統計的機械学習 2017 年夏学期 シケプリ

@asarihamaguri01

2017 年 7 月 6 日

1 はじめに

2017 年度春学期より開講の鶴岡先生による「統計的機械学習」のシケプリです。基本的には授業スライドのまとめ、 省略されている用語の解説や勾配計算、口頭で説明された内容をまとめています。数式についてはできるだけチェックしていますが、 もしミスなどあれば教えていただけると嬉しいです。

2 統計的機械学習の基本

2.1 教師あり学習と教師なし学習 (第一講)

統計的機械学習で扱われる問題は教師あり学習」「教師なし学習」、「強化学習」に大別できます。統計的機械学習の授業では教師あり学習、教師なし学習についての扱います。

教師あり学習」と「教師なし学習

教室き学習 (supervised Learning)

入力から出力を予測するモデルを学習、正解が与えられている。

教師なし学習 (unsupervised learning)

予測の対象となる出力 (正解) が与えられない、データの構造や関係を解析

授業で紹介されたアメリカ東海岸エリアの労働者の賃金データです Wage[*1]は それぞれの労働者の age, sex, year, race 等の情報と賃金のデータが与えられているため、このデータを用いてモデルを正しく学習させることで年齢や性別等の情報が与えられた時にその人の賃金を予測するモデルを作ることができます。 このように入力、出力が共に与えられ、入力から出力を予測する回帰、分類等の問題を教師あり学習と言います。

一方、遺伝子の発現データである NCI60 Data[*2]は 特に出力が与えられているわけではなく、データを様々な軸でクラスタリングします。こういった正解ラベルを与えることなく、データからある一定数のクラスに分類する問題は教師なし学習に分類されます。

[*1] https://vincentarelbundock.github.io/Rdatasets/doc/ISLR/Wage.html

[*2] https://www.rdocumentation.org/packages/made4/versions/1.46.0/topics/NCI60

2.2 記法について

まず p 個の特徴量 (年齢、性別など予測を行うために参照できる情報) を持つ n 個入力データについては、i 番目の観測データの j 番目の変数を x_{ij} とすると、以下のようなベクトル X で表します。

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

ここで i 番目の観測データを表すベクトルを $x_i^T = (x_{i1}\ x_{i2}\ \dots x_{ip})$ とすると [*3] 以下のように記述することも可能です。

$$\boldsymbol{X} = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix}$$

一方、予測の対象は何らかのスカラー値で表されます（分類問題の際にはカテゴリの名前等になることもあります)。

$$\boldsymbol{y} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

観測データはこの入力 X 及び出力 y のペアで以下のように表されます。

$$\{(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)\}$$

(行列ベクトルの気泡)

2.3 統計的学習とは (第 3 講)

統計的学習とは、入力 $X = (X_1, X_2, \dots, X_p)$ と出力 Y の間にある関係が存在すると過程、入力データ X から Y を推定できることを目指します。

[*3] 蛇足かもしれませんが T は行列の転置を意味しています

入力と出力

入力

統計的機械学習において、予測行うために与えるデータ。Advertisement データセットにおける TV, Radio, newspaper など。入力変数、予測変数、特徴量（素性）などと呼ばれる。

出力

学習されたデータに入力データを与えた時に返される予測結果。出力変数、応答変数、従属変数などど呼ばれる。

統計的機械学習では、以上の入力 $X = (X_1, X_2, \ldots, X_p)$ と出力 Y の間にある何らかのシステマティックな関係 f ををを予測することを目指す。

$$Y = f(X) + \varepsilon$$

またここで ε は確率的な誤差 (error) を示しており平均はゼロになります。

統計的推定はパラメトリックな手法とノンパラメトリックな手法があります。

パラメトリックな手法とノンパラメトリックな手法

パラメトリックな手法

f の関数形について例えば線形モデルの $f(X) = \beta_0 + \beta_1 X_1$ 等の過程をあらかじめ置いて置き、学習を通して仮定された関数の係数 $\beta_0, \beta_1 \ldots \beta_p$ を求める。

ノンパラメトリックな手法

f の関数形について事前に明示的な仮定を置かない手法。

パラメトリックな手法は f の推定問題を $p+1$ 個のパラメータの値を求めるというシンプルなものに帰着できるという利点がある一方、 仮定されたモデルが真の f を表現できるかどうかは事前にはわからず、表現力の高いモデルを仮定すると過学習 [*4] が起こってしまう恐れがあります。ノンパラメトリックな手法は複雑な f を表現できる可能性があり、また高精度の推定のためにはパラメトリックモデルよりも大量の観測データが必要になるケースが多いです。

x と y の間に存在する関数 f を推定する目的としては、予測及び推論という二つの目的があります。

目的 1：予測 (prediction)

Y が容易に得られない場合に、X から Y を予測し、推定する。例えば患者に初めての薬を処方する際、その患者が服用して副作用が起こる可能性があるかどうか、事前に予測する必要がる。ここで過去の副作用発生事例から 副作用を起こした患者、起こしていない患者の年齢、持病、性別等のデータ X より、この新しい患者がある薬に対して重篤な副作用が起きるリスクいう出力 Y を事前に推定することができる。このように予測が目的の時は推定結果が正確である限り、$\hat{f}$ はブラックボックス化して構わない。

目的 2：推論 (inference)

$X_1, X_2, \ldots, X_p$ が変化した時に Y がどのように変わるかを知りたい時に行う。これはどの入力変数が Y に関係しているかを知り、活用することを目指しています。 例えば Advertising データセットにおける学習では、知りたいのは「何らかのシステマティックな関係により売り上げがいくつかになる」 ではなく、例えば TV 広告を増やした時、新聞広告を増やした時でどのくらい売り上げが変わるのかを知

[*4] 学習データの中のノイズにも適応し過ぎてしまい、新しい入力に対して正しい予測をできなくなってしまうこと

るため、 Y と入力変数 X の間にある線形もしくは非線形な関係を知り、どの X が Y に影響を与えているかそのウェイトの大きさをみる。

目的を予測にするか推論にするかにより、選択されるべき適切なモデルは異なってきます。 例えば推論を目的とするならば、推定した関数の中身の解釈が比較的容易な線形モデルや、どのように条件を分岐させているかを可視化できる決定木モデルなど、 独立変数がどう従属変数に影響を与えているか容易に確認ができる解釈性の高いモデルが適しています。一方で、線形モデルでは複雑非線形な関係に対応できず、決定機は入力データに左右されるため、 常に高い精度が得られるわけでありません。
予測のみを目的とするならば、対象が複雑な現象であっても高い予測精度が得られる可能性がある非線形モデルが適しているかもしれませんが、 非線形モデルは中身の解釈が難しく、推論には使いにくい傾向にあります [*5]。このように、精度の高いモデルでは解釈性が低く解釈性の高いモデルでは精度が低くなりがちな現象のことを予測精度と解釈性のトレードオフと言います。

予測精度と解釈性のトレードオフ

- 表現力の低いモデルでは複雑な形の f をうまく近似できない一方、 Y と $X_1, X_2 \dots X_p$ の関係がわかりやすい。(予測精度は低いが解釈性が高い)
- 表現力の高いモデルでは多様な f に対応が可能ですが、Y と $X_1, X_2 \dots X_p$ の関係がわかりづらい。(予測精度は高いが解釈性が低い)

(予測誤差の大きさ)

3 回帰と分類

3.1 変数の種類

統計的機械学習ではある一つ以上の入力変数 X から、出力 y を推定する f を求めことを目的としますが、この入力変数 X には連続した実数値となる量的変数と、 離散的なカテゴリ（ラベル）になる質的変数があります。

量的変数と質的変数

量的変数

年齢、身長、収入、株価などの連続的な実数値をとる変数。そのまま入力変数として使用することができる。

質的変数

性別、学歴、Yes/No、カテゴリの種類など、K 個の異なるクラス (カテゴリ) からどれかの値をとる変数。そのままでは入力変数として用いることができず、 後述するダミー変数の使用などにより予測モデルに組み込む。

教師あり学習は回帰問題と分類問題に大別することができ、回帰問題 (regression Problem) では予測変数が

[*5] 現在でも広く使用されている機械学習モデルの一つに SVM(サポートヴェクターマシン) というものがあり、これは 非線形な関係にも対応でき一般的なタスクにおいて線形モデルより高い精度を発揮することも多いのですが、線形モデルと比較するとどの入力変数が出力とどの程度影響を与えているか直接知ることはできません。 また機械学習ではないですが、今話題の深層学習もモデル自体をブラックボックス化してしまい、人間が直接的にどうそれぞれの入力変数が影響を与えているのか知る方法はありません

金融商品の価格等の量的変数となる一方、 分類問題 (classification problem) では予測対象が債務不履行に陥ったか否かなどの質的変数になります。

補足ですが、質的変数を入力変数として用いた予測モデルを作る際には「女性/男性」などの情報はそのままでは数理モデルに組み込むことができないので、それぞれの質的変数（質的変数に含まれるカテゴリごとに) ダミー変数を作成することが一般的です。 例えば「女性/男性」「大人（18 歳以上）/子供 (18 歳未満)」等ある質的変数について完全に 2 値で分けられるときには以下のように「女性であれば 1/男性であれば 2」等のように変数を作ります。

$$x_i^{gender} = \begin{cases} 1 & \text{(if ith person is a female)} \\ 0 & \text{(if ith person is a male)} \end{cases}$$

ではこのように「true/false」の 2 値で分類できない場合はどうするのでしょうか。例えば ethnicity(民族) について、とりあえず Caucasian = 2, African American = 1, or Asian = 0 とか 適当に数字を振ってしまったら [*6] もともと順序や序列の存在しないデータに恣意的に大小関係を加えることになり、モデルの予測を歪めてしまいかねません。 そこでこういった複数の値を取りうる質的変数についてはその全てに対してダミー変数を作り、「Asian ならば 1、そうでければ 0」といった風に数字を割り当てます。

$$x_i^{asian} = \begin{cases} 1 & \text{(if ith person is an Asian)} \\ 0 & \text{(if ith person is not an Asian)} \end{cases}$$

3.2 モデルの精度

モデルの精度の測り方には様々な方法がありますが、回帰問題の場合、平均二乗誤差 (mean squared error, MSE) を、分類問題の場合 error rate を用いることが一般的です。

平均二乗誤差及び error rate

平均二乗誤差 (MSE)

平均二乗誤差とは n 個の 学習データから構築されたモデルの予測した結果と実際のラベルと差を二乗し、平均を撮ったものであり、以下の式で表される。

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

誤り率 (error rate)

error rate は n 個の学習データに対する予測のうち、誤った予測を行ってしまった数をデータの総数 n で割ったもの。

$$\text{error rate} = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

ここで $I(y_i \neq \hat{y}_i)$ は $y_i \neq \hat{y}_i$ ならば 1 を、そうでなければ 0（すなわち正解していれば 0）をとる関数。

ちなみにこの式で表されるのは学習データでの誤差 (training MSE) ですが、本当に最小化したいのは未知

[*6] どっかの団体に怒られそうな例だ...

