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（１）

かしま　ひさし
鹿島 久嗣
（数理６研）

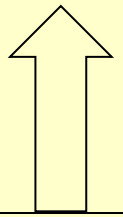
kashima@mist.i.～

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

回帰問題の定義

回帰は、実数値を予測する教師つき学習

- 回帰は教師つき学習問題の一種
  - 機械学習の問題の中で極めて王道、適用範囲も広い
- 目的は、入力 $x \in \mathcal{X}$ に対し、実数値 $y \in \mathcal{R}$を返すような関数 $f : \mathcal{X} \to \mathcal{R}$を得ることである。
  - たとえば x はある家、y はその家の価格を表す
  - $\mathcal{X}$ はこの世に存在しうる限りの家の集合、$\mathcal{R}$ は実数
- しかし、何の手がかりもなしに f を得ることは難しい。
- ある家Aの価格が2,000万円、家Bの価格が4,000万円、…といったように正解がいくつか与えられていれば、これらをもとに、家とその価格との関係について何らかの法則を見つけることができるかもしれない
  - ベンチマークデータ： UCI Machine Learning Repository / housing

「条件付き分布の推定」という本来の目的は一旦忘れます

- なお、前回までの文脈に則していえば、xが与えられたときのyの条件付き確率分布 $P(y|x)$ を得ること
  - これがわかれば、f は与えられた入力x に対して確率が最大になる出力を返す関数として得られる

$$f(x) = \mathrm{argmax}_y \ P(y|x)$$

  - とりあえずは $P(y|x)$ではなく$f(x)$を直接得るのが目的とする

※ むしろ、本当は「教師つき＝条件付き分布の推定」と限定すること自体が適当ではないのだが、前回からの話の流れ上

回帰では、入出力ペアの集合（訓練データ集合）を一般化することで、出力未知のデータへの対応を目指します

- N 軒の家と、それぞれの価格が分かっているものとする
- N 個の入力と出力の組 $(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(N)}, y^{(N)})$ が与えられたとする
  - 各々の入出力の組を**訓練データ**と呼ぶ
  - これらをまとめて**訓練データ集合**と呼ぶ
- 目的は、訓練データ集合をもとに関数 $f: \mathcal{X} \to \mathcal{R}$ を推定すること
- 単に、訓練データの入出力を再現するだけならば、訓練データを丸覚えしてしまえばよい → これでは「学習」とはいえない
- 本当に実現したいのは、訓練データに入っていない $\boldsymbol{x}$ に対しても正しく出力を予測できるような f を得ること
- **汎化**：訓練データを丸覚えするのではなく、これを一般化することによって、未知の $\boldsymbol{x}$ に対しても対応できるようにする

入力を D 次元の特徴ベクトルとして表現します

- 回帰における出力yは実数値であるが、一方入力$\boldsymbol{x}$は何かの集合（例えば家の集合）の要素であるということしか言っていない
- $\boldsymbol{x}$を扱いやすくするために、$\boldsymbol{x}$ をD次元の実数値の集まりとして表現する
  - 関数 $\boldsymbol{\phi}:\mathcal{X} \to \mathcal{R}^D$を考える
  - 関数$\boldsymbol{\phi}$は入力xから**特徴**を抽出する、特徴抽出器のようなもの
- 特徴ベクトル $\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}),\ldots,\phi_D(\boldsymbol{x}))^\top$：$x$を$D$次元の実数値ベクトルとして表現
- 家$\boldsymbol{x}$ の価格を予測するのに有効そうな特徴は何だろうか？
  - 部屋数、駅までの距離、地域の犯罪発生率、…
  - たとえば、$\phi_1(\boldsymbol{x})$が部屋数、$\phi_2(\boldsymbol{x})$が駅までの距離

離散的な特徴は、便宜的に実数値化して使用します

- 離散的な特徴は「便宜的に」実数値であるとして扱う
- 性別のような2値的な特徴は、男ならば1、女ならば-1など
  - これを{0,1}で符号化すると、結果も異なる
- 都道府県などのように複数の可能性があるならば、
  東京都である(1)かない(-1)かといった特徴を複数個用意するなどで
  対応する

