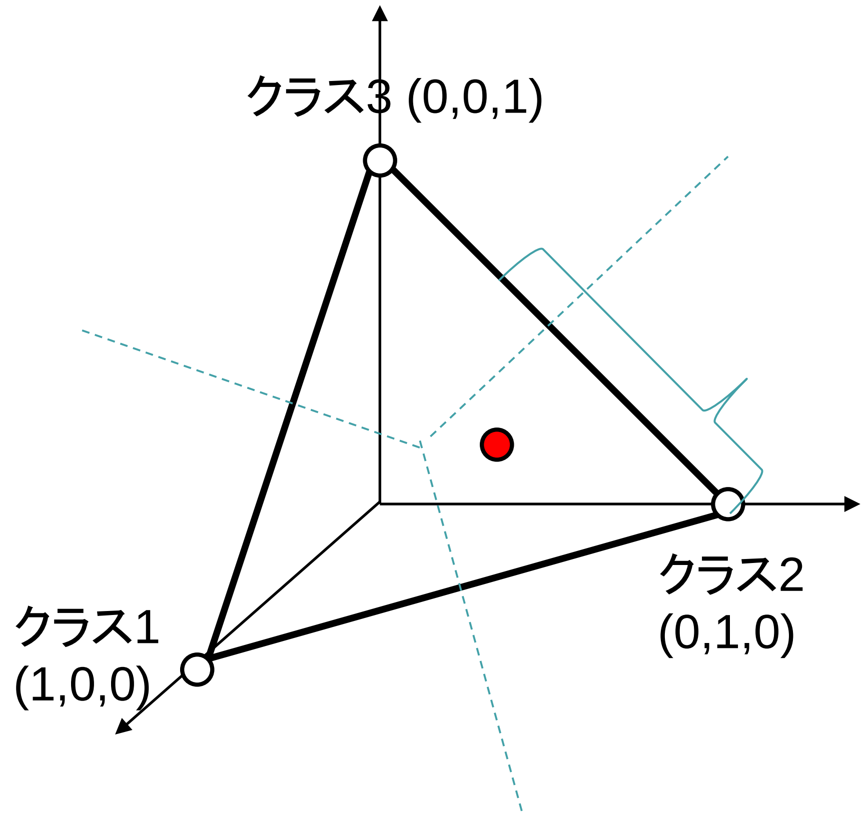
最小二乗識別

- 最も簡単な方法（解析解）
- 線形重回帰分析において、目的変数にダミー変数を導入しただけ
- 最小二乗線形判別ともよばれる $\hat{\mathbf{a}}_{LS} = \left(XX^T\right)^{-1} X\,\mathbf{y}$

- 意外と馬鹿にならない
  - ２クラス識別の場合は、フィッシャー判別分析による識別と等価
  - 多クラス識別の場合、クラス外分散を固定し、クラス内分散を最小化する線形射影となる
  - クラス代表ベクトルを基本正規直交底に取ると、最小２乗線形判別は **Bayes 識別則 の線形近似になっている** [大津,1981]

最小二乗識別





- 各クラスのone-hot-vectorとのユークリッド距離を測る
- 距離最小のクラスへ識別

損失関数の議論

- 準備：マージン $m_i = (\mathbf{a}^T \mathbf{x}_i) y_i$
  - 値が大きいほど、正しい側へ余裕をもって識別されていることを示す