データでの誤差（test MSE）です。気をつけなくてはいけないのは、特に過学習をしている場合などは学習データにおける MSE がいくら高くても Test MSE が必ずしも少なくなっているとは限りません。

$$\mathrm{Ave}((y_0 - \hat{f}(x_0))^2)$$

error rate も同様に上の式で得られるのは学習データでの error rate に過ぎず以下の式で表される Test error rate を下げる必要があります。

$$\mathrm{Ave}(I(y_i \neq \hat{y}_i)^2)$$

またモデルの精度を測る指標としてバリアンス (variance) とバイアス (bias) があり、テスト MSE はこれらバリアンス及びバイアスで表すことができる。

バリアンスとバイアス

バリアンス

学習データの違いによってどれだけモデルによって予測される $\hat{f}$ の値が変わるかを示す。未知のデータに対してどのくらい正しく予測できているかを測る。

バイアス

真のラベル f と予測した値 $\hat{f}$ がどれだけずれているかを示す。学習データに対して、どのくらい正しく予測できているかを測る。

平滑化スプライン等の表現力の高いモデルは、学習モデルに対しては 高い精度で予測できる一方、学習データにオーバーフィットしやすく新しい未知データに対して正しく予測できない可能性が高まるため、バリアンスが高い。 一方線形回帰などの表現力のあまり高くないシンプルなモデルは、このようなオーバーフィットをしにくく、バリアンスは小さくなる傾向にあるが、学習データに対してフィットしきれずバイアスが大きくなります。このように、バイアスとバリアンスの間にあるトレードオフの関係を The bias-variance trade-off と言います。

The bias-variance trade-off

テスト MSE は以下の 3 つの項の和に分解でき、バイアスとバリアンスにはトレードオフの関係にあることが確認できる。(この導出については参考資料に)

- $\hat{f}(x_0)$ のバリアンス（分散）
- $\hat{f}(x_0)$ のバイアスの二乗
- ε の分散

$$\mathrm{TestMSE} = E[(y_0 - \hat{f}(x_0))^2] = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{\mathrm{f}}(\mathrm{x}_0))]^2 + \mathrm{Var}(\varepsilon)$$

3.3 ベイズ分類器

条件付き確率が最大になるようなクラスにあるデータ X_i を分類する方法です。ベイズ最適決定とも呼ばれます。

ベイズ分類器

以下の条件つき確率が最大になるクラスを選び、期待誤り率を最小になることを目指した。

$$P(Y = j|X = x_0)$$

ベイズ分類器の誤り率はベイズ誤り率 (Bayes error rate) と呼ばれ、以下の式で表される。

$$1 - E[\mathrm{maxP_r}(Y = j|X)]$$

これは回帰問題での削減不可能な誤差に相当している。

ただ実際のところ、条件付き確率を事前に正しく知ることはできないので、これはあくまでも理論上の分類器になります。 ベイズ分類器で分類された結果に基づいてデータの間に引かれた境界のことをベイズ決定境界といいます。

3.4 K 最近傍法 (K nearest meighbors method)

K 最近傍法は入力変数からそれぞれの入力が属するクラスを予測する分類問題をとくモデルの一つであり、自分に最も近い上位 K 個の観測データの属するカテゴリで多数決を行い、自身の最近傍 K 個の観測データの中で最も多く現れたラベルをその入力データのラベルとします。これを定式化すると以下のようになります。

k 最近傍法

クラスの条件付き確率を以下のように推定し、この確率が最も大きくなったクラスに分類する。

$$P_r(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$

この K の値を小さく、例えば $K = 1$ などにすると、それぞれ自身に最も近い観測データのみ考慮するため、決定境界はより入り組んだ、表現力の高いモデルになります。この場合学習データに対しては高い精度を誇るものの、 テストデータでは謝り率が高くなることがあります。前述の通りバイアスとバリアンスの間にはトレードオフの関係があり、モデルの表現力は高すぎても低すぎてもテストエラーが高くなってしまうことに注意が必要です。KNN で求められたラベルに基づいてデータ間に惹かれた境界線が KNN 決定境界になります。 図 1 の $K = 1$ の場合と $K = 100$ の場合の KNN 決定境界を見ると、実際に K の数が小さいほど境界線が入り組んでいる、つまり表現力が高くなっていることがわかります。 この例では $K = 1$ のケースの方がわずかに $K = 100$ よりも高いテスト誤り率になってはいますが、モデルの表現力が高いからといって、テスト謝り率は必ずしも下がるわけではありません。(ただし学習謝り率については K の数を減らすほど小さくなう傾向にあります)

3.5 線形回帰 (Linear regression)

3.5.1 線形単回帰

線形単回帰とは、X と Y の関係は線形だと仮定して、一つの予測変数 X によって量的応答変数 Y を予測する。

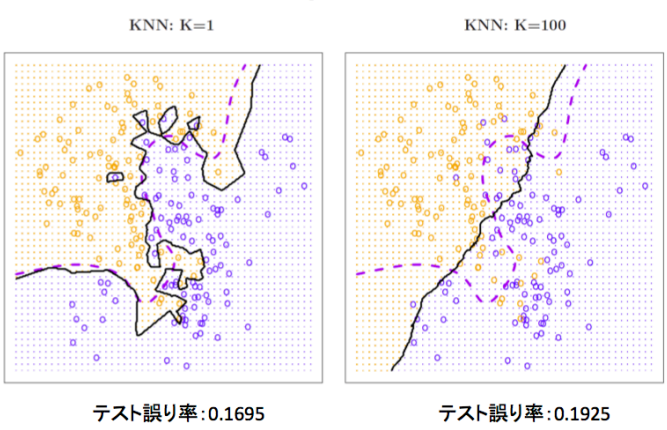

図 1　$K = 1$ 及び $K = 100$ の時の KNN 決定境界（黒太線）

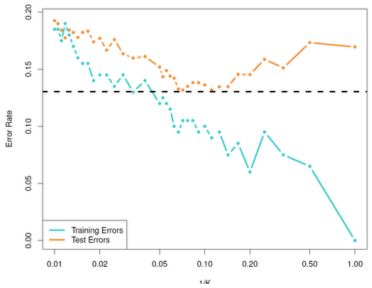

図 2　K の数に対する学習謝り率とテスト謝り率の推移

線形単回帰

X と Y の関係は以下の式のような線形で表されると仮定する。

$$Y \approx \beta_0 + \beta_1 X$$

学習データによって適切なパラメータを予測し、$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$ をより元の学習データの結果と近づけることを目指す。学習データを $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ とし、予測のずれを残差 $e_i = y_i - \hat{y}_i$ とすると、線形単回帰の目的関数は以下のように表される。RSS(残差平方和)

$$\mathrm{RSS} = e_1^2 + e_2^2 + e_3^2 + \ldots e_n^2 = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 \ldots$$

ここで RSS は学習データ X に対して推定されたパラメータ $\hat{\beta}_0$及び$\hat{\beta}_1$ を用いて線形のモデルで予測した $\hat{y}$ の値が、実際にそれぞれの y とどのくらいずれているかを二乗したものの和であり、 これを最小化することによりよりずれの少ないモデルを構築することができます。すなわち、線形単回帰においてパラメータを推定するときは残差平方和が最小になるようにパラメータを推定すればいいことになります。 目的関数 RSS を最小化する $\hat{\beta}_0$及び$\hat{\beta}_1$ は、RSS を $\hat{\beta}_0$及び$\hat{\beta}_1$ で微分した時に、それが 0 なるような $\hat{\beta}_0$及び$\hat{\beta}_1$ を求めれば良いため

$$\frac{\partial \mathrm{RSS}}{\partial \hat{\beta}_0} = 0, \frac{\partial \mathrm{RSS}}{\partial \hat{\beta}_1} = 0$$

を満たす $\hat{\beta}_0$及び$\hat{\beta}_1$ を求めると、以下のようになります。(詳細な変形などは参考資料に)

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2}, \hat{beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x}$$

ここで $\overline{x} = \frac{1}{n}\sum_{i-1}^{n} x_i, \overline{y} = \frac{1}{n}\sum_{i-1}^{n} y_i$ です。この結果求められた二つのパラメータ $\hat{\beta}_0, \hat{\beta}_1$ がどの程度正確なのかは母回帰直線を推定することにより求めることができます。

3.5.2　残差 RSS に関する性質

また、RSS を最小にする β_0, β_1 の導出より、以下の二つの性質が求められます。(導出は参考資料に)

