

統計的モデリング基礎⑤ ～ロジスティック回帰～

鹿島久嗣
(情報学科 計算機科学コース)

ロジスティック回帰



最尤推定：

データをもっともよく再現するパラメータを推定値とする

- n 個のデータ $x^{(1)}, x^{(2)}, \dots, x^{(n)}$ から確率モデル $f_{\theta}(x)$ のパラメータ θ を推定したい

- n 個のデータが（互いに独立に）生成される確率（尤度）：

$$L(\theta) = \prod_{i=1}^n f_{\theta}(x^{(i)})$$

- 尤度最大になるパラメータを推定値 $\hat{\theta}$ とする

実際には対数尤度で扱うことが多い

$$\hat{\theta} = \operatorname{argmax}_{\theta} \prod_{i=1}^n f_{\theta}(x^{(i)}) = \operatorname{argmax}_{\theta} \sum_{i=1}^n \log f_{\theta}(x^{(i)})$$

- もっともデータを生成する確率が高い（「最も尤もらしい」）

条件付き分布の最尤推定：

従属変数の値を最もよく再現するパラメータを推定値とする

- n 個のデータ： $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})\}$ から条件付き確率モデル $f_{\theta}(y^{(i)} | x^{(i)})$ のパラメータ θ を推定したい
- n 個のデータが（互いに独立に）生成される確率（尤度）：

$$L(\theta) = \prod_{i=1}^n f_{\theta}(y^{(i)} | x^{(i)})$$

- 尤度最大になるパラメータを推定値 $\hat{\theta}$ とする

$$\hat{\theta} = \operatorname{argmax}_{\theta} \prod_{i=1}^n f_{\theta}(y^{(i)} | x^{(i)}) = \operatorname{argmax}_{\theta} \sum_{i=1}^n \log f_{\theta}(y^{(i)} | x^{(i)})$$

- もっともデータを生成する確率が高い（「最も尤もらしい」）

最尤推定の利点： モデリングの自動化

- 最尤推定の利点：
確率モデルの形（データの生成プロセスの仮定）を決めればモデルパラメータが自動的に決まる
 - ただし、最大化問題を解く必要がある
 - 離散分布、ポアソン分布、正規分布などは解析的に解が求まる
 - 線形回帰（正規分布でノイズが載る）は連立方程式（一応、解析的な解）
 - 多くのモデルでは、最適化問題を数値的に解く必要がある

判別問題：

ダミー変数を従属変数として説明（予測）する問題

- データ（ n 組の独立変数と従属変数）
 - 独立変数： $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)})$
 - （ダミー）従属変数： $(y^{(1)}, y^{(2)}, \dots, y^{(n)}), y^{(i)} \in \{+1, -1\}$

以降、表記上の利便性からダミー従属変数を
 $\{0, 1\}$ でなく $\{+1, -1\}$ と表記する
(本質的な違いはナシ)

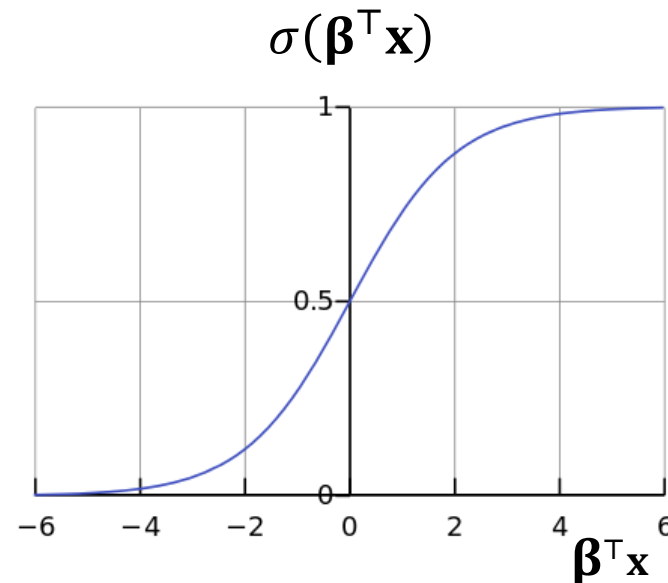
ロジスティック回帰モデル： ダミー変数を従属変数とするモデル

- 以前、重回帰モデルでダミー変数を従属変数とすると、
厳密には少しおかしいという話だった → もっときちんと扱いたい
 - 重回帰モデル $y = \boldsymbol{\beta}^\top \mathbf{x}$ の従属変数の値域は実数全体
- 従属変数の値域が $\{-1, +1\}$ もしくは $(0,1)$ ($Y = +1$ となる確率) となるようにしたい

- ロジスティック回帰モデル：

$$P(Y = 1 | \mathbf{x}; \boldsymbol{\beta}) = \frac{1}{1 + \exp(-\boldsymbol{\beta}^\top \mathbf{x})} = \sigma(\boldsymbol{\beta}^\top \mathbf{x})$$

