THE UNIVERSITY OF TOKYO

数理情報工学特論第一
【機械学習とデータマイニング】
2章：回帰（２）

かしま　ひさし
鹿島 久嗣
（数理６研）

kashima@mist.i.～

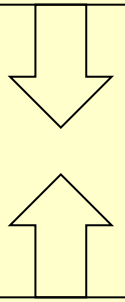
東京大学
THE UNIVERSITY OF TOKYO

DEPARTMENT OF MATHEMATICAL INFORMATICS

教師つき学習の王道である「回帰」について学びます

- 回帰問題の定義
- 線形回帰問題の定式化
- 線形回帰問題の初等的解法
- リッジ回帰：L_2正則化による過学習の回避

- 交差確認によるハイパーパラメータの決定
  - Leave-one-out交差確認
- 回帰問題の確率モデル的解釈



- 回帰の応用
- カーネル回帰
- L_1正則化

交差確認法によるハイパーパラメータの決定

正則化で出てくるハイパーパラメータは、モデルの真の性能を比較することで選びます

- 正則化によって汎化能力の現象に対処できることは分かったが、これによって、ユーザが指定する余分なパラメータ（ハイパーパラメータ）λ が出てきてしまう

$$L(\mathbf{w}) = \| \mathbf{\Phi w} - \mathbf{y} \|_2^2 + \lambda \| \mathbf{w} \|_2^2$$

- 訓練データを使った目的関数をハイパーパラメータλについて直接最適化すると、単にλ=0（正則化しないのが最良）となってしまう
- モデルの真の性能（の見積もり）によってλを決定する
- 真の性能の見積もりに最も良く用いられているのが**交差確認（クロスバリデーション）**である

交差確認では、モデルの真の性能を計ることができます

- 交差確認では、訓練データ集合をモデル推定用データ集合と、モデル検証用データ集合に分割する。
  - モデル推定は、前者のモデル推定用データ集合のみを用いて行う
  - 得られたモデルを、モデル検証用データ集合に適用し（予測値を計算し、損失を計算する）モデルの実際の性能を測る
- *K*-分割交差確認 (*K*-fold cross validation)
  - 訓練データ集合を*K*個に分割し、そのうち*K*-1個の塊をモデル推定に、残り1つの塊を検証に用いる
  - *K* 回繰り返して、平均の性能を得る
- Leave-one-out 交差確認（LOOCV）
  - 訓練データ集合のうち1つだけを検証用に残し、残りの*N*-1個の訓練データをモデル推定に用いる

ハイパーパラメータは交差確認によって決定できます

- ハイパーパラメータの決定には、交差確認を、ハイパーパラメータを変えながら繰り返す
- 例えば、線形回帰の例では$\lambda=0$の場合、$\lambda=0.1$の場合…などと、いくつかのλについて、それぞれK-分割交差確認を行い、もっとも性能のよいλを採用する
- 結構な計算量になる

線形回帰の場合には、leave-one-out交差確認(LOOCV)を非常に効率よく実行できます

- LOOCVの2乗誤差見積もりは

$$LOOCV \equiv \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - \tilde{y}^{(i)}\right)^2$$

  - ここで、$\tilde{y}^{(i)}$は、i番目の訓練データを除いた訓練データで推定したパラメータ$\mathbf{w}^{(-i)}$による、i番目の訓練データに対する出力の予測値

$$\tilde{y}^{(i)} \equiv \mathbf{w}^{(-i)\top}\boldsymbol{\phi}(x^{(i)})$$

- 問題は、$\mathbf{w}^{(-i)}$は全てのiに対して異なるので、leave-one-out交差確認(LOOCV)を行うには、逆行列計算をN回行う必要がある点
- 実は、全ての訓練データを用いたときのパラメータ推定値$\mathbf{w}$をもとに N 個の$\mathbf{w}^{(-i)}$を（逆行列の計算なしに）直接計算できる

LOOCVをナイーブに計算すると かなりの計算量が必要となってしまいます

- 訓練データ集合を全て用いて最尤推定されたパラメータは $\mathbf{w} = (\mathbf{\Phi}^\top \mathbf{\Phi})^{-1} \mathbf{\Phi}^\top \mathbf{y}$ と書くことができた。
  - これは計算できたものとする。
- ここで我々が欲しいのは全ての $i = 1,\ldots,N$ に対する $\mathbf{w}^{(-i)}$

$$\mathbf{w}^{(-i)} = (\mathbf{\Phi}^{(-i)\top} \mathbf{\Phi}^{(-i)})^{-1} \mathbf{\Phi}^{(-i)\top} \mathbf{y}^{(-i)}$$