$$\sum_{i=1}^{n} e_i = 0 \tag{1}$$

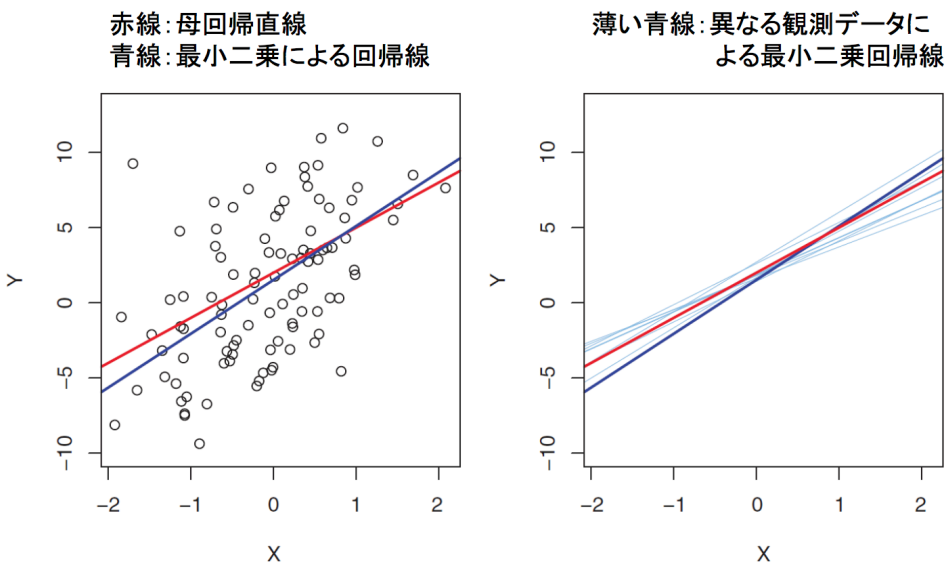

図 3　線形単回帰誤差

$$\sum_{i=1}^{n} e_i x_i = 0 \tag{2}$$

式 (1) は残差の合計が 0 ですことを、式 (2) は残差 e_i と入力 x はベクトルとして直行ですことを意味しています [*7]。

3.5.3　線形単回帰の信頼性

線形単回帰においては、母集団において X, Y の間に線形の関係性があると仮定しデータよりそのパラメータ β_0, β_1 を予測しており本来的に母集団の β_0, β_1 を予測することはできません。 そこでデータより得られたパラメータがその程度信頼できるのかを確認します。 母集団での X と Y の関係を $Y = \beta_0 + \beta_1 X + \varepsilon$ とすると、ε は母集団における統計的な誤差なので、母集団での回帰直線は $Y = \beta_0 + \beta_1 X$ であらわ s れる。 以下のように、$Y = 2 + 3X + \varepsilon$ という線形の関係を元に生成した人虎データに対する線形単回帰モデルに基づくパラメータ推定の結果は以下のようなグラフとなり、実際の母回帰直線と最小二乗による回帰直線の結果にややずれがあることがわかる。 また与えられた入力データにおける誤差項により異なる観測データに対する最小二乗回帰直線にはややズレがあることが右図よりわかります。では求められたパラメータより基礎統計や数理手法でやったように信頼区間を求めるにはどうしたらいいのでしょうか。そのためにまず各パラメータの標準誤差を求める必要があります。 個々の観測データの誤差は独立で分散が σ^2 と仮定するとパラメータの標準誤差は

$$SE(\hat{\beta}_0^2) = \sigma^2 \left[\frac{1}{n} + \frac{\overline{x}^2}{\sum_{i=1}^{n} (x_i - \overline{x})^2} \right]$$

$$SE(\hat{\beta}_1^2) = \frac{\sigma^2}{\sum_{i=1}^{n} (x_i - \overline{x})^2}$$

[*7] ベクトル a,b の内積がゼロですとき二つのベクトルは直行しているため。

ここで σ は未知ですため、RSS(残渣平方和) を用いて以下のように観測データから推定され RSE(redidual standard error) と呼ばれます。

$$\mathrm{RSE} = \sigma \approx \sqrt{\mathrm{RSS}/(n-2)}$$

SE の導出、及び次の信頼区間の導出については参考資料にあります。パラメータの信頼区間について、95% の確率でその区間にパラメータの真の値が含まれるための近似的な信頼区間はそれぞれ

$$\hat{\beta}_0 \pm 2\mathrm{SE}(\hat{\beta}_0), \hat{\beta}_1 \pm 2\mathrm{SE}(\hat{\beta}_1)$$

3.5.4 線形単回帰モデルがどの程度データにフィットしているか評価する

モデルがどの程度データにフィットしているか評価する指標として、RSD 及び決定係数 R^2 があります。決定係数とは回帰分析によって求められた目的変数の予測値が実際の目的変数の値とどのくらい一致しているかを表している指標です。厳密な定義としては「回帰分析をした結果が目的変数のばらつき分散をどれくらい説明しているか」ということを表しており、 もちろんですが 1 に近いほどデータを正しく説明できていることになります。

RSE(residualstandarderror)

誤差 ε の標準偏差の推定量のこと。以下の式で求めることができる。

$$RSE = \sqrt{\frac{1}{n-2}\mathrm{RSS}} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

次に紹介する決定係数も回帰モデルによって実データをどれくらい説明できているか、つまり回帰分析の制度を表す指標です。 こちらも 1 に近いほどモデルがデータをよく表現できていることを意味します。

決定係数 R^2

決定係数とは応答変数 Y の分散を X による回帰方程式で減らした割合。以下の式で定義される。

$$R^2 = \frac{TSS - \mathrm{RSS}}{TSS} = 1 - \frac{\mathrm{RSS}}{TSS}$$

ここで TSS(Total sum of square) とはもともとのデータが持っている分散の総和であり、以下の式で定義される。

$$TSS = \sum_{i=1}^{n}(y_i - \overline{y})^2$$

式からわかるように、第 2 項が小さいすなわち RSS $\ll TSS$ となり、元々のデータの持つ変動よりも回帰により予測された値と元データの間の変動の方が小さく、 モデルがうまくデータを説明できていることを意味します。

3.5.5 多重線形回帰

実際のデータにおいては、一つの入力変数だけでなく、複数の変数を用いて予測をしたいというシチュエーションも多く存在します。複数の単回帰モデルを個別に構築し、組み合わせるだけでは 予測変数同士の関係を考慮することができず、またどう個別のモデルを組み合わせるべきかは明らかになっていません。そこで多重線形回帰は、複数の予測変数を使う単一のモデルで回帰を行うことを目指しています。

多重線形回帰

以下のように複数の X と Y の関係を以下のようなモデルで回帰する。

$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$

β_j は他 予測変数を固定して、X_j を 1 単位増やしたときに増える Y 量の量を表している。学習データによって適切なパラメータを予測し、$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p$ を予測とし、学習データの残差平方和を以下のように表す。RSS(残差平方和)

$$\mathrm{RSS} = \sum_{i=1}^{n} (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} \ldots - \hat{\beta}_p x_{ip})^2$$

要は線形単回帰です一つの変数について、その入力変数が一単位増えたときに増加する Y の量 (パラメータ β_1) がモデルの傾きになっていましたが、 多重線形回帰ではある p 個の入力変数 $X_1, X_2, \ldots X_p$ それぞれにパラメータ β_i を求めてモデルを構築しただけです。多重線形回帰についても単線系回帰の場合と同様に、それぞれのパラメータで目的関数を微分したものが 0 となるよう計算することで求めることができます。 勾配情報を使って再急降下法、 準ニュートン法などを行うことによって計算できます。また、多重線形回帰の結果求められるのは回帰「平面」になります。(次元増えるからそれはそう)

3.5.6 どの予測変数が重要なのか

多重線形回帰を行う際に、全ての変数を元に Y を求めるても逆にノイズが混じってしまったり計算のコストがかかってしまったりと有効ではありません。 また全ての予測変数の考えうる組み合わせ全てを試して精度を調べるのもコストが大きいです。そこで多重線形回帰では事前にどの予測変数が重要ですかの見当をつけモデルを構築するのが一般的です [*8]。変数選択方には以下のような手法が一般的です。

変数選択法の種類

変数増加法

予測変数を全く入れない状態からモデルの構築を始め、ある一つの変数だけを今あるモデルに加えたとき、その RSS が最も小さい（つまりある変数を加えることによりよりデータを説明できるモデルを作ることのできる）予測変数を追加するということを繰り返す。

変数減少法

全ての予測変数を使うモデルから出発し、p 値が最も大きい（つまり結果と関係の小さい）予測変数を削除するということを繰り返す

変数増減法

変数増加法と変数削減法を組み合わせたもの。

蛇足かもしれませんが、p 値とは帰無仮説 H_0 のもとで，統計量（確率変数）T が、データから実際に計算した統計量の値 T_0 よりも「極端」な値を取る確率を意味し、 実用的な意味でいうと p 値が有意水準（5%）とかより小さければ帰無仮説を棄却します。

[*8] Azu ○ eML 勉強会で関係なさそう、もしくは id などノイズを与えてそうな Housing データの 入力変数を除いて見たりした人もいたのではないでしょうか。要はあんな感じです。

3.5.7 多重線形回帰モデルがどの程度データにフィットしているか評価する

線形単回帰モデルと同様に、多重線形回帰においてモデルがどのくらいデータに適合しているかは決定係数及び RSE によって評価することができます。決定係数は線形単回帰と同様、

$$R^2 = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}}$$

で表すことができますが、多重線形回帰の場合は 1+p だけ自由度が減少するため、以下の式で定義されます。

$$\mathrm{RSE} = \sqrt{\frac{1}{n-1-p}\mathrm{RSS}}$$

3.5.8 多重線形回帰モデルにおける変数同士の相互作用

線形単回帰モデルの時は入力変数は一つだけだったため、単一の入力変数と予測変数の間の関係のみに着目するだけですみました。 しかし多重線形回帰については、描く予測変数間の応答変数に対する効果は必ずしも独立しているとは言えず（例えば Advertisement データにおける TV と radio という二つの変数はそれぞれ独立して売り上げに貢献しているのではなく、その二つを一緒に行うことによってより宣伝効果をあげているかもしれません)、多数の変数間の相互作用を考える必要があります。 それぞれの変数が独立だとした時のモデルは以下のように関連していると考えられる変数の積を新たな変数としてモデルに追加することで実現できます。

$$Y = \beta_0 + \beta_1 X_1 + \beta_3 X_1 X_2$$

3.5.9 多重線形回帰で起こりうる問題

多重線形回帰では線形回帰モデルで解決できない非線形な関係に対応することを目指していました。しかし多重線形モデルでは以下のような問題も発生してしまいます。

誤差項の相関

時系列データ (time-series data) などでは誤差項に 相関があることが多い。(so what)

誤差項の相関

Y の値が大きいところでは誤差項の分散も大きくなってしまう。そこで Y を予測するのではなく、$\log Y$ を予測するようにするとこれを抑えることができる。

外れ値

計測エラーなどによる外れ値が推論に大きな影響を与える可能性があり、こういった外れ値は回帰直線には影響を与えにくいが決定係数などに大きな影響を与えうる。そこで絶対値が 3 を超えるような値は外れ値の可能性が高いため、 除外をした方が良い。

てこ比

共線性 (collinearity)　予測変数の間に強い関連がある場合間に強い相関がある場合、そのままではパラメータ推定の信頼性が落ちてしまう。 そのため相関行列により 2 つの予測変数の間の関係を検出する必要があり、仮に二つの予測変数が強い相関を持っていた場合、これらを一つに統合する。

多重共線性

三つ以上の変数の間に共線性がある場合、VIF によって検出が可能である。

$$VIF(\hat{beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}}$$

3.5.10 多項式回帰 (polynomial regression)