- σ : ロジスティック関数 ($\sigma: \mathbb{R} \rightarrow (0,1)$)

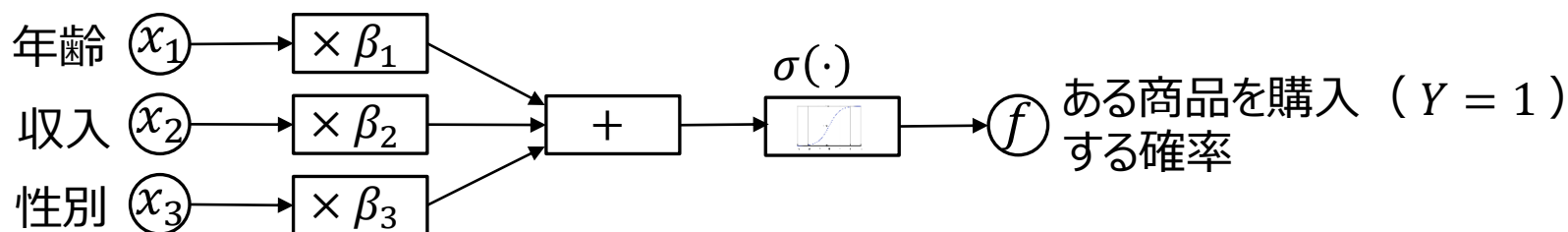


ロジスティック回帰モデルの例： 線形回帰モデルの出力を $[0,1]$ に変換

- ロジスティック回帰モデルは従属変数 $Y = 1$ となる確率を与える：

$$P(Y = 1|\mathbf{x}; \boldsymbol{\beta}) = \sigma(\beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_D x_D + \alpha)$$

- σ ：ロジスティック関数 ($\sigma: \mathbb{R} \rightarrow (0,1)$)
 - $\sigma(\cdot)$ の中身は線形回帰モデルと同じ ($\in \mathbb{R}$)
- モデルパラメータ $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_D, \alpha)^\top$ において
 - β_d ：独立変数 x_d が従属変数に与える影響
 - $\beta_d > 0$ のとき、 $x_d > 0$ は $Y = 1$ となる方向に貢献；
 $x_d < 0$ は $Y = -1$ となる方向に貢献している

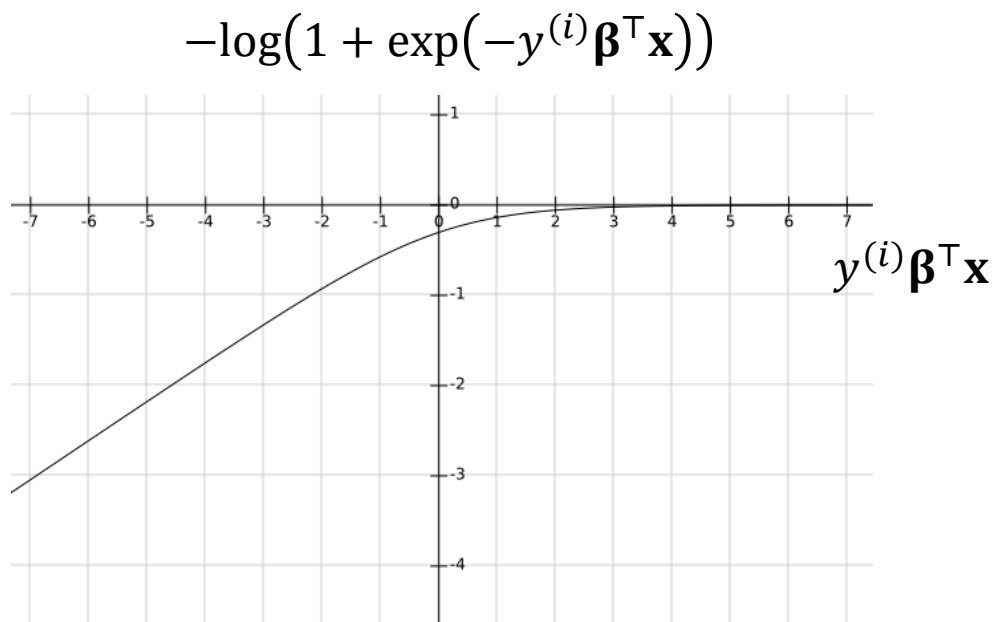


ロジスティック回帰モデルの対数尤度： 凸関数なので大局解が存在するが解析解はない

■ 対数尤度： $L(\boldsymbol{\beta}) = -\sum_{i=1}^n \log(1 + \exp(-y^{(i)} \boldsymbol{\beta}^\top \mathbf{x}^{(i)}))$
 $\left(= \sum_{i=1}^n \delta(y^{(i)} = 1) \log \frac{1}{1 + \exp(-\boldsymbol{\beta}^\top \mathbf{x}^{(i)})} + \delta(y^{(i)} = -1) \log \left(1 - \frac{1}{1 + \exp(-\boldsymbol{\beta}^\top \mathbf{x}^{(i)})} \right) \right)$