  - $\mathbf{\Phi}^{(-i)}$は、$\mathbf{\Phi}$の i 行目を抜いたもの

$$\mathbf{\Phi}^{(-i)} \equiv \left(\boldsymbol{\phi}(x^{(1)}), \boldsymbol{\phi}(x^{(2)}), \ldots, \boldsymbol{\phi}(x^{(i-1)}), \boldsymbol{\phi}(x^{(i+1)}), \ldots, \boldsymbol{\phi}(x^{(N)})\right)^\top$$

  - $\mathbf{y}^{(-i)}$は$\mathbf{y}$のi番目の要素を抜いたもの

$$\mathbf{y}^{(-i)} \equiv \left(y^{(1)}, \ldots, y^{(i-1)}, y^{(i+1)}, \ldots, y^{(N)}\right)^\top$$

- それぞれの逆行列計算がO(D^3)くらいかかる

シャーマン-モリソンの公式(Sherman-Morrison's formula)が $\mathbf{w}^{(-i)}$の効率的な計算の実現に重要な役割を果たします

- シャーマン-モリソンの公式(Sherman-Morrison's formula)：

$$\left(\mathbf{A}+\mathbf{u}\mathbf{v}^{\top}\right)^{-1}=\mathbf{A}^{-1}-\frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^{\top}\mathbf{A}^{-1}}{1+\mathbf{v}^{\top}\mathbf{A}^{-1}\mathbf{u}}$$

  - **A**は正方行列、**u**と**v**
- この公式のポイントは：
  - **A**にランク1の行列$\mathbf{u}\,\mathbf{v}^{\top}$を加えた行列の逆行列が$\mathbf{A}^{-1}$から計算できる
  - 一旦、$\mathbf{A}^{-1}$を計算しておけば、$\mathbf{A}+\mathbf{u}\,\mathbf{v}^{\top}$の逆行列は、逆行列よりもずっと少ない計算量で計算することができる。

シャーマン-モリソンの公式を使うことで、LOOCVを高速化できます

- 問題となっている$\boldsymbol{\Phi}^{(-i)\top}\boldsymbol{\Phi}^{(-i)}$を以下のように書き換える

$$(\boldsymbol{\Phi}^{(-i)^\top}\boldsymbol{\Phi}^{(-i)})^{-1} = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \boldsymbol{\phi}(x^{(i)})\boldsymbol{\phi}^\top(x^{(i)}))^{-1}$$

- この形はまさに、既に逆行列を計算した$\boldsymbol{\Phi}^\top\boldsymbol{\Phi}$に、ランク1の行列 $\boldsymbol{\phi}(\mathbf{x}^{(i)})\,\boldsymbol{\phi}(\mathbf{x}^{(i)})^\top$を加えたものの逆行列になっている。
- ここで、シャーマン-モリソンの公式の公式を使えば、

$$(\boldsymbol{\Phi}^{(-i)^\top}\boldsymbol{\Phi}^{(-i)})^{-1} = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1} - \frac{(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\phi}(x^{(i)})\boldsymbol{\phi}^\top(x^{(i)})(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}}{1 + \boldsymbol{\phi}^\top(x^{(i)})(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\phi}(x^{(i)})}$$

  - シャーマン-モリソンの公式において$\mathbf{A} \equiv \boldsymbol{\Phi}^\top\boldsymbol{\Phi}$、$\mathbf{u} \equiv \mathbf{v} \equiv \boldsymbol{\phi}(\mathbf{x}^{(i)})$ とする
- 結果、$\mathbf{w}^{(-i)}$を$\mathbf{w}$から逆行列なしで計算することができ、これをもとにi番目の訓練データに対する出力の予測値を予測値を計算できる。
- （L_2）正則化を行っている場合には、$\mathbf{A} \equiv \boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \lambda\,\mathbf{I}$

シャーマン-モリソンの公式を使わない（もうちょっと真面目な）導出

- ポイントは等式：

$$\begin{aligned}\mathbf{w}^{(-i)} &= (\mathbf{\Phi}^{(-i)\top}\mathbf{\Phi}^{(-i)})^{-1}\mathbf{\Phi}^{(-i)\top}\mathbf{y}^{(-i)} \\ &= (\mathbf{\Phi}^{\top}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\top}\tilde{\mathbf{y}}^{(-i)}\end{aligned}$$

  - $\tilde{\mathbf{y}}^{(-i)}$ は、$\mathbf{y}$の i 番目の要素を $\tilde{y}^{(-i)}$ で置き換えたベクトル、

$$\tilde{\mathbf{y}}^{(-i)} \equiv (y^{(1)}, \ldots, y^{(i-1)}, \tilde{y}^{(i)}, y^{(i+1)}, \ldots, y^{(N)})^{\top}$$

