https://goo.gl/tYmyZL

Kyoto University

統計的モデリング基礎⑤
～最尤推定（続き）～

鹿島久嗣
（情報学科 計算機科学コース）

Department of Intelligence Science and Technology

中間テスト：
6/7(水) 4限 総合研究8号館NSホール

- 講義内で行います
- 範囲は第１～７回
- 持ち込み等なし

マーケティング分野への応用を対象とした参考書

マーケティングの統計モデル

出版社：朝倉出版
発刊年月： 2015.8
ISBN ： 4254128533
A5判；192ページ

マーケティングを題材としながら、基本的な統計的モデリングの方法が学べる

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 $x_1, x_2, \ldots, x_n$ から確率モデル$f(x \mid \theta)$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）：

$$L(\theta) = \prod_{i=1}^{n} f(x_i \mid \theta)$$

実際には対数尤度で扱うことが多い

- 尤度最大になるパラメータを推定値$\hat{\theta}$とする

$$\hat{\theta} = \mathrm{argmax}_\theta \prod_{i=1}^{n} f(x_i \mid \theta) = \mathrm{argmax}_\theta \sum_{i=1}^{n} \log f(x_i \mid \theta)$$

– もっともデータを生成する確率が高い（「最も尤もらしい」）

線形回帰モデルの最尤推定：
線形回帰の確率モデル

- データ：$\mathbf{x} = (x^{(i)}, x^{(2)}, \dots, x^{(n)})$ と $\mathbf{y} = \left(y^{(1)}, y^{(2)}, \dots, y^{(n)}\right)$ に線形モデル： $g(x) = \beta x + \alpha$ を当てはめる
- 最小二乗法：$\ell(\alpha, \beta) = \sum_{i=1}^{n} \left(y^{(i)} - \left(\beta x^{(i)} + \alpha\right)\right)^2$ を最小化
- 一方、線形回帰モデルに対応する確率モデルを考えると：
  - 正規分布：$y^{(i)}$は平均$\beta x^{(i)} + \alpha$ 、分散σ^2の正規分布に従う
  - 確率密度：$f\left(y^{(i)} \mid x^{(i)}\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - (\beta x^{(i)} + \alpha)\right)^2}{2\sigma^2}\right)$
  - 「平均的に」回帰直線 $y = \beta x + \alpha$ に乗るデータを生成できる

線形回帰モデルの最尤推定：
線形回帰の確率モデルの最尤推定 = 最小二乗法

- 線形回帰モデルに対応する確率モデルを考える：
- 確率密度関数：$f\left(y^{(i)}\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - (\beta x^{(i)} + \alpha)\right)^2}{2\sigma^2}\right)$
- 対数尤度：$L(\alpha, \beta) = \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}\right)$

$$= -\frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(y^{(i)} - (\beta x^{(i)} + \alpha)\right)^2 + \text{const.}$$

- 対数尤度をα, βについて最大化すること（最尤推定）
＝二乗誤差をα, βについて最小化すること（最小二乗法）

線形回帰モデルの最尤推定：分散の最尤推定量

- 確率密度関数：$f\left(y^{(i)}\right) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(y^{(i)}-(\beta x^{(i)}+\alpha)\right)^2}{2\sigma^2}\right)$

- 分散については、対数尤度：

$$L(\sigma^2) = \frac{n}{2}\log\frac{1}{\sigma^2} - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y^{(i)} - (\beta x^{(i)} + \alpha)\right)^2 + \text{const.}$$

- $L(\sigma^2)$を最大化する最尤推定量は：

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - (\beta x^{(i)} + \alpha)\right)^2$$

※ 以上の議論は重回帰モデルの場合も同様

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（いちおう解析的な解）
  - 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

最尤推定量の性質：一致性

- モデルパラメータθの推定量として$\hat{\theta}$を得た（例えば最尤推定で）
- 推定量の良さはどのように評価するか
  - 不偏性 $E_\theta[\hat{\theta}] = \theta$：推定量の期待値が真の値に一致する
    - 分散の最尤推定量は不偏性をもたない
  - 一致性：標本サイズを大きくしていくと真の値に一致する：

$$\hat{\theta} \xrightarrow[n\to\infty]{} \theta$$

- 最尤推定は、適当な条件のもと一致性をもつ

漸近正規性：最尤推定は漸近正規性をもつ

- 最尤推定量の分布は$n \to \infty$で、真のパラメータθを平均とする正規分布に従う
- もう少し厳密にいうと：
  $\sqrt{n}(\hat{\theta} - \theta)$の分布が平均$\theta$ 、分散$I(\theta)^{-1}$の正規分布に近づく
  - $I(\theta)$はフィッシャー情報量：

$$I(\theta) = -E_\theta\left[\frac{\partial^2}{\partial\theta^2}\log f(x|\theta)\right]$$
$$= -\int\left(\frac{\partial^2}{\partial\theta^2}\log f(x|\theta)\right) f(x|\theta)\mathrm{d}x$$