■ $L(\boldsymbol{\beta})$ は凸関数：

- 大局解がある
- 解析解はない



ロジスティック回帰のパラメータ推定： 非線形最適化法によって、パラメータ更新を繰り返す

- 最尤推定の目的関数（最大化）：

$$L(\boldsymbol{\beta}) = - \sum_{i=1}^n \log(1 + \exp(-y^{(i)} \boldsymbol{\beta}^\top \mathbf{x}^{(i)}))$$

- 解析解は得られないが、凸関数（1次元の場合、2階微分が ≤ 0 ）
- 数値的な最適化手法を使う
 - パラメータの更新をくりかえす： $\boldsymbol{\beta}^{\text{NEW}} \leftarrow \boldsymbol{\beta} + \mathbf{d}$



パラメータの更新：

目的関数をもっとも改善するような更新を行う

- 更新 $\boldsymbol{\beta}^{\text{NEW}} \leftarrow \boldsymbol{\beta} + \mathbf{d}$ によって目的関数の値が変化する：

$$L_{\boldsymbol{\beta}}(\mathbf{d}) = - \sum_{i=1}^n \ln(1 + \exp(-y^{(i)}(\boldsymbol{\beta} + \mathbf{d})^{\top} \mathbf{x}^{(i)}))$$

- $L_{\boldsymbol{\beta}}(\mathbf{d})$ を最大化する更新差分 \mathbf{d}^* を見つけよ：

$$\mathbf{d}^* = \operatorname{argmax}_{\mathbf{d}} L_{\boldsymbol{\beta}}(\mathbf{d})$$

- 注意：この段階では問題は全く簡単になっていない
($\boldsymbol{\beta} = \mathbf{0}$ の場合は元の問題と全く同じ)

最良のパラメータ更新差分の決定： 目的関数をテイラー展開で2次近似

- 目的関数のテイラー展開（ β のまわりでの）：

3次以上の項

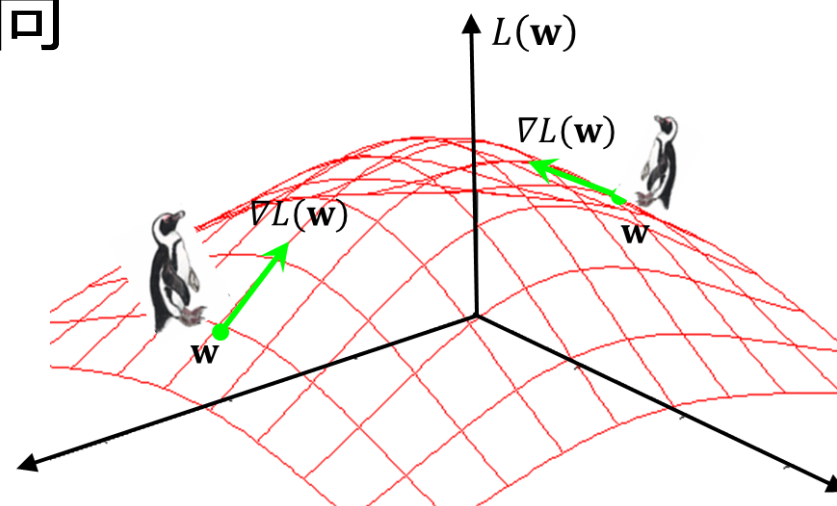
$$L_{\beta}(\mathbf{d}) = L(\beta) + \mathbf{d}^{\top} \nabla L(\beta) + \frac{1}{2} \mathbf{d}^{\top} \mathbf{H}(\beta) \mathbf{d} + O(\mathbf{d}^3)$$

- 勾配： $\nabla L(\beta) = \left(\frac{\partial L(\beta)}{\partial \beta_1}, \frac{\partial L(\beta)}{\partial \beta_2}, \dots, \frac{\partial L(\beta)}{\partial \beta_D} \right)^{\top}$

- β において目的関数が最も急な方向

- ヘッセ行列： $[H(\beta)]_{i,j} = \frac{\partial^2 L(\beta)}{\partial \beta_i \partial \beta_j}$

- β 周辺での目的関数の「曲がり方」



ニュートン法：

2次近似した目的関数を最小化する解を求める

- テイラー展開で3次以降の項を無視する：

3次以上の項

$$L_{\beta}(\mathbf{d}) \approx L(\beta) + \mathbf{d}^{\top} \nabla L(\beta) + \frac{1}{2} \mathbf{d}^{\top} \mathbf{H}(\beta) \mathbf{d} - \cancel{O(\mathbf{d}^3)}$$