- 「i番目の訓練データを除いたときの推定パラメータ$\mathbf{w}^{(-i)}$は、$\mathbf{w}^{(-i)}$をつかってi番目の訓練データの予測値を書き換えたデータを使った推定パラメータと等しい」
  - ある損失関数を最小化するパラメータで予測した出力値をもつデータを訓練データセットに加えても、これらについては初めからモデルの予測値と訓練データの出力値が一致しているため、損失関数は増加しない

別の導出

- 式 $\mathbf{w}^{(-i)} = (\mathbf{\Phi}^\top \mathbf{\Phi})^{-1} \mathbf{\Phi}^\top \tilde{\mathbf{y}}^{(-i)}$ を用いれば、
- i 番目の訓練データに対するleave-one-outの予測値 $\tilde{y}^{(-i)}$ は

$$\tilde{y}^{(i)} = \phi^\top(x^{(i)})(\mathbf{\Phi}^\top \mathbf{\Phi})^{-1} \mathbf{\Phi}^\top \tilde{\mathbf{y}}^{(-i)}$$

- 簡単のため $\mathbf{h}(x^{(i)}) \equiv \left(\phi^\top(x^{(i)})(\mathbf{\Phi}^\top \mathbf{\Phi})^{-1} \mathbf{\Phi}^\top\right)^\top$ とおくと

$$\tilde{y}^{(i)} = \mathbf{h}(x^{(i)})^\top \tilde{\mathbf{y}}^{(-i)}$$

- 右辺を展開すると

$$\tilde{y}^{(i)} = \sum_{j=1,\ldots,i-1,i+1,\ldots,N} h_j(x^{(i)}) y^{(j)} + h_i(x^{(i)}) \tilde{y}^{(i)}$$

⇩

$$\tilde{y}^{(i)} = \frac{\sum_{j=1,\ldots,i-1,i+1,\ldots,N} h_j(x^{(i)}) y^{(j)}}{1 - h_i(x^{(i)})}$$

回帰の確率モデル的解釈

回帰の確率モデル的解釈：「損失の最小化」は最尤推定として解釈できます

- 「教師つき学習とは条件付き確率$P(y|x)$を推定する問題である」とひとまず定義し、**最尤推定**がその推定手段であると述べたが、ここまでの議論にはそのような確率モデル的な観点は現れていない
- ここでは、実はここまでに述べたことは、条件付き確率 $P(y|x)$ の最尤推定と等価である
- 最尤推定は、訓練データ集合$\{ (x^{(i)},y^{(i)})\}_{i=1}^{N}$に対して、対数尤度の和

$$L(\mathbf{w}) \equiv \sum_{i=1}^{N} \log P(y^{(i)}|x^{(i)};\mathbf{w})$$

を最大化するようなパラメータを、推定パラメータ$\mathbf{w}^*$とする考え方

$$\mathbf{w}^* \equiv \operatorname*{argmax}_{\mathbf{w}} L(\mathbf{w})$$

線形回帰モデルの確率モデル的解釈：出力にガウス的ノイズが載るものとします

- 回帰問題に対応する条件付き確率$P(y|x)$を定義する
- 出力yが、平均が$f(x; \mathbf{w})=\mathbf{w}^\top \phi(x)$で分散が$\sigma^2$であるような正規分布に従って発生するものと仮定する。

$$P(y|x;\mathbf{w}) \equiv \mathcal{N}(f(x;\mathbf{w}),\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y-f(x;\mathbf{w}))^2}{2\sigma^2}\right)$$

  - なお、$\mathcal{N}(\mu, \sigma^2)$ は平均μ、分散σ^2をもつ一次元の正規分布

$$y \sim \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$$

線形回帰に対する対数尤度の最大化は2乗損失の最小化と一致します

- 回帰問題の場合に対数尤度の和を計算すると

$$L(\mathbf{w}) = \sum_{i=1}^{N} \log\left(\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y - f(x;\mathbf{w}))^2}{2\sigma^2}\right)\right)$$
$$= -N\log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y - f(x;\mathbf{w}))^2$$

- 1項目が$\mathbf{w}$に依存しないことに注意すると、最尤推定の解は

$$\mathbf{w}^* \equiv \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y - f(x;\mathbf{w}))^2$$

となり、まさに2乗損失を損失関数とした線形回帰に一致する

リッジ回帰は、ベイズ統計的な解釈ができます

- 正則化によって、目的関数の2乗損失にパラメータのノルムを加えることで、過学習を緩和できる
- 目的関数に2乗損失を用いることが最尤推定に対応しているならばこれにパラメータのノルムを加えたものは何に対応しているだろうか？
- 正則化の枠組みは、ベイズ統計の立場から解釈できる