これまでは全ての変数について、ある入力変数 X^p の予測変数 Y にに与える影響は線形的であるとしてモデルを考えてきました。 しかし実際には Youtube で流れる広告と売り上げの間に二次関数的な関係性が存在する可能性もあります。こういった非線形な関係を考慮するためには多項式回帰を行います。 予測変数を追加する必要があります。

3.6 パラメトリックモデルとノンパラメトリックモデルの比較

線形単回帰や多重単回帰のようにデータから目的関数を最小化するようなパラメータを推定し、そのパラメータに基づいてある入力 X に対して予測を行うモデルをパラメトリック、 KNN のようにパラメータを推定するのではなく逐次元のデータを読み込み、予測するモデルをノンパラメトリックモデルと言います [*9]。与えられたデータに非線形性が強いなど、ノンパラメトリックな手法の方が有利に働くケースも存在しないわけではないのですが、 パラメトリックなモデルノンパラメトリックなモデルの精度が同じくらいなのではあればパラメトリックなモデルの方が良いとされることが多いです。理由としては

- ノンパラメトリック手法では一度学習したデータを保持し続ける必要があり、一度学習すればあとはパラメータ飲み保持して予測を行うことができるパラメトリックなモデルより取り回しのしにくいという点があります。
- ノンパラメトリックなモデルの方が次元の呪い（入力変数の数が増える、すなわち予測する空間の次元が大きくなるほど適切な予測ができにくくなる）の影響を強く受けてしまう

ことが挙げられます。

4 分類

4.1 分類問題とは

分類問題とは応答変数が質的変数です予測問題であり、クラスを予測する問題とも言えます。 例えば患者の症状から疾患を予測する、メールがスパムかどうかを判定するなどはこのような分類問題の例です。 ところでなぜ分類問題と回帰問題を別の問題として考えなくてはいけないのでしょうか。これは特に 3 クラス以上の分類などにおいて、 線形回帰により分類を行おうと考えると、まずカテゴリに順序等をつけ出てくるスコアの大きさに応じて三段階に振り分ける、などそもそも明確にカテゴリ同士に順序がないはずのものについても 恣意的な仮定を置く必要が出てきてしまい、正しい予測をできない可能性があるためです。

[*9] パラメトリックモデルとノンパラメトリックモデル http://www.snap-tck.com/room04/c01/stat/stat99/stat0103.pdf

	Coefficient	Std. error	Z-static	p-value
Intercept	−10.6513	0.3612	−29.5	< 0.0001
balance	0.0055	0.0002	24.9	< 0.0001

図 4　ロジスティック回帰により求められた値

4.2　ロジスティック回帰による分類問題

以下では回 d 帰の中でも一般的な、ロジスティックモデルについて説明します。

ロジスティックモデル

ロジスティックモデルとは、以下のロジスティック関数を用いて確率値を計算するモデル。

$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

オッズ

$$\frac{p(x)}{1-p(x)} = \exp \beta_0 + \beta_1 x$$

ロジット

$$\log \frac{p(x)}{1-p(x)} = \log \exp \beta_0 + \beta_1 x = \beta_0 + \beta_1 x$$

ロジスティック回帰におけるパラメータは確率モデルのパラメータ推定に広く使われる、最尤推定により決定されます。

最尤推定

パラメータの尤度 (学習データが得られる確率) を計算する以下のような関数を尤度関数といい、これを最大にするような β_0, β_1 の値を計算する。

$$l(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y'_i=0} (1-p(x'_i))$$

ロジスティック回帰で求められたパラメータの値から予測値を求める方法は、以下のようにある人の講座残から債務不履行 (default) する確率を予測するモデルのパラメータをロジスティック推定により求めた結果が以下の場合、切片 ($\hat{\beta}_0$) が-10.6513、balance について 1 単位増加させた時の予測変数の増加分 $\hat{\beta}_1$ が 0.0055 となるので、例えば $1000 の人が default する確率は

$$\hat{p}(X) = \frac{\exp(\hat{\beta}_0 + \hat{\beta}_1 X)}{1+\exp(\hat{\beta}_0 + \hat{\beta}_1 X)} = \frac{\exp(-10.6513 + 0.0055 \times 1000)}{1+\exp(-10.6513 + 0.0055 \times 1000)} = 0.00576$$

よって balance が $1000 の人がデフォルトする確率は 0.5% と極めて低いことがわかる。質的変数（あるカテゴリに対してそれぞれダミー変数を設定したもの）を用いて予想を行う場合は以下のような学生であるかどう

	Coefficient	Std. error	Z-static	p-value
Intercept	−3.5041	0.0707	−49.55	< 0.0001
student [Yes]	0.4049	0.1150	3.52	0.0004

図 5　ロジスティック回帰により求められた値 (定性情報)

かというダミー変数に対して予測されたパラメータを用いて予測を行う場合、学生である人は $x = 1$、学生でない人は $x = 0$ とすれば良いので、学生のデフォルトする確率は

$$\hat{p}(X) = \frac{\exp(\hat{\beta}_0 + \hat{\beta}_1 X)}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_1 X)} = \frac{\exp(-3.5041 + 0.4049 \times 1)}{1 + \exp(-3.5041 + 0.4049 \times 1)} = 0.00431$$

上記の式で x=0 とした時の確率が 0.0292 より、学生の方がややデフォルトする確率が高いとこのモデルからは予測できます [*10]。

またロジスティック回帰において応答変数（予測するラベル、default するかどうか、患者の病気の症状など）が多数の値を取る場合は、ソフトマックス回帰により、最も確率が高くなるラベルを選択肢します。

ソフトマックス回帰

ソフトマックス関数は、n 次元の実数ベクトル $x = (x_1 \dots x_n)$ を受け取って以下で定義される n 次元実数ベクトル $y = (y_1 \dots y_n)$ を返す関数である。

$$y_i = \frac{e^{x_i}}{e^{x_1} + e^{x_2} \dots + e^{x_n}}$$

ここである予測変数のカテゴリ k について、$e_k = \exp(\sum_p \beta_{kp} X_p)$ とすると、

$$\mathrm{softmax}(x)_k = P(Y = k|X) = \frac{e_k}{\sum_{k=1}^{n} e_i}$$

ソフトマックス回帰で行なっているのは、ラベルの数が n 個存在していたとき、その k 番目のラベル（カテゴリ) の発生確率はそれぞれのオッズを正規化（オッズを足し合わせて、k 番目のラベルのオッズをその総和で割っている）しある入力 X が与えられた時、それが多数うカテゴリの中の k 番目のラベルである確率 を求めています。つまり、それぞれのカテゴリの予測確率を足し合わせると必ず 1 になります [*11]。

4.3　複数の予測変数を用いたロジスティック回帰

線形単回帰と同様に、ロジスティック回帰も複数の予測変数を取り入れたモデルに拡張することができます。複数の予測変数を利用したロジスティック回帰を多重ロジスティック回帰といいます。

[*10] 実際には後述するように必ずしも学生がデフォルトしやすいわけではなく交絡によりそう見えているだけだったりします。

[*11] ちなみにみんな大好き DEEP LEARNING ではとてもよくでてきます。きになる人は「mnist sotmax」とかでググってみてください。

	Coefficient	Std. error	Z-static	p-value
Intercept	−10.8690	0.4923	−22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student [Yes]	−0.6468	0.2362	−2.74	0.0062

図 6　多重ロジスティック回帰

多重ロジスティック回帰 (multiple logistic regression)

多数の予測変数を用いたロジスティック回帰モデルを多重ロジスティック回帰という。

$$\text{ロジット} = \frac{p(X)}{1-p(x)} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \ldots \beta_p X_p$$

$$p(X) = \frac{\exp\left(\beta_0 + \beta_1 X_1 + \beta_2 X_2 \ldots \beta_p X_p\right)}{1 + \exp\left(\beta_0 + \beta_1 X_1 + \beta_2 X_2 \ldots \beta_p X_p\right)}$$

多数ロジスティック回帰についても、計算方法は入力変数を一つにしたロジスティック回帰と基本的には同じです。先ほどと同様に、balance, income, student を使ってある人が default するかどうかを予測してみます。balance が $1,500、income が $40,000 の学生が default する確率は、上の表で切片 $\beta_0 = -10.8690$、balance にかかるパラメータ $\beta_1 = 0.0057$、income にかかるパラメータ $\beta_2 = 0.0030$、student にかかるパラメータ $\beta_3 = -0.6468$ より、

$$\hat{p}(X) = \frac{\exp\left(-10.8690 + 0.0057 \times 1500 + 0.0030 \times 40,000 + (-0.6468) \times 1\right)}{\exp\left(-10.8690 + 0.0057 \times 1500 + 0.0030 \times 40,000 + (-0.6468) \times 1\right)} = 0.058$$

ちなみに式からわかるかもしれませんが、このモデルにおいて学生かどうかを表すダミー変数にかかる $\beta_3 < 0$ となっているため、 同じ balance, income であれば学生ではない方が default する確率は低いと予測されます。これが先ほど軽く補足した交絡 (confounding) による効果であり、 交絡とは統計モデルの中の従属変数と独立変数の両方に（肯定的または否定的に）相関する外部変数が存在することを意味します。例えば「飲酒者と非飲酒者では飲酒者の肺癌発生率が高くなる」というのがデータからは確かに観測されるのですが、 これは直接的に飲酒が肺がんの要因になっているわけではなく [*12]、非飲酒者と比較して飲酒者における喫煙者の割合が高いために、 結果的にデータから「飲酒者の方が肺がんになりやすい」という傾向がでてしまうことになります。今回の問題では実際には学生だから default を起こしやすいのではなく、学生の方が非学と比べ income が少ない、credit card balance が大きい等の要因によりデフォルト率が高くなっています。

4.4　線形判別分析（第 7 講）

ロジスティック回帰のように直接 $\Pr(Y|X = x)$ をモデル化するのではなくある入力変数 X が与えられた時にそれがある応答変数 Y となる確率を予測する別のモデルに「線形判別分析 (LDA)」があります。

[*12] もしかしたら何か関係あるのかもしれないけどそういうのは医学部の人に聞いてください

線形判別分析 (linear discriminant analysis)