- 最大化するために \mathbf{d} で微分： $\frac{\partial L_{\beta}(\mathbf{d})}{\partial \mathbf{d}} \approx \nabla L(\beta) + \mathbf{H}(\beta) \mathbf{d}$

- これを $= \mathbf{0}$ とおいて解くと： $\mathbf{d} = -\mathbf{H}(\beta)^{-1} \nabla L(\beta)$

実際には連立方程式を解く

- ニュートン法：

$$\beta^{\text{NEW}} \leftarrow \beta - \mathbf{H}(\beta)^{-1} \nabla L(\beta)$$



線形探索付きニュートン法：

近似は厳密には正しくないので線形探索と組み合わせる

- ニュートン法の更新 $\boldsymbol{\beta}^{\text{NEW}} \leftarrow \boldsymbol{\beta} - \boldsymbol{H}(\boldsymbol{\beta})^{-1} \nabla L(\boldsymbol{\beta})$ は2次近似が正確であることを仮定している：

$$L_{\boldsymbol{\beta}}(\mathbf{d}) \approx L(\boldsymbol{\beta}) + \mathbf{d}^{\top} \nabla L(\boldsymbol{\beta}) + \frac{1}{2} \mathbf{d}^{\top} \boldsymbol{H}(\boldsymbol{\beta}) \mathbf{d}$$

- しかし、**近似**なので、厳密には正しくない
- そこで、更新の向きのみを採用して、更新の量 η は別途決める：
 $\boldsymbol{\beta}^{\text{NEW}} \leftarrow \boldsymbol{\beta} - \eta \boldsymbol{H}(\boldsymbol{\beta})^{-1} \nabla L(\boldsymbol{\beta})$
- 更新の量（学習率） $\eta > 0$ の決定法：
 - ステップ数とともに適当に減衰させる
 - あるいは、線形探索： $\eta^* = \operatorname{argmax}_{\eta} L(\boldsymbol{\beta} - \eta \boldsymbol{H}(\boldsymbol{\beta})^{-1} \nabla L(\boldsymbol{\beta}))$

適当な初期値から始めて、
目的関数が改善しない間
は η を半分にしていく

最急降下法※:

ヘッセ行列を使わずに、シンプルで軽い更新を繰り返す

- ヘッセ行列の逆行列（もしくは連立方程式を解く）は高コスト：

- ニュートン法の更新： $\beta^{\text{NEW}} \leftarrow \beta - \eta H(\beta)^{-1} \nabla L(\beta)$

- 最急降下法：

単位行列

- ヘッセ行列の逆行列 $H(\beta)^{-1}$ を $-I$ で置き換える：

$$\beta^{\text{NEW}} \leftarrow \beta + \eta \nabla L(\beta)$$

- 勾配 $\nabla L(\beta)$ は最も急な（目的関数が最も増加する）向き
- 学習率 η は線形探索で求める：



※ 最急降下法の名前は、最小化問題を想定しているが、ここでは最大化問題としているので降下じゃなくて上昇しているが、便宜上こう呼ぶことにする

確率的最適化とミニバッチ学習： データの部分集合を用いた効率的な推定

- 目的関数は各データの対数尤度の和： $L(\boldsymbol{\beta}) = \sum_{i=1}^n \ell^{(i)}$
- 勾配 $\frac{\partial L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^n \frac{\partial \ell^{(i)}}{\partial \boldsymbol{\beta}}$ の計算は $O(n)$ かかる
- 勾配をデータ1個で近似する： $\frac{\partial L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \approx n \frac{\partial \ell^{(i)}}{\partial \boldsymbol{\beta}}$
 - 確率的最適化：毎回データをランダムに選ぶ
 - オンライン推定も可能（時刻 t のデータの $\ell^{(t)}$ を使う）
- あるいはミニバッチ学習： $1 < m < n$ 個のデータで勾配を近似：
$$\frac{\partial L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \approx \frac{n}{m} \sum_{j \in \text{MiniBatch}} \frac{\partial \ell^{(j)}}{\partial \boldsymbol{\beta}}$$

i 番目のデータの
対数尤度

ロジスティック回帰の勾配計算： 比較的簡単に計算可能

- 対数尤度： $L(\boldsymbol{\beta}) = -\sum_{i=1}^n \ln(1 + \exp(-y^{(i)} \boldsymbol{\beta}^\top \mathbf{x}^{(i)}))$

- $$\begin{aligned} \frac{\partial L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} &= -\sum_{i=1}^n \frac{1}{1 + \exp(-y^{(i)} \boldsymbol{\beta}^\top \mathbf{x}^{(i)})} \frac{\partial (1 + \exp(-y^{(i)} \boldsymbol{\beta}^\top \mathbf{x}^{(i)}))}{\partial \boldsymbol{\beta}} \\ &= \sum_{i=1}^n \frac{1}{1 + \exp(-y^{(i)} \boldsymbol{\beta}^\top \mathbf{x}^{(i)})} \exp(-y^{(i)} \boldsymbol{\beta}^\top \mathbf{x}^{(i)}) y^{(i)} \mathbf{x}^{(i)} \\ &= \sum_{i=1}^n (1 - f(y^{(i)} | \mathbf{x}^{(i)}; \boldsymbol{\beta})) y^{(i)} \mathbf{x}^{(i)} \end{aligned}$$