  - $n \to \infty$で$\hat{\theta} \to \theta$

ポアソン回帰：
非負整数の回帰モデル

- 例えば、ある機械の各日の故障件数をモデル化したい
  - 曜日や気温などに依存して平均的な故障件数が変わるとする
- 独立変数に依存する回数のモデル：ポアソン回帰

$$P(Y = k \mid \mathbf{x}, \boldsymbol{\beta}) = \frac{(\boldsymbol{\beta}^\top \mathbf{x})^k}{k!} \exp(-\boldsymbol{\beta}^\top \mathbf{x})$$

  - ポアソン分布の平均が線形モデルで表される
    - ポアソン分布：$P(Y = k \mid \lambda) = \frac{\lambda^k}{k!} \exp(-\lambda)$
    - 重回帰モデル：$\lambda = \boldsymbol{\beta}^\top \mathbf{x}$

    組み合わせる

ポアソン回帰の最尤推定：
解析解は得られなさそう

- 独立変数：$\left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\right)$ # n日分の測定
- 従属変数：$\left(y^{(1)}, y^{(2)}, \dots, y^{(n)}\right)$ # n日分の故障数
- 対数尤度：

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

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

- これを最大化する$\boldsymbol{\beta}$を求めたいが、解析解は得られない

判別問題：
ダミー変数を従属変数とする

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

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）

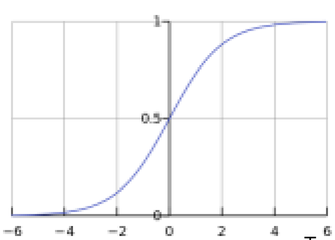
ロジスティック回帰：
ダミー変数を従属変数とするモデル

- 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 → もっとちゃんと扱いたい
  - 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：

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

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

$\sigma(\ \ )$



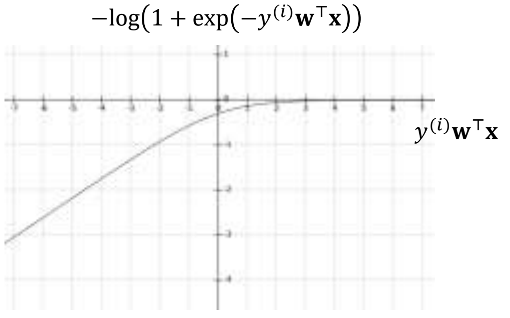
$\mathbf{w}^\top \mathbf{x}$

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得られるが解析解はない

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

$$\left(= \sum_{i=1}^{n} \delta\left(y^{(i)}=1\right)\log\frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x}^{(i)})} + \delta\left(y^{(i)}=-1\right)\log\left(1-\frac{1}{1+\exp(-\mathbf{w}^\top\mathbf{x}^{(i)})}\right)\right)$$

- $L(\mathbf{w})$は凸関数：
  - 大局解が得られる
  - 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

- 最尤推定の目的関数（最大化）：

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

  - 解析解は得られないが、凸関数（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：$\mathbf{w}^{\mathrm{NEW}} \leftarrow \mathbf{w} + \mathbf{d}$

パラメータ更新：目的関数をもっとも改善する更新を行う

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

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

- $L_{\mathbf{w}}(\mathbf{d})$を最大化する更新分 $\mathbf{d}^{*}$ を見つけよ：
  - $\mathbf{d}^{*} = \mathrm{argmin}_{\mathbf{d}}\, L_{\mathbf{w}}(\mathbf{d})$

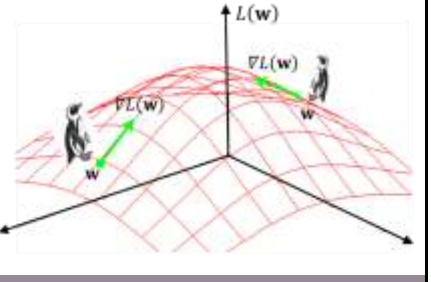
最良のパラメータ更新：目的関数をテイラー展開で近似

- 目的関数のテイラー展開：

3次以上の項

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

  - 勾配：$\nabla L(\mathbf{w}) = \left(\frac{\partial L(\mathbf{w})}{\partial w_1}, \frac{\partial L(\mathbf{w})}{\partial w_2}, \dots, \frac{\partial L(\mathbf{w})}{\partial w_D}\right)^{\top}$
    - 目的関数が最も急な方向
  - ヘッセ行列：$[H(\mathbf{w})]_{i,j} = \frac{\partial^2 L(\mathbf{w})}{\partial w_i \partial w_j}$



ニュートン法：
2次近似を最適化する

- テイラー展開で 3 次以降の項を無視する：

3次以上の項

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

- 最大化するために$\mathbf{d}$で微分：$\frac{\partial L_{\mathbf{w}}(\mathbf{d})}{\partial \mathbf{d}} \approx \nabla L(\mathbf{w}) + \boldsymbol{H}(\mathbf{w})\mathbf{d}$
- これを$= \mathbf{0}$とおいて解くと： $\mathbf{d} = -\boldsymbol{H}(\mathbf{w})^{-1}\nabla L(\mathbf{w})$