- 二乗誤差 (L2損失) は、識別問題における精度の指標として適切か？

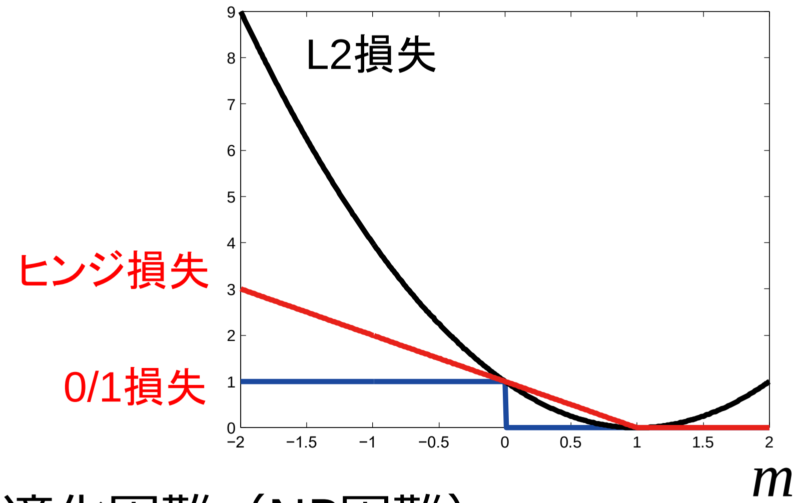
$$J_{LS}(\mathbf{a}) = \sum_{i=1}^{N} (y_i - \mathbf{a}^T \mathbf{x}_i)^2 = \sum_{i=1}^{N} y_i^2 \left(1 - \frac{\mathbf{a}^T \mathbf{x}_i}{y_i}\right)^2 = \sum_{i=1}^{N} (1 - (\mathbf{a}^T \mathbf{x}_i) y_i)^2 = \sum_{i=1}^{N} (1 - m)^2$$

$y_i = \pm 1$

損失関数の議論

- 理想的には0/1損失が望ましい
  - 重要なのは出力の符号だけ

$$J_{0/1}(\mathbf{a}) = \frac{1}{2}\sum_{i=1}^{N}\left(1 - \mathrm{sign}(m_i)\right)$$



  - しかし、原点で微分不可能なので最適化困難（NP困難）

- 仕方ないので、解ける範囲で近似
  ⇒ ヒンジ損失

$$J_{\mathrm{Hinge}}(\mathbf{a}) = \sum_{i=1}^{N}\max\left(0, 1 - m_i\right)$$

Support Vector Machine (SVM)

- 現在、最も標準的に用いられる識別器の一つ
- ヒンジ損失＋正則化項

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \left[\sum_{i=1}^{n} \max\left(0, 1 - \left(\mathbf{a}^T \mathbf{x}_i\right) y_i\right) + \lambda \left\|\mathbf{a}\right\|^2 \right]$$

- 経験的に、さまざまなタスクで優れた汎化性能を有する（とされる）
- ライブラリ多数 (libsvm等)

Support Vector Machine (SVM)

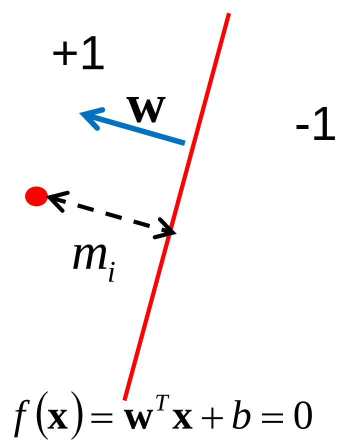
- 現在、最も標準的に用いられる識別器の一つ
- ヒンジ損失＋正則化項

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \left[\sum_{i=1}^{n} \max\left(0, 1 - \left(\mathbf{w}^T \mathbf{x}_i + b\right) y_i\right) + \lambda \|\mathbf{w}\|^2 \right]$$

- (最小)マージンの最大化を行う手法
  - あるデータi の正規化マージン

$$m_i = \frac{y_i f(\mathbf{x}_i)}{\|\mathbf{w}\|} \qquad f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