線形判別分析とは、$\Pr(Y|X=x)$ を正規分布でモデル化し、ベイズの定理によって $\Pr(Y|X=x)$ を計算するモデル。K をクラスの数、π_k をラス k の事前分布、$f_k(x)$ をクラス k のデータの密度関数とすると、$\Pr(Y|X=x)$ は以下のように表される。

$$\begin{aligned}\Pr(Y|X=x) &= \frac{\Pr(Y=k)\Pr(X=x|Y=k)}{\Pr(X=x)} \\ &\frac{\Pr(Y=k)\Pr(X=x|Y=k)}{\sum_{l=1}^{K}\Pr(X=x,Y=l)} \\ &= \frac{\Pr(Y=k)\Pr(X=x|Y=k)}{\sum_{l=1}^{K}\Pr(Y=l)\Pr(X=x|Y=l)}\end{aligned} \tag{3}$$

$$p_k(x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K}\pi_l f_l(x)} \tag{4}$$

$f_k(x)$ が正規分布だと仮定すると、$f_k(x)$ は以下の式で表される。

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_{x_k})^2\right) \tag{5}$$

分散が全てのクラスで等しい場合、ある入力 x が与えられた時、それが k である確率は式 (5) を式 (4) に代入することで求めることができ、

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_{x_k})^2\right)}{\sum_{l=1}^{K}\pi_l \frac{1}{\sqrt{2\pi}\sigma_l}\exp\left(-\frac{1}{2\sigma_l^2}(x-\mu_{x_l})^2\right)} \tag{6}$$

式 (6) より、$p_k(x)$ は $\pi_k \frac{1}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_{x_k})^2\right)$ に比例するため、 これが最も大きくなるクラスに入力データを分類すればいいことがわかります。これについて対数をとると

$$\log\pi_k + \log 1 - \log\sqrt{2\pi}\sigma + \left(-\frac{1}{2\sigma^2}(s-\mu_k)^2\right)$$

となり、予測に関わるのは π_k を含んだ項のみとなるため、

$$\log\pi_k + x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} = \delta_k(x) \tag{7}$$

として、この $\delta_k(x)$ を最大にするクラス k に事例 x を分類すると考えることができます。すなわち K 個のクラスに対して μ_k及びσ_k^2 がわかれば分離境界 x を求めることができます。

5 リサンプリング法

リサンプリング法とは、手持ちの限られたデータをある固定された割合で分割し、訓練データとテストデータに分けて学習するのではなく、 ランダムサンプリングと学習によるパラメータ推定を複数回行い、データの分割の仕方による予測モデルのばらつきなどを少なくすることを目指しています。 試行回数が増える分計算コスト画像化するため、かつてはこれがネックになっていましたが、計算機の能力向上によりこういったコストの問題は小さくなっています。 以下で代表的な理サンプリング手法について紹介します。

5.1 検証セットによる手法

データセットをランダムに「学習用データセット (training set)」と「検証用データセット (validation set)」の二つにランダムに分割し、 学習用データでモデルのパラメータを推定し、検証用データで精度を評価するという、最もベーシックな方法です。シンプルな一方で、 データをどう分割するかによって制度の推定値やパラメータの予測値は大きく変わってしまうこと、 またデータ自体が少ない時にデータ全体の数割から半分ほどのデータを学習に用いることができないため、制度も本来達成可能な制度より低めに推定される可能性があります。

5.2 交差検証

検証セットによる精度を測定する方法を補うため、交差検証法はデータセットの全てのデータを用い、また繰り返し学習したパラメータの平均値を取ることでデータの分割の仕方による ばらつきの影響を排除することを目指しています。

Leave-One-Out Cross Valdation(LOO 黄砂検証)

N 個の学習データ $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$ のうち、1 つをのぞいた N-1 個で学習を行い、毎回残された 1 個のデータで性能を評価、 これを全ての事例に対して行いその平均値をとｓることで、最終的な評価指標の推定値とする。

$$\mathrm{MSE}_1 = (\hat{y}_1 - y_1)^2$$

$$\dots$$

$$\mathrm{MSE}_\mathrm{n} = (\hat{y}_n - y_n)^2$$

$$\mathrm{CV(n)} = \frac{1}{n}\sum_{i=1}^{n} MSE_i$$

LOO 交差検証には以下のようなメリットデメリットが存在します。

LOO 交差検証のメリット

学習には毎回 n-1 個のデータが使用され n 個全てのデータを用いた際とほぼ同じ性能のモデルを作ることができる。 また分割方法の違いによる性能評価のばらつきもない。

LOO 交差検証のデメリット

n 回の学習が必要になり、かつ学習データと検証データに一定割合で分割する際と比べ学習に 使用するデータの数も増えるため、データの数が多いほど、学習に必要なコストが大きく増加する。

このように、LOO 交差検証は学習データと検証データに分割する際のデメリットを克服できる一方、 計算コストが大きく増加してしまいます。そこで、一つだけを性能評価用に残し、n 回学習を行うのではなく、データをある数 k に分割し、一つのグループ以外のデータで学習、残された一つのデータで性能検証を行うことを繰り返すことで 計算量を抑えつつも性能の向上を目指すのが次の k 分割交差検証です。

k-fold Cross Valdation(k 分割交差検証)

データを k 個のグループに分割し一つを検証データ、残りの $k+1$ 個を学習データとして MSE を計算し、これを k 回繰り返して平均を取り、 この平均をテスト MSE として推定する。一般的に $k = 5, 10$ 等が用いられる。

$$CV(k) = \frac{1}{k}\sum_{i=1}^{k} \mathrm{MSE}_i$$

LOO 交差検証では一般に n 回の学習を行う必要があったが、k 分割時黄砂検証では学習回数はたかだか k 回ですむ一方、LOO 交差検証と同様、k 回の学習を通して全ての入力データをみることができます。また LOO 及び k 分割交差検証の間には以下のような Bias-variance trade-off が存在します。

Bias-variance trade-off

- k 分割交差検証の場合、学習に使えるデータの数は LOO 比べ少なくなるので、バイアスが大きくなる可能性がある。
- LOO 交差検証には学習に使うデータが毎回ほとんど同じなので出力間に強い相関があり、バイアスが大きい。

5.3 ブートストラップ

また、こういったデータを分割し、繰り返し学習左折以外に、ブートストラップという方法もあります。

ブートストラップ

標本からの復元抽出によるランダムサンプリングを繰り返すことで行う統計的推論手法。確率モデルのパラメータの信頼区間などを簡単に求めることができる。

復元抽出とは、抽出を行う際に一度抽出したサンプルが再び抽出の対象となりうる方法のこと。例えば受験数学でよくでてくる「赤い玉 3 個、白い玉 4 個入った袋から中身を見ずに一個取り出し色を確認して戻す作業を 3 回行った時、 全て赤い玉であった確率は」などは復元抽出の例であり、「一度出した玉は元に戻さない」であればこれは非復元抽出になります。具体的なブートストラップ方の利用としては、

1. 手持ちのデータセットからランダムサンプリングによって新たなデータセットを生成。
2. この抽出されたデータを元にモデルのパラメータを推定する。
3. これを多数回繰り返す。
4. 得られたパラメータの分布より信頼区間などを計算する。

6 正則化（第 10 講）

6.1 パラメータ推定法と縮小推定法

最小二乗法、最尤推定法などのパラメータ推定方では、モデルがデータにフィットするようにパラメータを最適化する。 一方リッジ回帰、Lasso 回帰は縮小推定に分類され、各パラメータの絶対値が大きくならないようにする。正則化ともいう。 最小二乗法による線形回帰パラメータ推定は、以下の RSS を最小化にするパラ

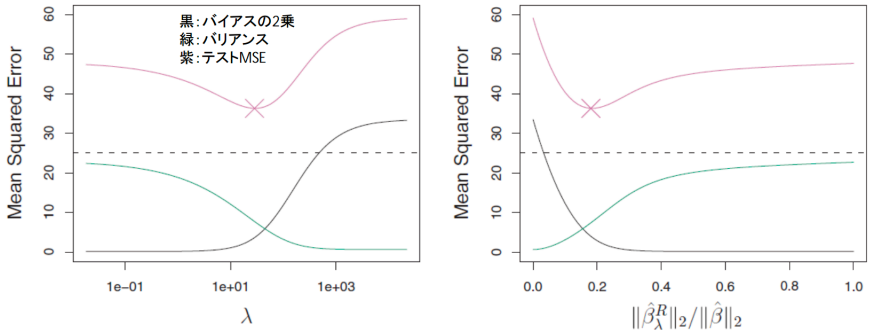

図 7　Credit データセットに対してリッジ回帰

メーラ $\beta_0, beta_1..., \beta_p$ を決定します。 これは学習データへのフィティングしか考慮していないため、過学習してしまいやすく、未知のデータに対して適切に予測できない恐れがあります。 そこでリッジ回帰やロッソ回帰などの縮小推定法により過学習を回避し、未知データに対しても頑健なモデルを構築します。

6.2　リッジ回帰

リッジ回帰

リッジ回帰の目的関数は以下で表される。

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\underline{\sum_{j=1}^{p}\beta_j^2}$$

下線で示され項がパラメータの絶対値が大きくなることに対するペナルティ（L2 正則化項）となる。

λ : チューニングパラメータ。

L2 正則化項の λ を大きくするほど、各パラメータの値 $\beta_0, \beta_1..., \beta_p$ はゼロに近づく。 またリッジ回帰においては、値の大きな予測変数が大きな影響力を持つようになるため予測変数のスケールが性能に大きな影響を与えてしまいます。 そのため、前処理として以下のような式で予測変数の標準化を行う必要があります。

$$\widetilde{x_{ij}}=\frac{x_i j}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i j-\overline{x_j})^2}} \tag{8}$$

λ を大きくするとバリアンスが小さくなる。しかしあまりにも小さくすると、$\beta_1, \beta_2..., \beta_p$ のパラメータは に 0 近づき、 元のデータから離れすぎてしまうため、今度はバイアスが大きくなってしまいます。そこでちょうどいい λ を選択することにより、バイアス、バリアンスを 小さくすることができます。

$$E[(y_0-\hat{f}(x_0))^2]=\mathrm{Var}(\hat{f}(x_0))+[\mathrm{Bias}\hat{f}(x_0)]^2+\mathrm{Var}(\varepsilon) \tag{9}$$