※ 旧来の（統計的でない）機械学習においては、
むしろ離散がデフォルト → 後に連続化 という道筋であった

- 統計的機械学習では、むしろ連続的なモデルがデフォルト

多くの機械学習問題における重要な仮定：データは独立であるとします

- ある家Aのデータが、2000年当時のものと、2010年当時のものの2つある場合を考えてみる
  - 2010年での価格は、2000年での価格に依存する
- 大抵の機械学習手法は、データはお互いに独立であることを仮定している
- つまり、特徴ベクトルとラベルの組(ϕ, y)の上の確率分布 $P(\phi, y)$ からそれぞれ互いに独立に生成(サンプリング)されているものとしている
- データの独立性を仮定しているので、同じ家から2回以上データを取得するのは厳密には若干問題がある
  - 「サンプリングの偏り（バイアス）」問題
- 当面のところ、独立性を仮定する

回帰問題の応用

- 回帰問題の応用例：
  - 価格予測：ある商品 x がいくら（y）で売れるか？
  - 需要予測：ある商品 x がどのくらい（y）需要があるか？
  - 売上予測：ある商品 x がどのくらい（y）売れるか？
  - 活性予測：ある化合物 x がどのくらい（y）活性をもつか？
- ほか、若干抽象度は異なるが（後で述べる）：
  - 時系列予測：ある過去の履歴 x が与えられたときの、次の時点での値 y はいくつか？
  - 分類問題：出力 y が実数値ではなく、離散値を取る場合
    - 分類問題に特化した手法は後の回で紹介する。

線形回帰問題の定式化

回帰問題を、線形回帰問題として定式化してみます

- 回帰問題を定式化するにあたり、定義しなければならないのは
  - モデル：どのような形式で x から y を予測するか？
  - 目的関数：訓練データをどのように用いるか？
- ここでは、最も標準的な定義を用いることにする
  - モデル：線形モデル（線形回帰）
  - 目的関数：2乗損失（2乗誤差）

モデルの定義：線形モデル を考えます

- モデルとしては単純な線形モデルを考えることにする

$$f(x; \mathbf{w}, b) \equiv \mathbf{w}^\top \boldsymbol{\phi}(x) + b = \sum_{d=1}^{D} w_d \phi_d(x) + b$$

  - 入力xが与えられたときに、出力の予測値として $f(x; \mathbf{w}, b)$ を返す
  - モデルのもつパラメータ（モデルを一意に決定するもの）は
    - ベクトル$\mathbf{w}$：特徴空間と同じD次元のベクトル $\mathbf{w} \equiv (w_1, w_2, \ldots, w_D)$ で、d次元目の値w_dは、d 番目の特徴 $\phi_d(x)$の出力への寄与を表す
    - スカラーb は入力に関わりなく出力を全体的に調整する項

※ のちに非線形な場合も考えるが、これらも基本的には線形モデルの枠組みで扱うことができる

表記を簡単にするためにバイアス項bを消します

- バイアス項 bを特別扱いするのは面倒なので特徴ベクトルに含めてしまう
- パラメータと特徴ベクトルを以下のように再定義する

$$\mathbf{w} \equiv (\mathbf{w}^\top, w_{D+1})^\top$$
$$\boldsymbol{\phi}(x) \equiv (\boldsymbol{\phi}(x)^\top, 1)^\top$$

- 前掲の線形モデルは、以下のようにバイアス項bを使わずに書くことができるので扱いやすい

$$f(x; \mathbf{w}) \equiv \mathbf{w}^\top \boldsymbol{\phi}(x) = \sum_{d=1}^{D} w_d \phi_d(x)$$

- パラメータ推定の方式によっては必ずしも等価ではなくなる（後で述べる「正則化」を行う場合）

目的関数の定義：2乗誤差を用います

- 与えられた訓練データ$\{ (\boldsymbol{x}^{(i)}, y^{(i)}) \}_{i=1}^{N}$から、パラメータwを決定するために、回帰を最適化問題として定式化する。
- 我々の持つ情報は訓練データであるから、
  - 与えられた訓練データ$\{ (\boldsymbol{x}^{(i)}, y^{(i)}) \}_{i=1}^{N}$のそれぞれに対して、
  - モデルの出力$f(\boldsymbol{x}^{(i)};\mathbf{w})$を、正しい出力$y^{(i)}$になるべく近づける
- 損失関数$l(f(\boldsymbol{x};\mathbf{w}), y)$：モデルの出力と正しい出力の「遠さ」を測る指標
- 具体的な例としては、2乗誤差（L_2損失）がよく用いられる