ベイズ統計では「パラメータの上での確率分布」を考えます

- これまで、パラメータ$\mathbf{w}$には「真の値」があるとしており、これをデータから計り知るための方法が最尤推定などであった
- ベイズ統計では、パラメータは一意に決まっているようなものではなく、何らかの確率分布に従って発生するもの、もしくは、パラメータの分布自体がパラメータの性質や意味を示していると考える
- ベイズ統計で重要な役割を果たすのが、**事後分布**と呼ばれる$P(\mathbf{w}|\mathcal{D})$
  - 訓練データ集合$\mathcal{D}$が与えられた時のパラメータの上での確率分布
  - 事後分布$P(\mathbf{w}|\mathcal{D})$は訓練データ集合$\mathcal{D}$を「見た後」に、どういった$\mathbf{w}$が確からしいかを表す

ベイズ統計における「学習」は、訓練データを見る前の事前分布から、観た後の事後分布への変化に対応します

- 事前分布と事後分布
  - 事前分布 $P(\mathbf{w})$ ：訓練データ集合$\mathcal{D}$を**見る前**のパラメータ上の分布
  - 事後分布 $P(\mathbf{w}|\mathcal{D})$：訓練データ集合$\mathcal{D}$を**見た後**のパラメータ上の分布
- 事前分布は、そもそもどのあたりのパラメータがそれらしいかを表す事前知識
- 訓練データ集合を与えられることによって、パラメータの上での確率分布が、事前分布$P(\mathbf{w})$から、事後分布$P(\mathbf{w}|\mathcal{D})$に変化する
  - これが、ベイズ統計における「学習」である
  - このあたりが、ベイズ統計に初めて触れる際、違和感を感じる部分

実は、正則化は事後確率最大化に対応します

- ベイズ統計ではパラメータは点ではなく分布で与えられるが、正則化の枠組みでは、確かに何か一つにパラメータが決まった
- 実は、正則化は、事後分布を最大化するようなパラメータが最良であるとする**事後確率最大化**(MAP; Maximum A Posteriori)}という考え方に従ってパラメータを決定していることと等価である

事後確率最大化の目的関数は、最尤推定の目的関数＋事前分布による項 で補正したものと解釈できます

- ベイズの公式によって$P(\mathbf{w}|\mathcal{D})$を書き換えてみる

$$P(\mathbf{w}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathbf{w})P(\mathbf{w})}{P(\mathcal{D})}$$

- 形式的に、対数を取ると

$$\log P(\mathbf{w}|\mathcal{D}) = \log P(\mathcal{D}|\mathbf{w}) + \log P(\mathbf{w}) - \log P(\mathcal{D})$$

- 事後確率最大化 $\mathbf{w}^* \equiv \underset{\mathbf{w}}{\operatorname{argmax}} \log(\mathbf{w}|\mathcal{D})$ は、

$$\mathbf{w}^* \equiv \underset{\mathbf{w}}{\operatorname{argmax}} \log P(\mathcal{D}|\mathbf{w}) + \log P(\mathbf{w})$$

  - 1項目は対数尤度
  - 2項目は事前分布の対数を取ったもの
  - 最尤推定の目的関数に、事前分布で補正をかけている

- 最尤推定で求まるパラメータを、事前分布最大となるパラメータに「少し引き戻す」というイメージ

リッジ回帰の事後確率最大化としての解釈

- 事前分布$P(\mathbf{w})$を各次元w_dが平均$\mathbf{0}$、分散η^2の正規分布とする

$$P(w_d) = \frac{1}{\sqrt{2\pi}\eta} \exp\left(-\frac{w_d^2}{2\eta^2}\right)$$

  - パラメータ$\mathbf{w}$はなるべく$\mathbf{0}$に近いものがよいとする正則化の気持ち

- 事前分布の対数を取ると、$\log P(w_d) = -\log\sqrt{2\pi}\eta - \frac{1}{2\eta^2} w_d^2$ より

$$\log P(\mathbf{w}) = -D \log\sqrt{2\pi}\eta - \frac{1}{2\eta^2} \| \mathbf{w} \|_2^2$$

  - 2項目に正則化項と同じ、パラメータの2-ノルムが現れる
  - 1項目は$\mathbf{w}$を含まないため無視できる

- 事後確率最大化がリッジ回帰に一致する（$\lambda \equiv \sigma^2 / \eta^2$）

$$\mathbf{w}^* \equiv \arg\max_{\mathbf{w}} \log P(\mathcal{D}|\mathbf{w}) + \log P(\mathbf{w})$$

$$= \arg\min_{\mathbf{w}} \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y - f(x; \mathbf{w}))^2 + \frac{1}{2\eta^2} \| \mathbf{w} \|_2^2$$