以下のグラフのバツ印のところで、バイアス、バリアンス共に小さくなっており、予測において使用すると良い λ の値になっていることがわかります。

- Credit データセット

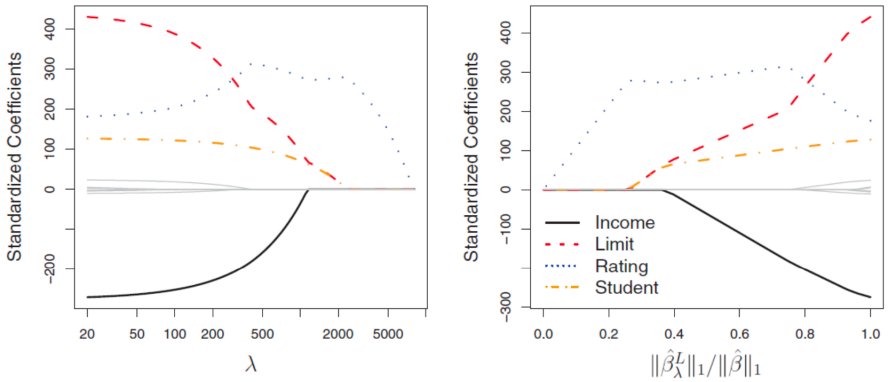

図 8　Credit データセットに対して Lasso 回帰

6.3　Lasso 回帰

Lasso 回帰は、正則化項をパラメータの二乗ではなく、絶対値とした目的関数を最小化することによりパラメータを求める方法です。

Lasso 回帰

Lasso 回帰の目的関数は以下で表される。

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\underline{\sum_{j=1}^{p}|\beta_j|}$$

下線で示され項がパラメータの絶対値が大きくなることに対するペナルティ（L1 正則化項）となる。

λ：チューニングパラメータ。

Lasso 回帰の目的関数はリッジ回帰と比べ、最後の正則化項が異なっているだけに見えますが、この結果得られるパラメータの性質が異なります。 リッジ回帰でパラメータ β が 0 に近づくにつれ、L2 正則化項はより急速に 0 に近づく。一方 Lasso 回帰では、正則化項は $beta$ の絶対値の和になるため、 正則化項は β に比例して小さくなる。結果として、Lasso 回帰では λ の値を大きくするにつれ、多くのパラメータが急速に 0 に近づくため、より重要なパラメータ以外の 枝葉のパラメータが自動的に覗かれてくる。特に多くのパラメータを保持したくない場合（例えば携帯端末に保存するモデルを作るときなど）はこちらの方が適していると言える。 以下のグラフは Credit データセットに対して Lasso 回帰を行い、λ を大きくした際のパラメータと出力の Coefficient を示しており、λ が大きく なるにつれて、多くのパラメータが急速に 0 に近づいていることがわかる。

リッジ回帰、Lasso 回帰どちらが適切なのかはデータの性質によって異なる。また、適切な λ の選択については、黄砂検証によってもっとも $TestMSE$ が小さくなる λ を選択する。

6.4 交差検証による λ のチューニング

リッジ回帰、Lasso 回帰における正則化項にかかる係数 λ の調整においては以前の項で紹介された交差検証により行います。様々な λ の値に対して交差検証で Test MSE を推定し、最も精度が良くなる λ を選びます。
` そのほか補足 Elastic Net-¿L1 正則化項と L2 正則化項の両方を追加
最適化 (パラメータ推定)-¿L1 正則化では勾配の計算が難しくなる。L2 は事情が並んでいるだけなので簡単。L1 は絶対値があるので簡単ではない。最適化が難しくなってしまう。 パラメータを減らすことができるという利点はあるが。データサイズが大きい時は勾配の計算（数値計算）が面倒になる。

6.5 高次元データでの学習

入力データによっては、予測変数（特徴量）p が学習事例の個数 n よりも大きくなることが大きくなることがある。例えば、血圧の予測では、特徴量として年齢、性別、BMI の他に遺伝子情報（SNPs）を特徴量として入れる場合、遺伝子情報は個々人によって少しずつ異なりそれぞれにダミー変数を振り分けていくと予測変数の数は膨大に増加する（$p \approx 500000$)。仮に患者データが 200 人分ほど ($n \approx 200$) であれば $p \gg n$ となってしまう。 他にもスパムメールの分類で単語を特徴量として用いる時、すでにスパム、スパムでないのラベルがついたメールが 1000 件前後しかない時に、そこに含まれている全ての単語を特徴量として用いれば、 特徴量の数はデータの数をたやすく上回ってし舞います。$p \geq n$ のとき、必要ではないノイズな予測変数（例えばスパムメール内でも実際には分類に大きく関係しない単語は多々存在しています）でもフィットしてしまうため、 過学習に陥る可能性が高くなります。

これに対する対処法としては、正則化や変数選択などによりモデルの柔軟性を下げ、必要ではない予測変数を減らします。しかしこの方法にはいくつか注意点があり、例えば多重共線性 [*13] により、変数選択によって選ばれた変数以外にも有用な変数がぞんざいする可能性があります。 また学習データで得られる RSS や決定係数は、実際のデータでも完全に当てはまるとはいえず、モデル の性能の指標としては無意味ですことも注意が必要です。

7 決定木 (decision tree)

木構造によってて回帰や分類のための条件を表すモデルであり、回帰木（regression tree)、分類木 (classification tree) がある。

7.1 回帰木

回帰木とは、ある数値（連続値）の推定のルールをツリーで表現したものです。Hitters データセット [*14] は、選手の在籍年数（Years）と打数 (Hits) より Log(Salary) を予測するデータセットです。回帰木を用いて Salary を予測する時、以下のようにまず「Years ¡ 4.5」なら選手の Log(Salary) は 5.11 に決定され、 そうでない、かつ「Hits ¡ 117.5」ならば 6.00、「Years ¡ 4.5」かつ「Hits $\geq$ 117.5」ならば 6.74 となる。回帰木は予

[*13] ある変数間で相関がある時、それが解析に支障が出るほど X 同士の相関が強い時に、「多重共線性（たじゅうきょうせんせい）があると言います。 http://xica.net/vno4ul5p/
[*14] https://vincentarelbundock.github.io/Rdatasets/doc/vcd/Hitters.html

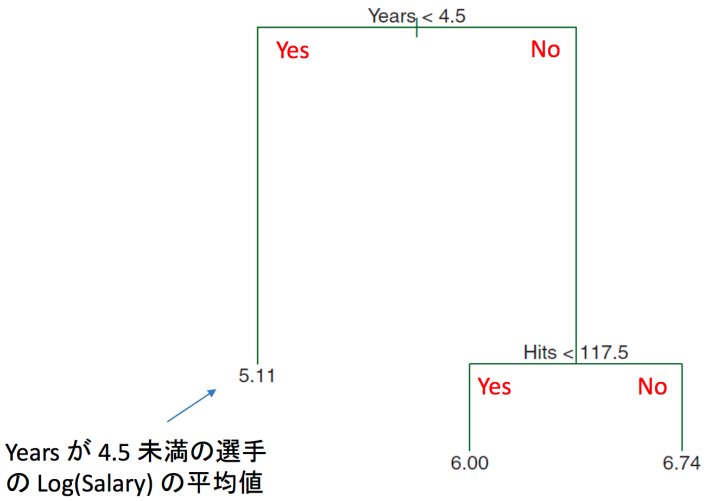

図 9　Hitters データに対する回帰木予測

測変数の空間を K 個のリージョンに分割することにより、分岐条件を求める。

回帰木の大まかの作り方は以下のようになります。

1. 予測変数の空間を J 個の領域 $R_1, R_2, \ldots, R_j$ に区分けをする。
2. 予測変数の値が領域 R_j に含まれる入力に対し、R_j に含まれる学習データの応答変数の平均値を出力する。

では予測変数の空間を J 個の領域 $R_1, R_2, \ldots, R_j$ に区分ける方法として、RSS を最小化する区分けは計算量的に難しいため、とりあえず今ある空間を全ての予測変数と分割の基準値の組み合わせを考え、 学習データの正解ラベルとモデルによって予測された値の差分の二乗和が最小になるよっっっっそく変数と分割の基準値を見つける方法があります。

貪欲法による領域の分割

全ての予測変数 $X_1, X_2, \ldots, X_p$ と分割の基準値 s を組み合わせ、以下が最小になる分割方法を選択する。

$$\sum_{i:x_i \in R_1(j,s)} (y_i - y_{R_i})^2 + R_2(\hat{j,s})(y_i - \hat{y}_{R_2})^2$$

$$R_1(j,s) = X|X_j < s$$

$$R_2(j,s) = X|X_j \geq s$$

また、分割によって作られた二つの領域をさらに同様に分割し、領域に含まれる事例の数が閾値以下になるまで繰り返し再帰的に領域を構築していくが、 こういった再帰的な分割方法だと領域同士が入り組んだ分割を行うことはできない。

回帰木はデータに対して容易にオーバーフィットしてしまうため、枝刈りによって木のサイズを小さくして過学習を抑制する必要があります。

枝刈り

一旦木を構築してしまってから、Cost Complexity Pruning によって多すぎる枝分かれ（領域）を減らしていくこと。

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{Rm})^2 + \alpha |T|$$

ここで T は木のノード数です。

Cost Complexity Pruning は、木のノードの数（すなわち枝分かれの数）に対してペナルティをつけて行き、ただ RSS でフィットさせるのではなく、どこの枝同士をマージすべきかを計算して求めます。ここで α は任意の値であり、 α を増やすと木は縮みます。このチューニングも縮小推定の λ のチューニングと同様、黄砂検証によって行います。 実際に Hitters データでは、枝仮によってある程度エラーが。。。

7.2 分類木

分類木とは質的変数（カテゴリなど）を応答変数（予測したい値、Y）とし、末端ノードの出力は回帰木のような平均ではなく、多数決でラベルを決定する。 分類木の構築は回帰木と同様、再帰的な分割を繰り返すことにより行います。しかしここで値を求める際の指標では分類誤り率（正しくカテゴリが予測されているか否か）ではなく、 Gini index や entropy 等が用いられることが多いです。単なる正解率で考えないのは多数決により分類木のラベルを決定するため、単純な正解率によるモデル評価はこの多数決で少数派となったラベルについての情報が考慮されなくなってしまうためです。

Gini inde と entropy

$\hat{p}_{mk}$ を領域 m に含まれるクラス k の事例の割合とすると、

Gini index

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

entropy

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$

枝刈りの結果、NO と NO、YES と YES に分岐するノードが存在するが、これは予測するクラスは同じではあるが、「どのくらいの確率で YES/NO か」が異なってくる。

7.3 決定木の長所短所

決定木の長所としては、以下の二点を上げることができます。

- モデルが実際に利用した条件分岐などを確認でき、またそれも極めて人間の行う意思決定の方法に近いため学習結果の解釈が容易。
- 質的変数を扱う際にダミー変数を作らなくてよい