現在のパラメータ $\boldsymbol{\beta}$ におけるロジスティック回帰モデルが
各データに対して与える確率

練習問題：

ポアソン回帰の最尤推定

- （前回出てきた）ポアソン回帰の最尤推定

- 対数尤度：

$$L(\boldsymbol{\beta}) = \sum_{i=1}^n y^{(i)} \boldsymbol{\beta}^\top \mathbf{x}^{(i)} - \sum_{i=1}^n \exp(\boldsymbol{\beta}^\top \mathbf{x}^{(i)}) + \text{const.}$$

- ロジスティック回帰同様、これも解析解は求まらない
- 最急勾配法の更新式を求めている

ロジスティック回帰モデルの発展

ロジスティック回帰モデルの発展： 従属変数の型に合わせた発展

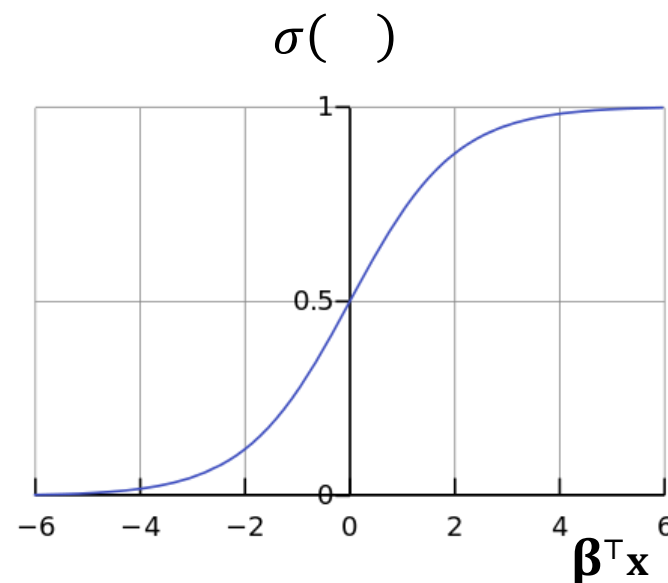
- 確率モデルはデータの生成モデルである
- 分析対象のデータに合わせてモデルが変わる
 - 質的変数（ダミー変数0, 1）の場合：ロジスティック回帰モデル
→ 選択肢が複数の場合：多値ロジスティック回帰
 - 量的変数（連続値）の場合：線形回帰モデル
→ 順序尺度（例えば5段階評価）の場合：順序回帰
 - 比較：一対比較のモデル（例：2つのうちどちらがよいか？）
- 多層化による非線形モデルの実現：ニューラルネットワーク

ロジスティック回帰モデル： ダミー変数を従属変数とするモデル

- 従属変数 Y が（2値の）ダミー変数であるモデル
- ロジスティック回帰モデル： $Y = +1$ となる確率

$$P(Y = 1|\mathbf{x}; \boldsymbol{\beta}) = \frac{1}{1+\exp(-\boldsymbol{\beta}^\top \mathbf{x})} = \sigma(\boldsymbol{\beta}^\top \mathbf{x})$$