= データ点と分離超平面との距離



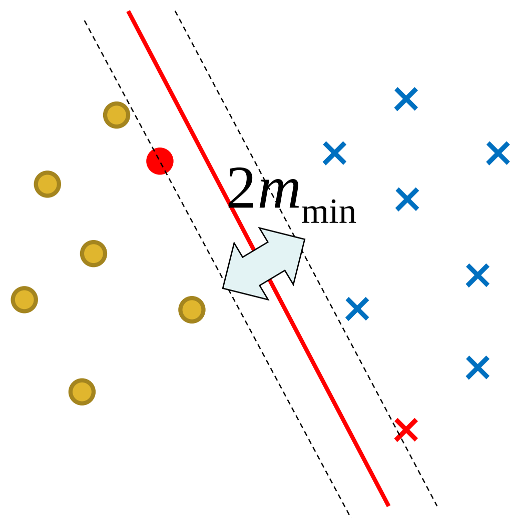
最小マージン

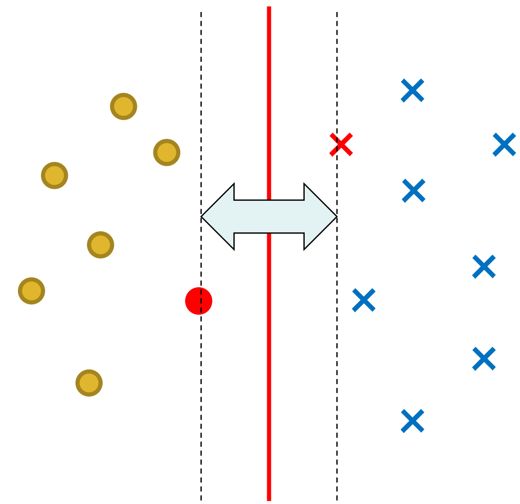
- 最小マージン：データごとの正規化マージンの最小値
  - その識別平面にとって、もっとも「難しい」データ（＝サポートベクター）のマージン

$$m_{\min} = \min m_i$$

- 最小マージンを最大とする識別平面は一般に汎化性が高い

どっちがよい識別平面？





ハードマージンSVM

- 最小マージンを最大化する識別超平面を求める

$$\max_{\mathbf{w},b}\left(\min_i y_i f(\mathbf{x}_i)/\|\mathbf{w}\|\right) \qquad f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$

  - 線型分離可能（全学習データを正しく分離する平面が引ける）ことが前提

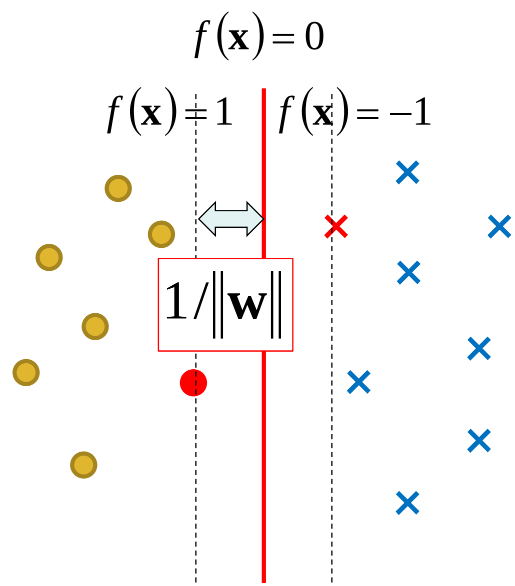
- 二次計画問題（等価な表現）

$$\min_{\mathbf{w},b}\|\mathbf{w}\|^2 \quad \longleftrightarrow \quad \max 1/\|\mathbf{w}\|$$

subject to $\quad y_i f(\mathbf{x}_i) \geq 1 \;\; \text{for} \;\; \forall i$

$y_i f(\mathbf{x}_i) > 0$

w,bに関してスケールの自由度があるので置き換え可能（サポートベクターがf(x)=±1上にのるように）



線形SVM：双対化

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$

主問題

subject to $\quad y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq 1 \;\text{ for }\; \forall i$

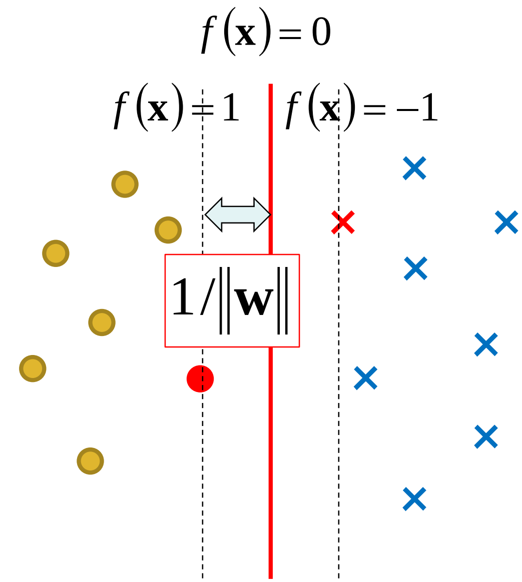
このまま最適化は難しいので
ラグランジュ未定乗数 $\{\alpha_i\}_{i=1}^n$
を導入して書き換えると…

