EViews 11新機能 Elastic Net and LASSO

株式会社　ライトストーン

2019年10月

構成

- EViews 11の新機能であるElastic Net, LASSO, リッジ回帰について解説します
- この資料はEViews 11英語マニュアル, ユーザーズガイド 第36章 Elastic Net and LASSOを要約したものです
- 構成
  1. オーバーフィッティングと正則化
  2. クロスバリデーション
  3. LASSOによる推定例

オーバーフィッティングと正則化

- 既存の時系列データに対するフィットを良くするために多くの説明変数を使うような状況を考えます
- 時間の経過により新しい観測値が入手し，同じモデルを推定しフィットを行うと，フィットが格段に悪くなってしまうことがあります．このような場合，オーバーフィッティングが発生していると考えられます
- 正則化と呼ばれる数学的手法を用いて，このオーバーフィッティングの問題に対応することが可能です
- パラメータを推定する際にパラメータ自体の大きさに罰則を掛ける考え方を利用します

オーバーフィッティングと正則化

- 線形回帰モデルの費用関数として次のような式を考えます

$$J = \frac{1}{2m}\sum_{i=1}^{m}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\left[\frac{(1-\alpha)}{2}\sum_{j=1}^{p}\beta_j^2 + \alpha\sum_{j=1}^{p}\left|\beta_j\right|\right] \tag{1}$$

- 罰則パラメータ λ から始まる第二項における α の値によってリッジ回帰, LASSO, Elastic Net と呼び方が変わります
- 罰則パラメータ λ はゼロ以上の値をとります. m はサンプルサイズ, p は説明変数の数です
- J を最小化するような β を求めます. この時, β をなるべく小さくするようなメカニズムが働きます

リッジ回帰

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

- λが大きいほど, βは小さくなります
- λの選択が重要な役割を果たすことが直感的に分かります
- λの決定にはクロスバリデーションと呼ばれる手法を利用します

クロスバリデーション

- データをパラメータ推定用の training 部と予測誤差を計測する test 部に分けます
- 様々な λ の値を使って予測誤差を求め，最も予測誤差が小さくなるものを $\lambda_{\min}$ とします
- 推定ダイアログの Lambda のテキストボックスに候補となる値を 1 つ，または，複数個入力します
- 設定を EViews にまかせる場合はブランクのままにします

クロスバリデーション

- 4種類の方法が用意されています
- Suffle: ダイアログボックスでtraing部とtest部の割合を決めます. Shffle repsで設定した回数分, traing部とtest部のデータの順番を変更(Shffle)してパラメータを推定し, 予測誤差の最小化を計ります
- K-Fold:K個の均等な数のデータセットを作成します. そのうちの1つをtest部に, 残りのK-1個をtraing部とします
- Leave One Out: K-Foldと原理は同じですが, Kをデータの個数とします
- Leave P Out:K-Foldと原理は同じです. test部のデータは保持して, 残りのデータでtraing部を繰り返し作成します

バイアスと分散のトレードオフ

- 推定量の期待値と推定値の差異をバイアス, 推定値の不確実性を分散と呼ぶことにします
- 正則化を利用する目的はトレーニングデータでのフィットの良さを維持したままモデルの複雑さを取り除き, 予測力を維持することです
- バイアスが少なければ, トレーニングデータへのフィットは良くなります
- 一方, 分散が小さければ, 簡単なモデルでも予測力は優れています
- 最小二乗法の回帰分析では説明変数の数が増えれば, より複雑なモデルになります(オーバーフィッティングのリスクが高くなります)

バイアスと分散のトレードオフ

- **説明変数を減らすと分散も小さくできますが**, **バイアスが生じる可能性が高まります**
- **このことをバイアスと分散のトレードオフと呼びます**
- Elastic Net, **リッジ回帰**, LASSO **では係数を小さくすることで**, **モデルの複雑さを取り除く**, **というのが基本的な考え方です**

LASSO

- LASSO(Least Absolute Shrinkage and Selection Operator)

$$J = \frac{1}{2m}\sum_{i=1}^{m}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}\left|\beta_j\right| \tag{3}$$

- **リッジ回帰とは異なり，係数は小さくなるだけでなく，ゼロにもなりえます**
- **係数の和に対してパラメータ λ を 1 つだけ推定します**

Elastic Net

- Elastic Net の費用関数はリッジ回帰 $(\alpha = 0)$ と LASSO$(\alpha = 1)$ の組合せとなっています

$$J = \frac{1}{2m} \sum_{i=1}^{m} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \left[\frac{(1-\alpha)}{2} \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} \left| \beta_j \right| \right]$$

- 説明変数間にある程度の相関が認められる場合に，よいモデルを提供します

LASSOによる推定例

- EViews 11のLASSOによるモデル推定の機能を試してみましょう
- サンプルデータはprostate.wf1です．Help/Quick Help Reference/Sample Programs & Data と操作します
- EViews 11 Manual Dataの項目をクリックします．Chapter 36のフォルダに目的のファイルがあります

LASSOによる推定例

- 目的のファイルを開くと，すでに推定式のオブジェクトlasso，ols，ridgeの3つが用意されていることが分かります
- ここではlassoのオブジェクトを開いて推定結果を確認します
- 推定結果の画面について説明します．罰則パラメータλについて minimumの値が手元のファイルでは0.01274になっています
- 乱数を利用しますので，再推定すると若干異なることが考えられます
- 正則化の考え方にしたがうLASSOアルゴリズムにおいて，λがこの値をとる場合に予測誤差は最小化されます
- 予測誤差を若干，大きくした場合(+1SEと+2SE)の推定値を合わせて表示します

LASSOによる推定例

- View/Lambda Coefficient と操作すると, λ の大きさによって係数の変化する様子が分かります
- λ を大きくしていくと, パラメータが次々にゼロに収束することが分かります
- +1SEの0.157を採用すると, モデルの複雑さはかなり削除され, 変数は3つしか残りません.

まとめ

- 説明変数を数多く利用するモデルでは，観測値の個数が増減するだけで，フィットが極端に悪くなるオーバーフィッティングが発生する場合がある
- 費用関数にペナルティを課すことで，この問題を回避する
- Elastic Net は LASSO とリッジ回帰の中間に位置する
- 罰則パラメータ λ によって推定結果は左右される．よって，推定後に λ とパラメータの関係を考察する必要がある