- σ ：ロジスティック関数（ $\sigma: \mathbb{R} \rightarrow (0,1)$ ）



多値ロジスティック回帰モデル： 多値の従属変数を説明するモデル

- 従属変数 Y が多値である場合（ $Y \in \mathcal{Y} = \{1, 2, \dots, k\}$ ）

- ただし、 $1, 2, \dots, k$ の並び順に意味はないことに注意

- 多値ロジスティック回帰モデル： $Y = y$ である確率

- 各 $y \in \mathcal{Y}$ ごとにパラメータ $\boldsymbol{\beta}_y$ をもつ

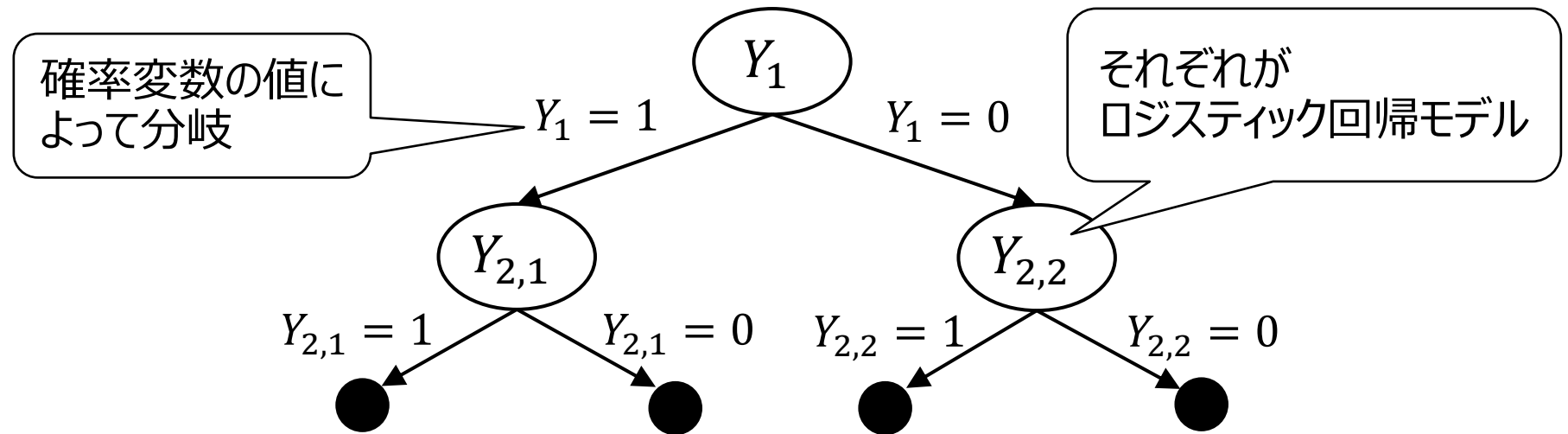
いわゆるソフトマックス関数

$$P(Y = y | \mathbf{x}; \{\boldsymbol{\beta}_{y'}\}_{y' \in \mathcal{Y}}) = \frac{\exp(\boldsymbol{\beta}_y^\top \mathbf{x})}{\sum_{y' \in \mathcal{Y}} \exp(\boldsymbol{\beta}_{y'}^\top \mathbf{x})}$$

- $\mathcal{Y} = \{+1, -1\}$ のときは $\boldsymbol{\beta} = \boldsymbol{\beta}_{+1} - \boldsymbol{\beta}_{-1}$ とすると通常のロジスティック回帰に一致

多段ロジスティック回帰モデル： 多段に連結されたロジスティック回帰モデル

- 複数の連続した従属変数：
 - 階層的な分類（大カテゴリ→中カテゴリ→小カテゴリ）
 - 段階的な意思決定プロセス（購入の有無→商品）
- ロジスティック回帰モデルを連結する



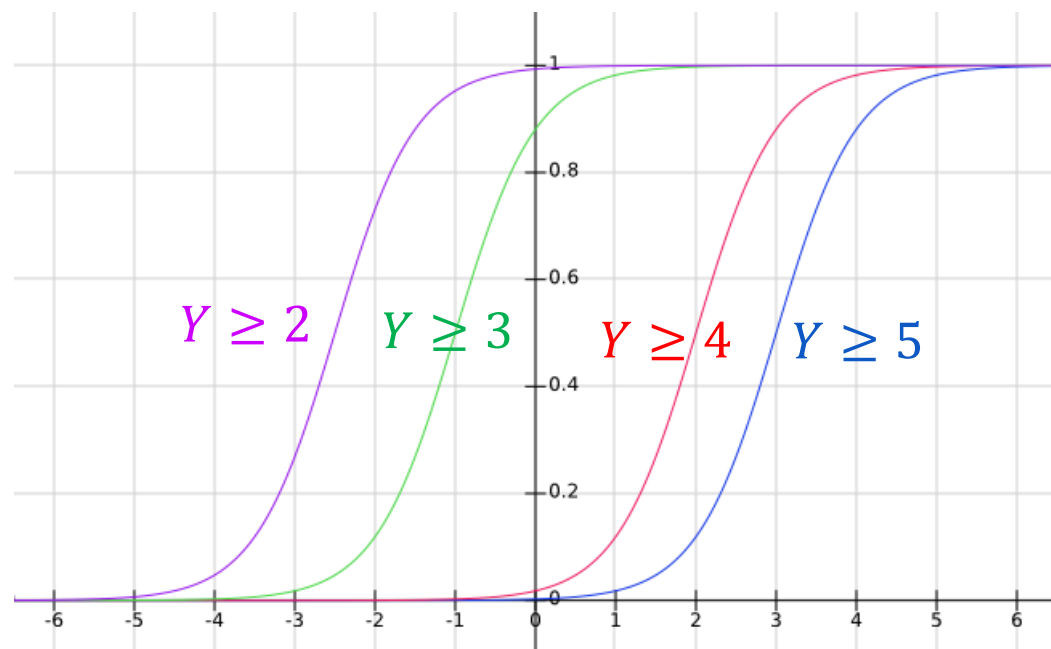
順序回帰： 順序尺度をもつ従属変数を説明するモデル

この順序に意味がある場合
(5段階評価など)