$$\max_{\mathbf{w},b,\boldsymbol{\alpha}} \left[\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^n \alpha_i \left\{1 - y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right)\right\} \right]$$

subject to $\quad \alpha_i \geq 0 \quad \text{for} \;\; \forall i$

$\mathbf{w}, b$ で偏微分すると

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \qquad \sum_{i=1}^n \alpha_i y_i = 0$$



線形SVM：双対化（続き）

これらを戻すと…

双対問題

$$\max_{\boldsymbol{\alpha}} \left[\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \underline{\mathbf{x}_i^T \mathbf{x}_j} \right]$$

内積しか出てこない

subject to $\underline{\alpha_i \geq 0}$ for $\forall \mathrm{i}$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$\alpha_i > 0$ に対応する $\mathbf{x}_i$ が
サポートベクター

$$y_i \left(\mathbf{w}^T \mathbf{x}_i + b \right) \geq 1,$$

$$\alpha_i \left(y_i \left(\mathbf{w}^T \mathbf{x}_i + b \right) - 1 \right) = 0 \quad \text{for} \quad \forall i$$

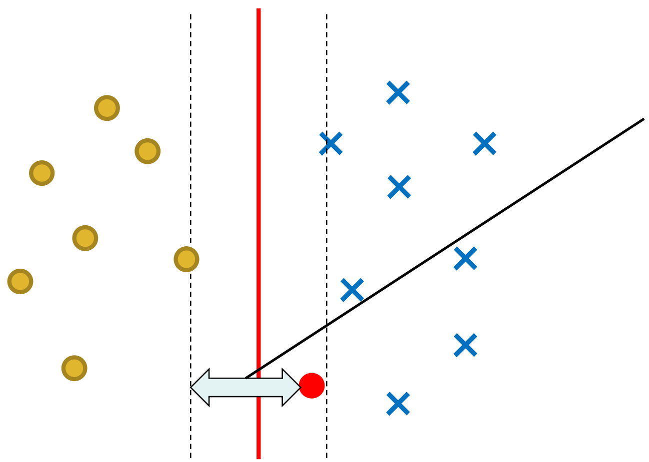
KKT条件

Karush-Kuhn-Tucker condition

を満たす $\mathbf{w}, b$ が解として得られる
つまり全てのデータは $\alpha_i = 0$ または $y_i \left(\mathbf{w}^T \mathbf{x}_i + b \right) = 1$

ソフトマージンSVM

- 実際には線形分離可能でないことがほとんどなので、少しの誤差 ξ_i を各データに許容する



ξ_i ：データ i が超平面の反対側に
はみ出した距離
（スラック変数）

$$\min_{\mathbf{w},b,\xi}\left(\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i\right)$$

$C > 0$ は許容誤差の程度を決めるパラメータ
（とても重要！）

subject to $\quad y_i f(\mathbf{x}_i) \geq 1 - \xi_i,\ \ \xi_i \geq 0 \ \text{ for } \ \forall \mathrm{i}$

ソフトマージン線形SVM：双対化

$$\max_{\mathbf{w},b,\boldsymbol{\alpha},\boldsymbol{\mu}} \left[\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i\left\{1 - y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) - \xi_i\right\} - \sum_{i=1}^{n}\mu_i\xi_i \right]$$

subject to $\alpha_i \geq 0,\ \ \mu_i \geq 0 \quad \text{for}\ \ \forall i$

KKT条件: $\alpha_i \geq 0, \mu_i \geq 0, \xi_i \geq 0$

$$1 - y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) - \xi_i \leq 0$$

$$\alpha_i\left(1 - y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) - \xi_i\right) = 0$$

$$\mu_i\xi_i = 0$$

$\mathbf{w}, b, \xi_i$ で偏微分すると

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i \qquad \sum_{i=1}^{n}\alpha_i y_i = 0 \qquad \alpha_i = C - \mu_i$$