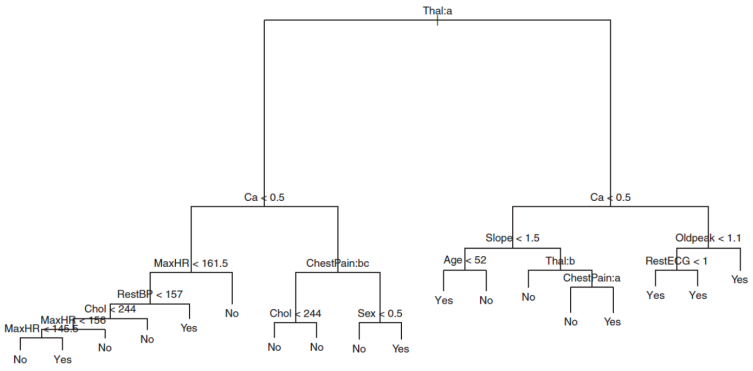

図 10　分類木による Heart データセット予測（枝刈り後）

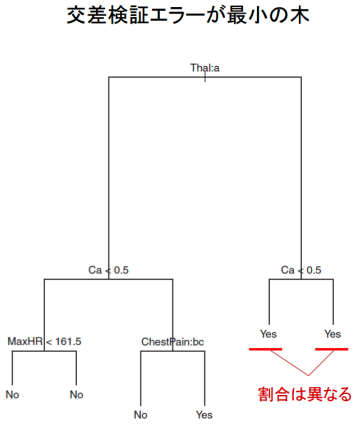

図 11　分類木による Heart データセット予測（枝刈り後）

逆に短所としては

- 回帰や分類の精度はあまり高くない（特殊なデータを除き、一般的な回帰モデル、分類モデルの方が高い精度を示すことが多い）
- 学習データにオーバーフィットしやすいため、学習データの小さな違いで構築される木が大きく変わることがある

例えば映画レビューをネガティブなものかポジティブなものか判断するとき、決定木モデルを用いるとある単語の有無で条件分岐を行うため、 とてもネガティブな内容ではあるものの、学習データのポジティブなメールに多く含まれていた単語が多用されているレビューに対しては正しく予測できない可能性がより高くなる。ただ決定木はアンサンブル学習の「ランファムフォレスト」、ブースティング等により制度の向上が可能で、以下では以下に決定機による予測精度を向上するかについて説明します。

7.4　バギング

バギングはブートルトラップ（一つ抜き法によるサンプリング）によってバリアンス（学習データからのずれ）を減らすことを目指す方法です。 この出力は回帰木の出力の平均になります。Out-of-bag (OOB) エラー推定が便利です。複数の決定木の平均を取ることにより、ここの決定木よりも精度が高くなります。

バギング (bagging)

手順は以下の通りになる。

1. 学習データから n 個の事例をランダムに復元抽出[a] し、抽出された n 個の事例を学習データとして決定木を構成する。
2. これを B 回繰り返す。
3. 得られた B 個の回帰木の出力の平均値が出力となる。分類木ならば平均を取る代わりに多数決で出力ラベルを決定する。

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x)$$

[a] 上の方で紹介された、一度取り出したサンプルデータをデータセットに再び戻すサンプリングの方法ですね。

また、バギングで用いられるエラー推定法を Out-of-bag エラー推定法といます。これは LOO 交差検証とほぼ同等の推定結果を交差検証を行うことなく得られるので、 効果的な推定法であると言えます。

Out-of-bag(OOB) エラー推定

7.5 ランダムフォレスト

バギングはシンプルに B 個の決定木を構成し、その平均を取っているシンプルなモデルでした。これだけでも単一の決定機よりは精度が出るのですが、これをさらに改善した手法に「決定木」というモデルがあります。 決定木はそのシンプルかつ高い精度を実現するモデルであり SVM 等と共に現実社会でも（割と）使われている機械学習モデルです。

ランダムフォレスト (random forest))

基本的な実行手順はバギングと同じ。ランダムフォレストでは、これに加えて決定木を構築するための再帰的な分割の際に考慮する予測変数を制限する。ランダムフォレストでは、各分割で考慮する予測変数を（分割ごとに）ランダムに選択された m 個の予測変数に限定する。$m = \sqrt{p}$ がよく用いられる。

ランダムフォレストで使用する入力変数を $m = p$ としたランダムフォレストモデルはバギングと考えることもできます。あえて入力変数を減らした方が複数の決定木のアンサンブルにおいて精度が向上する理由としては、 このように入力変数をランダムに選ぶことで、単純なバギングよりも構築される決定木のバリエーションが増え、構成された決定木の相関が減るためです。ちなみに分類の場合は $m = \sqrt{p}$ が一般的ですが、回帰の場合は $m = p/3$ が用いられることが多いらしいです。

バギングやランダムフォレストはシンプルな方法で精度を向上することが可能である一方、モデルの解釈が難しくなってしまう、またランダムフォレストの場合は分割によって RSS や Gini Index が減少してしまうという問題があります。回帰木におけるブースティングでは、小さな回帰木を多数作り、構築されたモデルによる予測結果と実際のデータとの残差を応答変数として回帰木を作ります。最終的なモデルの出力は構築された多数の回帰木の出力の和になります。

8 サポートベクターマシン (第 12 講)

8.1 サポートベクターマシンとは？

サポートベクターマシンは元々二値分類 [*15] のための識別モデルで、1990 年代に開発されその高い精度から現在でも様々な場面で活用されています。 一般的に「サポートベクターマシン」と総称することが多いのですが、

- 最大マージン分類器： マージンを最大化することによる分類。線形分離可能な場合以外は適応が難しい。
- サポートベクター分類器：線形分離可能でない場合でも適用可能だが、分離超平面は線形な関係に基づいている。
- サポートベクターマシン：カーネルにより非線形分類が可能であり線形分離の極めて難しい入り組んだデータでも対応は可能。

と実は細かく定義が分かれています [*16]。

8.2 最大マージン分類器

ある入力 X がそのクラスに所属するかを超平面 (hyperplane) により分割するものです [*17]。 超平面とは言ってはいますが、入力が二次元であれば直線、三次元であれば平面になります。

最大マージン分類器

p 次元空間中の超平面とは、p-1 次元の平坦な部分空間であり、以下の式で表される。

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$$

空間は超平面により超平面より大きいか小さいかにより、二つに区分される。

$$\begin{cases} \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p > 0 & (10) \\ \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p < 0 & (11) \end{cases}$$

では、このような超平面はどのように求められるのでしょうか。まず、以下のような $n \times p$ のデータ行列 X、$n \times 1$ のクラスラベル y から構成される学習データと、$1 \times p$ で構成されるテストデータ x^* を考えます。 つまり、p 個の入力変数をもつ n 個のデータとそれに対応した正解ラベルのペアを使って、新しい 1 個のテスト事例（同様に p 個の入力変数を持つ）を正しく分類する分類器をこれから構築していくことを考えます。

$$X = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1p} \\ x_{21} & x_{22} & \ldots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{np} \end{pmatrix}$$

[*15] クラスが二つの場合の分類問題

[*16] 実際に使用する場面ではほとんど区別していないと先生談

[*17] ちなみに SVM は若干概念的にこれまでのモデルよりややこしく感じるかもしれません。この資料とかがわかりやすいかも。 http://www.sakurai.comp.ae.keio.ac.jp/classes/infosem-class/2005/07SVM.pdf

$$Y = (y_1\ y_2\ \ldots\ y_n)^T \in \{-1, 1\}$$

$$\boldsymbol{x}^* = \begin{pmatrix} x_1^* \\ x_2^* \\ \vdots \\ x_p^* \end{pmatrix}$$

分離超平面

学習事例をすべて正しく分類する超平面を最大まー人分類器における分離超平面という。

$$\begin{cases} \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p > 0 \text{ if } y_i = 1 & (12) \\ \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p < 0 \text{ if } y_i = -1 & (13) \end{cases}$$

より、全ての i に対して $y_i(\beta_0 + \beta_1 x_{i2} + \ldots \beta_p x_{ip}) > 0$ が成立し、テスト事例の分類は x^* をこの分離超平面に代入した時の正負により行えれる。

つまり、とりあえず全ての学習事例を正しく分類できてれば良いため、分離超平面は無数に存在しうることになります。ただ学習事例を分離可能な複数な超平面の中にも、 少しノイズが入っただけで誤った分類をしてしまいそうな危うい超平面よりは、どちらのクラスよりもある程度離れている（つまり超平面が入力データのギリギリに惹かれているわけではない）超平面を選んだ方が、 新たな入力に対しても頑健なモデルになりそうです。

最も近い学習事例までの距離をマージン、超平面に最も近い事例をサポートベクターと言います。最大マージン分類器は、このマージンを最大にすることにより、様々な入力に対して適切に分類を行うことのできるモデルの構築を目指しています。

最大マージン分類器の構築

学習事例を $x_1, \ldots x_n$、それに対応したクラスラベル $y_1 \ldots y_n$ の最大マージン超平面は以下の最適化問題をとくことにより求められる。

$$\begin{aligned} &\underset{\beta_0, \beta_1, \ldots, \beta_p}{\text{Maximize M}} \\ &\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1 \\ &y_i(\beta_0 + \beta_1 x_{i2} + \ldots \beta_p x_{ip}) \geq M \forall i = 1, \ldots, n \end{aligned}$$

つまり、$\sum_{j=1}^{p} \beta_j^2 = 1$ の時、$y_i(\beta_0 + \beta_1 x_{i2} + \ldots \beta_p x_{ip})$ は超平面への距離を表しているため、全ての事例のうちこれが最も小さくなる事例（サポートベクター）におけるマージン M を最大にすることにより、超平面を求めています。(なぜこれで超平面までの距離を表すことができるかはスライドを参照してください。)

最大マージン分類器は以下のような問題点があります。

- そもそもデータが入り組んでおり、分離超平面を作ることができない、また分離超平面があった場合でも、学習データの変化に対して敏感に反応してしまうため、
- 与えるデータによって大きく異なるモデルとなってしまうという

8.3 サポートベクター分類器の構築

こういった最大マージン分類器の問題に対応することを目指しているのがサポートベクターマシンであり、マージンが小さく、設定されたマージンの内側に入ってしまうような事例や正しく判別されない事例を許容することで、データに大きく左右されないより頑健な [*18] モデルを構築することを目指しています。