実際には連立
方程式を解く

- ニュートン法：

$$\mathbf{w}^{\mathrm{NEW}} \leftarrow \mathbf{w} - \boldsymbol{H}(\mathbf{w})^{-1}\nabla L(\mathbf{w})$$

$$\mathbf{w} \xrightarrow{-\boldsymbol{H}(\mathbf{w})^{-1}\nabla L(\mathbf{w})} \mathbf{w} - \boldsymbol{H}(\mathbf{w})^{-1}\nabla L(\mathbf{w})$$

線形探索付きニュートン法：
近似は必ずしも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 $\mathbf{w}^{\mathrm{NEW}} \leftarrow \mathbf{w} - \boldsymbol{H}(\mathbf{w})^{-1}\nabla L(\mathbf{w})$ は 2 次近似が正しいことを仮定している：

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

  – 実際には正しくない

- そこで更新の向き $\boldsymbol{H}(\mathbf{w})^{-1}\nabla L(\mathbf{w})$ のみを採用して：
  $\mathbf{w}^{\mathrm{NEW}} \leftarrow \mathbf{w} - \eta \boldsymbol{H}(\mathbf{w})^{-1}\nabla L(\mathbf{w})$
- 学習率 $\eta > 0$ の決定法：
  – 適当にステップ数とともに適当に減衰

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

  – 線形探索：$\eta^* = \mathrm{argmax}_\eta\, L\left(\mathbf{w} - \eta \boldsymbol{H}(\mathbf{w})^{-1}\nabla L(\mathbf{w})\right)$

最急勾配法：
ヘッセ行列を使わずシンプルで軽い更新

- 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：$\mathbf{w}^{\mathrm{NEW}} \leftarrow \mathbf{w} - \eta \boldsymbol{H}(\mathbf{w})^{-1} \nabla L(\mathbf{w})$
- 最急勾配法：
  - ヘッセ行列の逆行列 $\boldsymbol{H}(\mathbf{w})^{-1}$ を単位行列 $\boldsymbol{I}$ で置き換え：

$$\mathbf{w}^{\mathrm{NEW}} \leftarrow \mathbf{w} - \eta \nabla L(\mathbf{w})$$

勾配

- $\nabla L(\mathbf{w})$ は最も急な（目的関数が最も変化する）向き
- 学習率 η は線形探索で求める：

$$\mathbf{w} \xrightarrow{-\eta \nabla L(\mathbf{w})} \mathbf{w} - \eta \nabla L(\mathbf{w})$$

ロジスティック回帰の場合の勾配：
比較的簡単に計算可能

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

- $$\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = -\sum_{i=1}^{n} \frac{1}{1+\exp\left(-y^{(i)}\mathbf{w}^\top \mathbf{x}^{(i)}\right)} \frac{\partial\left(1+\exp\left(-y^{(i)}\mathbf{w}^\top \mathbf{x}^{(i)}\right)\right)}{\partial \mathbf{w}}$$

$$= \sum_{i=1}^{n} \frac{1}{1 + \exp(-y^{(i)}\mathbf{w}^\top \mathbf{x}^{(i)})} \exp\left(-y^{(i)}\mathbf{w}^\top \mathbf{x}^{(i)}\right) y^{(i)} \mathbf{x}^{(i)}$$

$$= \sum_{i=1}^{n} \left(1 - f(y^{(i)} | \mathbf{x}^{(i)}, \mathbf{w})\right) y^{(i)} \mathbf{x}^{(i)}$$

現在のパラメータでのモデルが与える確率

確率的最適化とミニバッチ：データの部分集合を用いた効率的な推定

- 対数尤度は各データの対数尤度の和：$L(\mathbf{w}) = \sum_{i=1}^{n} \ell^{(i)}$
- 勾配 $\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = \sum_{i=1}^{n} \frac{\partial \ell^{(i)}}{\partial \mathbf{w}}$ の計算は $O(n)$ かかる
- 勾配をデータ1個で近似：$\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} \approx n \frac{\partial \ell^{(i)}}{\partial \mathbf{w}}$
  - 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$\ell^{(i)}$を使う）
- ミニバッチ：$1 < m < n$ 個のデータで勾配を近似：

$$\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} \approx \frac{n}{m} \sum_{j \in \mathrm{MiniBatch}} \frac{\partial \ell^{(i)}}{\partial \mathbf{w}}$$

練習：ポアソン回帰の最尤推定

- ポアソン回帰の最尤推定
  - 対数尤度：

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{N} y^{(i)} \log\left(\boldsymbol{\beta}^\top \mathbf{x}^{(i)}\right) - \sum_{i=1}^{N} \boldsymbol{\beta}^\top \mathbf{x}^{(i)} + \mathrm{const.}$$

  - 解析解は求まらない
- 最急勾配法の更新式を求めてみる