- Y が多値で順序尺度をもつ場合 ($Y \in \mathcal{Y} = \{1, 2, \dots, k\}$)
- $Y \geq y$ となる確率を与えるモデルを考える：

$$P(Y \geq y | \mathbf{x}; \boldsymbol{\beta}) = \frac{1}{1 + \exp(-\boldsymbol{\beta}^\top \mathbf{x} + \alpha_y)} = \sigma(\boldsymbol{\beta}^\top \mathbf{x} - \alpha_y)$$

- パラメータ $\boldsymbol{\beta}$ は y によらず共通
- 「切片」 α_y が y ごとに異なる：
 $\alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_k$
(平行移動)



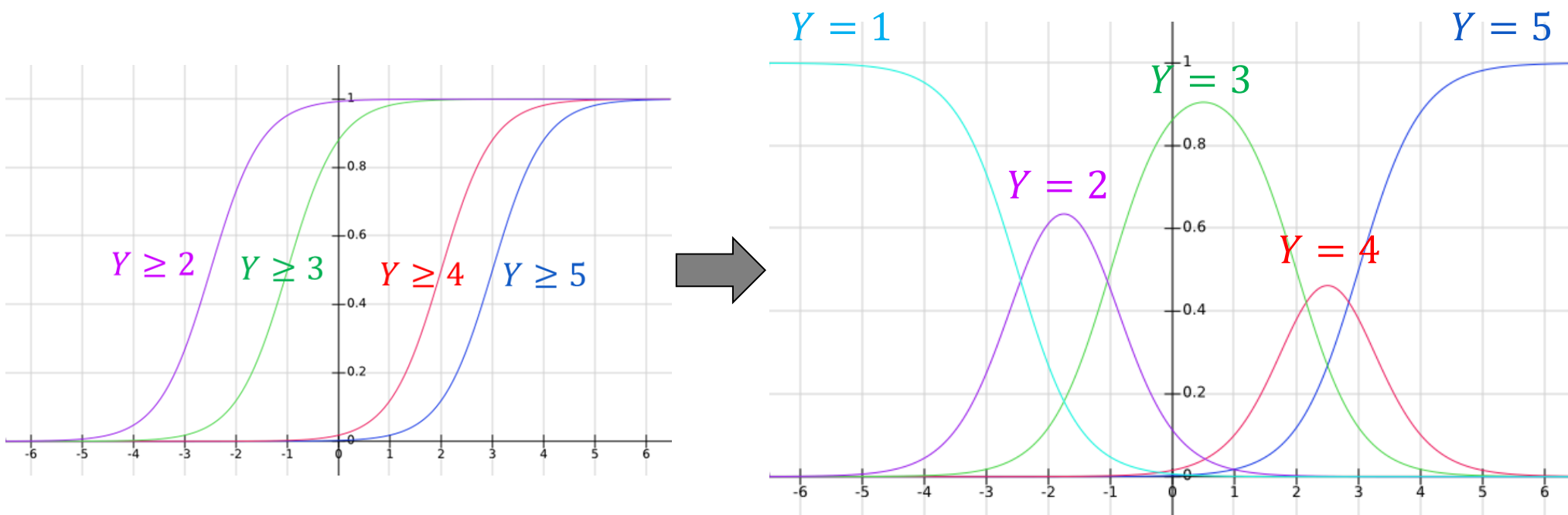
順序回帰モデル： 順序尺度をもつ従属変数を説明するモデル

- 順序回帰モデルは $Y \geq y$ となる確率がロジスティック回帰モデル：

$$P(Y \geq y | \mathbf{x}; \boldsymbol{\beta}) = \sigma(\boldsymbol{\beta}^\top \mathbf{x} - \alpha_y)$$

- $Y = y$ である確率は上記モデルの差を用いて表現できる：

$$P(Y = y | \mathbf{x}; \boldsymbol{\beta}) = P(Y \geq y | \mathbf{x}; \boldsymbol{\beta}) - P(Y \geq y + 1 | \mathbf{x}; \boldsymbol{\beta})$$

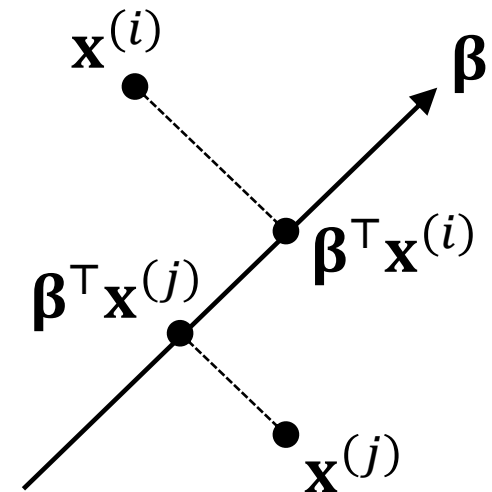


ランキングの確率モデル： 一対比較のモデル

- 感性評価などは絶対評価を与えにくい
- 一対比較：「どちらがよいか」のほうが答えやすい
- データ*i*がデータ*j*よりも上位である（ $i > j$ ）確率：