サポートベクター分類器

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\varepsilon_1,\ldots,\varepsilon_n}{\text{Maximize M}}$$

$$\text{subject to } \sum_{j=1}^{p}\beta_j^2 = 1$$

$$y_i(\beta_0 + \beta_1 x_{i2} + \ldots \beta_p x_{ip}) \geq M(1-\varepsilon_i) \forall i = 1, \ldots, n$$

$$\varepsilon_i \geq 0,\ \sum_{i=1}^{n}\varepsilon_i \leq C$$

ε をスラック変数といい、$\varepsilon = 0$ ならば違反を許さず、この分類器は上述の最大マージン分類器と同地になりますが、$\varepsilon > 0$ ならば、ある事例がマージン M より ε 分だけ分離超平面に 接近していることを許すため、マージンの制約に違反していることがわかります。また、$\varepsilon > 1$、例えば ε を 2 に設定すると、$1 - \varepsilon = 1 - 2 = -1$ となり、クラス A に分類されるはずの事例が超平面の向こう側 (つまりクラス B) に分類されることを許容していることになります。 つまり ε を大きくすることでより許容度を上げていくことができるのですが、全て許容していてては逆に全く正しくない超平面となってしまうためある定数 C に制限し、違反を合計でどれだけ許すかを定めます。C の値は交差検証等で適切な値を探すことが多いのですが、この大小はモデルに対して以下のような影響を与えます。

C の値が大きい時

- C を大きくすると、超平面の反対側やマージンの内側に事例が存在することが多くなるため、サポートベクターの数が多くなる。
- 入力事例の変化には頑健になるためバリアンスは小さくできるが、バイアスは大きくなってしまう。

C の値が小さい時

- C が小さければモデルはより最大マージン分類器に近くなりサポートベクターの数は少なくなる。
- C が小さいと入力される事例の変化によって超平面がどんどん動くので、バリアンスは大きいが、今あるデータにはよくフィットするためバイアスは小さくなる。

ただサポートベクター分類器は許容度をあげてはいるものの、線形な境界に基づいた分類であるため完全に入り組んだ（螺旋状にデータが散らばっている等）の事例にはうまく対応ができません。次のサポートベクターマシンはこういったケースに対応するため非線形な境界による分類を可能にしています。

[*18] 余談ですがこういった入力データにいちいち左右されない頑健なモデルのことをロバスト (robust) なモデルと言います

8.4 サポートベクターマシン

来週の授業で喋るっぽいので来週追加します。

9 参考資料

授業で「参考」とされた内容や数式の計算過程等をまとめてあります。

9.1 リッジ回帰、Lasso 回帰を不等式制約付きの最適化問題として捉える

リッジ回帰、Lasso 回帰は以下のような不等式条件付きの最適化問題として捉えることができ、この見方をするとなぜ Lasso 回帰においてパラメータが 0 に収束しやすいのか 理解がしやすくなります。

不等式条件つき最適化問題としての等式化

リッジ回帰

$$\begin{aligned} \underset{\beta}{\text{minimize}} \quad & \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \\ \text{subject to} \quad & \lambda\sum_{j=1}^{p}\beta_j^2 \leq s \end{aligned}$$

Lasso 回帰

$$\begin{aligned} \underset{\beta}{\text{minimize}} \quad & \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \\ \text{subject to} \quad & \lambda\sum_{j=1}^{p}|\beta_j| \leq s \end{aligned}$$

9.2 予測誤差の大きさ（第 3 講）

予測結果 $\hat{Y}$ と実際のラベル Y の二乗誤差の期待値を計算してみる。統計的機械学習において、$Y = f(X)+\varepsilon$ で表されるため、

$$E[(Y-\hat{Y})^2] = E[((f(X)+\varepsilon) - \hat{f}(X))^2]$$

9.3 Bias-variance trade-off の導出（第 4 講）

9.4 線形単回帰のパラメータの推定法 (第 4 講)

線形単回帰の RSS を最小にする $\hat{\beta}_0, \hat{\beta}_1$ は、ぞれぞれで RSS を偏微分し、偏微分した結果を 0 にする $\hat{\beta}_0, \hat{\beta}_1$ を 連立方程式にとけば求めることができます。

$$\mathrm{RSS} = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 \ldots (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2$$

$$(y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 = y_1^2 + {\hat{\beta}_0}^2 + {\hat{\beta}_1}^2 x_1^2 - 2y_1\hat{\beta}_0 + 2\hat{\beta}_0\hat{\beta}_1 x_1 - 2\hat{\beta}_1 x_1 y_1 \text{より}$$

$$\begin{cases} \dfrac{\partial \mathrm{RSS}}{\partial \hat{\beta}_0} = 0 \iff \displaystyle\sum_{i=1}^{n} y_i - \beta_1 \sum_{i=1}^{n} x_i - \beta_0 n = 0 & (14) \\ \dfrac{\partial \mathrm{RSS}}{\partial \hat{\beta}_1} = 0 \iff \displaystyle\sum_{i=1}^{n} x_i y_i - \beta_1 \sum_{i=1}^{n} x_i^2 - \beta_0 \sum_{i=1}^{n} x_i = 0 & (15) \end{cases}$$

式 (3) より

$$\beta_0 = \frac{1}{n}\sum_{i=1}^{n} y_i - \beta_1 \sum_{i=1^n} x_i \tag{16}$$

ここで

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{17}$$

式 (5)(6) を式 (4) に代入し、β_1 について解くと、

$$\begin{aligned} \beta_1 &= \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n}(\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i)}{\sum_{i=1}^{n} x_i^2 - \frac{\sum_{i=1}^{n} x_i^2}{n}} \\ &= \frac{\sum_{i=1}^{n}(x - \overline{x})(y - \overline{y})}{\sum_{i=1}^{n}(x - \overline{x})^2} \end{aligned}$$

また式 (5) より

$$\beta_0 = \overline{y} - \beta_1 \overline{x}$$

9.5 RSS の性質の導出 (第 4 講)

式 (5) について、全て一つのシグマでくくると

$$\sum_{i=1}^{n} y_i - \beta_1 x_i - \beta_0 = \sum_{i=1}^{n} e_i = 0 \tag{18}$$

よって、RSS に関する一つ目の性質が示された。また同様に式 (6) より

$$\sum_{i=1}^{n} x_i y_i - \beta_1 x_i^2 - \beta_0 x_i = \sum_{i=1}^{n} x_i(y_i - \beta_1 x_i - \beta_0) = \sum_{i=1}^{n} e_i x_i = 0 \tag{19}$$

よって二つ目の性質も RSS からパラメータを推定する際に用いた式より導くことができる。

自由度	信頼係数：95%	信頼係数：99%
1	12.706	63.657
2	4.303	9.925
3	3.182	5.841
4	2.776	4.604
5	2.571	4.032
6	2.447	3.707
7	2.365	3.499
8	2.306	3.355
9	2.262	3.250

図 12　t 分布表の見方

9.6　標準誤差の求め方及び標準偏差 (第 4)

母平均の信頼区間 = 標本平均 $\pm t \times$ 標本標準偏差 $\div \sqrt{\text{標本数}}$ です。

$$\text{標準誤差 } SE = \frac{\text{標準偏差 } SD}{\sqrt{\text{標本数 } N}}$$

より、パラメータ β_0, β_1 の信頼区間はそれぞれ

$$\hat{\beta}_0 \pm t\mathrm{SE}(\hat{\beta}_0), \hat{\beta}_1 \pm t\mathrm{SE}(\hat{\beta}_1)$$

となる。t の値は t 分布表から t の値を求めることができる。信頼区間の当たる確率、すなわち 95% 信頼区間であれば 95% を信頼係数といい、 標本数から 1 を引いた数を自由度と呼び、t 分布表ではこの自由度と信頼係数より t の値を求めることができます。 例えば標本数が 10 で 95% 信頼区間を求めたい時、以下の表より t は自由度が 9、信頼係数が 95% の 2.262 ですことがわかる。 標本数が大きくなるとこの t の値はほぼ正規分布の値と変わらず 2 前後になるため、線形単回帰のパラメータの信頼区間の推定において近似的に $t = 2$ とすることができる。(ザックりな説明ですみません… https://blog.apar.jp/data-analysis/4632/)

9.7　縮小推定の手計算（第 10 講）

$n = p, \beta_0 = 0$

$$X = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

という問題に対して、パラメータ推定及び縮小推定（リッジ推定、Lasso 推定）を行う。

まず、最小二乗法でパラメータを推定する際には、最小二乗法を用いる場合、その目的関数は以下のように

なる。目的関数を最小化する β_i は目的関数の微分が 0 になる時となるため、

$$L = \sum_{j=1}^{p}(y_i - \beta_j)^2 \tag{20}$$

$$\frac{\partial L}{\partial \beta_i} = \frac{\partial}{\partial \beta_i}\sum_{j=1}^{p}(y_i - \beta_j)^2 = \frac{\partial}{\partial \beta_i}(y_i^2 - 2y_i\beta_i + \beta_i^2) = -2(y_i - \beta_i)$$

L に対して β_i の微分を取る時、i 番目以外のパラメータ β は無視してよいので、結果的に L を最小にする β_i の値は以下のようになる。

$$\beta_i = y_i \tag{21}$$

リッジ回帰においては目的関数が以下 (5) で表現されるため、同様に β_i で微分をし、L を最小化する β_i を求めると以下の (6) のようになる。

$$L = \sum_{i=1}^{n}(y_i - \beta_i)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 \tag{22}$$

$$\frac{\partial L}{\partial \beta_i} = \frac{\partial}{\partial \beta_i}\left(\sum_{j=1}^{p}(y_i - \beta_j)^2 + \lambda\sum_{j=1}^{p}\beta_j^2\right) + 2\lambda\beta_i = \frac{\partial}{\partial \beta_i}(y_i^2 - 2y_i\beta_i + \beta_i^2) = 2(1 + \beta_i) - 2y_i$$

$$\hat{\beta_i^R} = \frac{y_i}{1 + \lambda} \tag{23}$$

(6) より、$\lambda = 0$ ならば目的関数を最小化する β_i がパラメータ推定の結果求められた (5) と一致することがわかり、 $\lambda = 0$ の縮小推定の結果はパラメータ推定により求められる値と一致することが確認できる。

Lasso 回帰においては目的関数が以下 (7) で表現される。Lasso 回帰においては、正則化項が絶対値符号を含むため、β_i が正か負かで場合わけをする必要がある。

$$L = \sum_{i=1}^{n}(y_i - \beta_i)^2 + \lambda\sum_{j=1}^{p}|\beta_j| \tag{24}$$