線形SVM：双対化（続き）

以上まとめると…

双対問題

$$\max_{\boldsymbol{\alpha}} \left[\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \underline{\mathbf{x}_i^T \mathbf{x}_j} \right]$$

subject to $\underline{0 \le \alpha_i \le C}$ for $\forall \mathrm{i}$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

解き方

- 最急降下法（勾配法）でもよいが…
  - 効率はよくない（特にカーネル法を使う場合）
  - 制約条件を満たすための工夫が必要

- <u>ワーキング集合法</u>
  - 教師データを分割して部分的に解くことを繰り返す
  - これが使えるのが双対化の実用的なメリット

Sequential Minimal Optimization (SMO)

[J. Platt, 1998]

- 二つ組の問題を繰り返し解く

  $$0 \le \alpha_1, \alpha_2 \le C$$

  $\alpha_1 y_1 + \alpha_2 y_2 = k$　　の条件下で α_1, α_2 を最適化

- アルゴリズム
  - 1. KKT条件を破るラグランジュ乗数 α_1 を見つける。
  - 2. 第2の乗数 α_2 を選び、α_1, α_2 のペアを最適化する。
  - 3. 収束するまで1、2を繰り返す。

- 詳しくはコード参照…

ロジスティック回帰

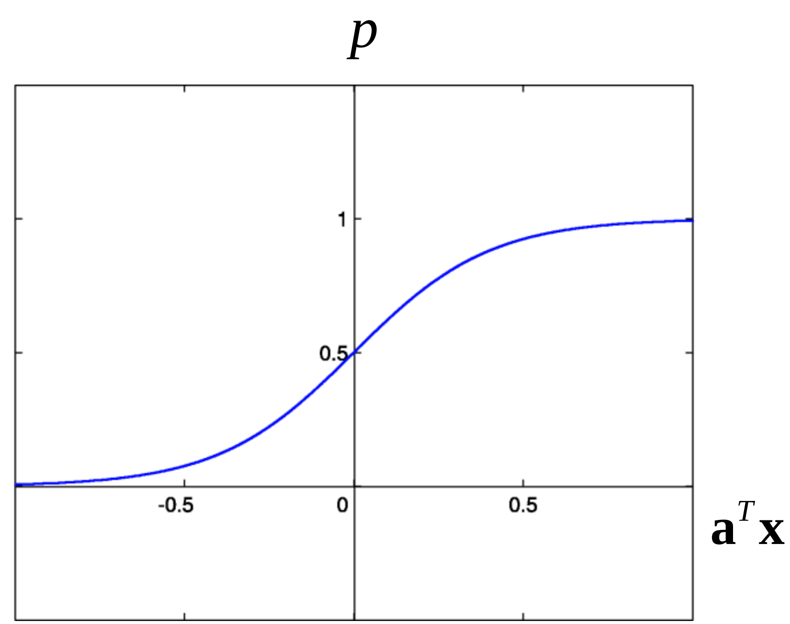
- 事後確率を直接推定（＝識別モデル）
- 二値データ（質的データ）の回帰

$$P(y=1|\mathbf{x}) = p$$

$$P(y=-1|\mathbf{x}_i) = 1-p$$

$$p = \frac{\exp(\mathbf{a}^T\mathbf{x})}{1+\exp(\mathbf{a}^T\mathbf{x})}$$

（ロジスティック関数）



最尤推定による解き方

- 最尤推定

$$t_i = \begin{cases} 1 & \text{if } y_i = 1 \\ 0 & \text{if } y_i = -1 \end{cases}$$

訓練データ集合 $\{\mathbf{x}_i, t_i\},\ t_i \in \{0, 1\}$

尤度関数は $L = \prod_{i=1}^{N} p_i^{t_i} \{1 - p_i\}^{1-t_i}$ $\qquad p_i = \dfrac{\exp(\mathbf{a}^T \mathbf{x}_i)}{1 + \exp(\mathbf{a}^T \mathbf{x}_i)}$

負の対数尤度は $E(\mathbf{a}) = -\ln L = \sum_{i=1}^{N} \{t_i \ln p_i + (1 - t_i)\ln(1 - p_i)\}$

$$\frac{E(\mathbf{a})}{\partial \mathbf{a}} = \sum_{i=1}^{N} \underline{(p_i - t_i)\mathbf{x}_i}$$