$$\ell(f(x;\mathbf{w}), y) \equiv (f(x;\mathbf{w}) - y)^2$$

  - 他、絶対誤差（L_1損失）やHuber損失などが、特にロバスト化（データの外れ値に強くする）の観点から用いられる

目的関数の定義：2乗誤差を用います

- 目的関数として、損失関数の和を考える

$$L(\mathbf{w}) \equiv \sum_{i=1}^{N} \ell(f(x^{(i)}; \mathbf{w}), y^{(i)})$$

- 損失関数を2乗損失とするならば、

$$L(\mathbf{w}) \equiv \sum_{i=1}^{N} \left(f(x^{(i)}; \mathbf{w}) - y^{(i)}\right)^2$$

- パラメータの推定値$\mathbf{w}$^は、この損失関数の和 $L(\mathbf{w})$を最小化するように決定される

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \; L(\mathbf{w})$$

線形回帰の初等的解法

2乗誤差を用いた線形回帰問題を解いてみます

- 2乗誤差の和は、線形モデルを仮定すると、

$$L(\mathbf{w}) = \sum_{i=1}^{N} \left(\mathbf{w}^\top \boldsymbol{\phi}(x^{(i)}) - y^{(i)} \right)^2$$

- これは以下のように書き換えることができる。

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

  - 訓練データの特徴ベクトル集合 $\{\boldsymbol{\phi}(x^{(i)})\}_{i=1}^{N}$をまとめて行列とし

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

    - この行列は**デザイン行列**と呼ばれる
  - 対応する出力集合$\{ y^{(i)} \}_{i=1}^{N}$をまとめて、ベクトルとして

$$\mathbf{y} \equiv \left(y^{(1)}, y^{(2)}, \ldots, y^{(N)} \right)^\top$$

  - $\| \cdot \|_2^2$ は2-ノルム $\| \mathbf{a} \|_2^2 \equiv \mathbf{a}^\top \mathbf{a}$

線形回帰問題の解は閉じた形で得られます

- 目的関数 $L(\mathbf{w}) = \| \mathbf{\Phi w} - \mathbf{y} \|_2^2$ を最小化する$\mathbf{w}$を求めるために

  $\mathbf{w}$で偏微分する $\dfrac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = 2\mathbf{\Phi}^\top (\mathbf{\Phi w} - \mathbf{y})$

- これを$\mathbf{0}$とおくことで $\mathbf{\Phi}^\top \mathbf{\Phi w} = \mathbf{\Phi}^\top \mathbf{y}$
- これを解くと、以下のように閉じた形で解が得られる

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

  - ここで、$\boldsymbol{\Phi}^\top\boldsymbol{\Phi}$は $D \times D$行列
  - 実際に解くには連立方程式を解くなどする
    - MATLABでは $\mathbf{w} = (\mathbf{\Phi}' * \mathbf{\Phi}) \backslash (\mathbf{\Phi}' * \mathbf{y})$

逆行列の解を安定させるための「正則化」

- 解 $\mathbf{w} = (\Phi^{\top}\Phi)^{-1}\Phi^{\top}\mathbf{y}$ が存在する条件は$\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi}$ が正則であること、すなわち、フルランクであること
  - 通常、訓練データ数Nは、特徴空間次元数 Dよりも大きいため、$\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi}$はフルランクとなることが多い
- そうでない場合には$\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi}$の対角成分に小さな正の値を加え、$\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi}+\lambda\boldsymbol{I}$（ただし、$\lambda$>0は小さな正の値）とすることで、$\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi}$ を正則にする

$$\left(\Phi^{\top}\Phi + \lambda\mathbf{I}\right)\mathbf{w} = \Phi^{\top}\mathbf{y} \quad \Rightarrow \quad \mathbf{w} = \left(\Phi^{\top}\Phi + \lambda\mathbf{I}\right)^{-1}\Phi^{\top}\mathbf{y}$$