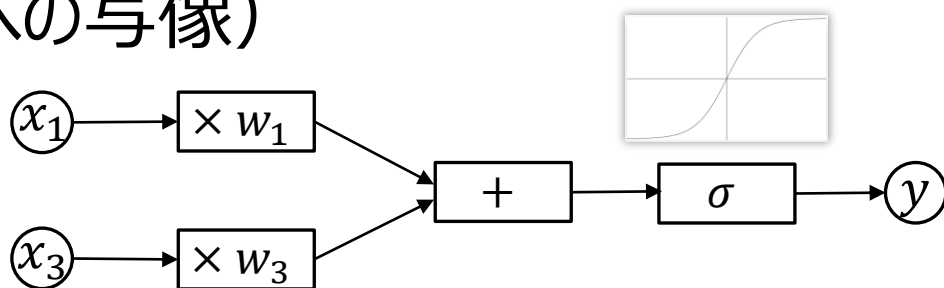
$$P(i > j | \mathbf{x}^{(i)}, \mathbf{x}^{(j)}, \boldsymbol{\beta}) = \frac{\exp(\boldsymbol{\beta}^\top \mathbf{x}^{(i)})}{\exp(\boldsymbol{\beta}^\top \mathbf{x}^{(i)}) + \exp(\boldsymbol{\beta}^\top \mathbf{x}^{(j)})}$$
$$= \sigma(-\boldsymbol{\beta}^\top (\mathbf{x}^{(i)} - \mathbf{x}^{(j)}))$$

- 2つのデータの独立変数の差 $\mathbf{x}^{(i)} - \mathbf{x}^{(j)}$ によって、ペアに対する独立変数としたロジスティック回帰

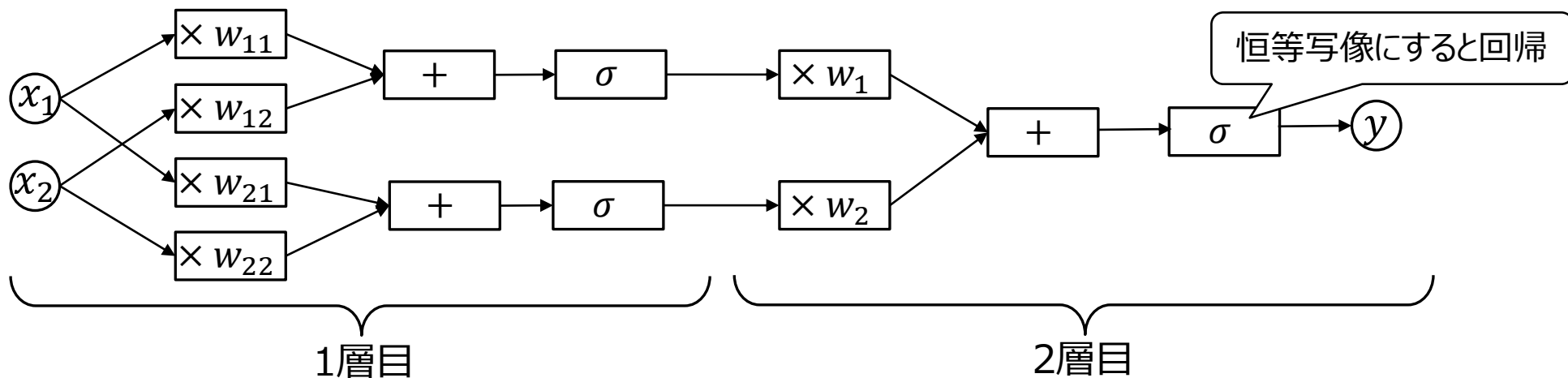


ニューラルネットワーク： ロジスティック回帰の多層化

- ロジスティック回帰は線形回帰モデルの出力に非線形写像を適用
($(0,1)$ 区間への写像)



- ニューラルネットワークはこれを多層化し非線形性を導入したもの
 - 非線形写像は必ずしもロジスティック関数である必要はない



まとめ：

ロジスティック回帰

■ ロジスティック回帰：

- ダミー変数 $y \in \{+1, -1\}$ を従属変数として、 $y = +1$ である確率を出力するモデル
- 最尤推定の対数尤度は、大域解をもつが、解析解をもたない
- 非線形最適化法によって、最適解を求める：
ニュートン法、再急降下法、確率的勾配法、...

■ ロジスティック回帰モデルの発展：

多クラスロジスティック回帰モデル、順序回帰モデル、
ランキング（一対比較）モデル、ニューラルネットワーク