エラーに説明変数をかけたもの

損失関数の観点からの解釈

- 負の対数尤度を整理すると

$$E(\mathbf{a}) = \sum_{i=1}^{N} \{t_i \ln p_i + (1-t_i)\ln(1-p_i)\}$$

$$= -\sum_{i=1}^{N} \ln\{1+\exp(-y_i \mathbf{a}^T \mathbf{x}_i)\} = -\sum_{i=1}^{N} \ln\{1+\exp(-m_i)\}$$

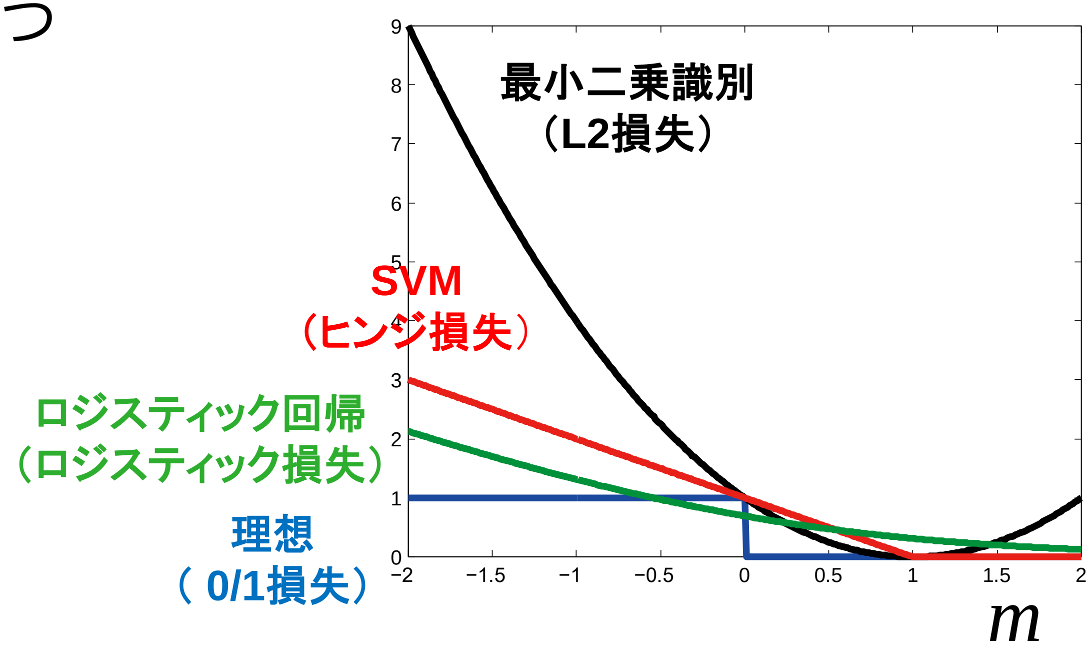
- つまり、対数尤度最大化規準は

$$\underset{\mathbf{a}}{\arg\min} \sum_{i=1}^{N} \ln(1+\exp(-m_i))$$

と書けることから、ロジスティック損失 $\ln(1+\exp(-m))$
を最小化する識別関数の学習を行っていることが分かる

識別的アプローチ（線形モデル）

- 大枠としては、どれもパーセプトロン
- 損失の測り方が違う



- 実用的には
  - とりあえず、SVM、ロジスティック回帰を試してみることが多い
  - 正則化などは、回帰の章と同様のテクニックが導入可能

まとめ

- クラス分類（クラス識別）
  - 前提として、特徴抽出は重要
  - ベイズの定理、事後確率、ベイズ識別則
  - 識別的アプローチ、生成的アプローチの違い

- 識別的アプローチ：クラス間の“違い”だけ分かればよい
  - 線形識別関数：SVM、最小二乗識別
  - 線形識別モデル：ロジスティック回帰
- 生成的アプローチ：クラスの分布も知りたい
  - k-NN、ナイーブベイズ

- 次回：非線形識別