  - あとで述べる「正則化」と密接な関係

リッジ回帰：過学習を防ぐ

訓練データ数よりも特徴空間の次元数が大きい場合には過学習と呼ばれる性能悪化の現象が起こります

- 多くのスジの良いケースでは、訓練データの数Nは、特徴空間の次元Dよりも十分に大きいため、前述の方法で、良い予測精度（汎化性能）を持つモデルが得られる
- しかし、特徴空間の次元が比較的高い場合には、いわゆる**過学習**と呼ばれる問題が起こる。
  - 過学習：訓練データに過剰に適合してしまうことで、むしろ汎化能力を失ってしまう現象
    - 我々の本来の目的は、訓練データの入出力関係を忠実に再現することではなく出力が未知の入力に対して、その出力を正しく出力すること（汎化）
- 特徴空間の次元Dと比較して、訓練データ数Nが大きくない場合には、連立方程式において、$\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi}$が正則でない、すなわち、フルランクでないことが多く、実質的に、変数の数よりも等式制約の数が少なくなってしまうため、解がいくらでも存在することになる。

モデル選択の一つの基準：なるべくシンプル（≒滑らかな）なモデルを採用せよ

- たくさんある解の中でよいモデルとは何だろうか？
- 「オッカムの剃刀」の教え：なるべくシンプルなモデルを採用せよ
- 「シンプルなモデルを採用することが本当に理論的に良いのか？」という問いに対する答えは後回しにして、とりあえず、その教えを信じることにする
  - なお、単純に経験則としてみても、シンプルなモデルを採用することは大抵の場合良い方向に働く
- 「シンプルなモデル」という気持ちの表現は様々考えられるが、ここでは「滑らかなモデル」とする

モデルのシンプルさはパラメータの2-ノルムで表現する

- モデルの「滑らかでなさ」は、$\mathbf{w}$ の2-ノルム$\|\mathbf{w}\|_2^2$ で表現する
- 我々が最小化すべき目的関数に$\|\mathbf{w}\|_2^2$を加える

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

  - λは0以上の定数（$\|\mathbf{w}\|_2^2$に対するペナルティの強さを調節）
- λは使用者が決定する必要があり**ハイパーパラメータ**と呼ぶ
  - のちにハイパーパラメータを自動的に決定する方法について学ぶ

2-ノルムをペナルティに用いた線形回帰の解は前述の「$\Phi^\top\Phi$ の正則化」と一致します

- 改めて新しい目的関数を$\mathbf{w}$について最小化してみる

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

- $\mathbf{w}$で偏微分すると $\dfrac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = 2\mathbf{\Phi}^\top (\mathbf{\Phi w} - \mathbf{y}) + 2\lambda \mathbf{w}$
  - 2項目が$\| \mathbf{w} \|_2^2$に由来する項
- これを $\mathbf{0}$ とおくと $\left(\mathbf{\Phi}^\top \mathbf{\Phi} + \lambda \mathbf{I}\right) \mathbf{w} = \mathbf{\Phi}^\top \mathbf{y}$

前述の式に一致する

- 2-ノルム正則化を用いた線形回帰のことを**リッジ回帰**と呼ぶ

正則化：パラメータノルムにペナルティを入れる

- また、パラメータのノルムにペナルティを課すことを**正則化**と呼ぶ
- パラメータの2-ノルム$\|\mathbf{w}\|_2^2$にペナルティを課すことをL2正則化／ティホノフ正則化／リッジ正則化などと呼ぶ
- 2-ノルムの他、よく使われるものとしては**1-ノルム**がある

$$\|\mathbf{w}\|_1 \equiv |w_1| + |w_2| + \cdots + |w_D|$$

  - 1-ノルムを用いた正則化をL_1正則化と呼ぶ
  - L_1正則化を施した線形回帰をラッソ（Lasso）と呼ぶ
  - これはL_2正則化と並び重要であるので、後ほど改めて述べる
- シンプルなモデル
  - 変数の少ないモデル：0-ノルム（凸でない）
  - 重みの小さいモデル：1-ノルム、2-ノルム（凸）

ここまでのまとめ

- 回帰問題は、実数値を予測する問題
- その代表的な定式化は2乗誤差を目的関数として使った線形回帰
- 線形回帰は逆行列（連立方程式）で解ける
- 次元数がデータ数と比較して大きい場合には過学習を防ぐために「シンプルなモデル」を選ぶ正則化を用いる
- 2ノルム正則化（L_2正則化／ティホノフ正則化）を使った線形回帰（リッジ回帰）も逆行列によって解ける