

クレジット:

UTokyo Online Education 統計データ解析Ⅱ 2018 小池祐太

ライセンス:

利用者は、本講義資料を、教育的な目的に限ってページ単位で利用することができます。特に記載のない限り、本講義資料はページ単位でクリエイティブ・コモンズ 表示-非営利-改変禁止 ライセンスの下に提供されています。

<http://creativecommons.org/licenses/by-nc-nd/4.0/>

本講義資料内には、東京大学が第三者より許諾を得て利用している画像等や、各種ライセンスによって提供されている画像等が含まれています。個々の画像等を本講義資料から切り離して利用することはできません。個々の画像等の利用については、それぞれの権利者の定めるところに従ってください。



# 統計データ解析 (II) 第 13 回

小池祐太

2018 年 7 月 12 日

## 1 時系列解析

## 2 クラスター分析

- 目的
- $k$ -平均法

# 時系列解析

## ● 時系列データ

- ▶ 時間軸に沿って観測されたデータ
  - ▶ 観測の順序に意味があることや, 異なる時点間での観測データの間の従属関係が重要であることが特徴
- 統計学では, 時系列データは**確率過程** (時間を添え字として持つ確率変数列:  $X_t, t = 0, 1, \dots, T$  (あるいは  $t = 1, \dots, T$ )) によってモデル化する

# ホワイトノイズ

- 平均 0, 分散  $\sigma^2$  で互いに無相関な確率変数列からなる確率過程を**ホワイトノイズ**  $WN(0, \sigma^2)$  と呼ぶ
- 平均 0 で有限の分散を持つ同一の分布に従う独立な確率変数列がホワイトノイズの典型的な例
- このことから, ホワイトノイズのシミュレーションには乱数発生器 (`rnorm()` や適当な自由度の `rt()` など) で行うことが多い

# 自己回帰平均移動モデル (ARMA モデル)

- $\epsilon_t$ ,  $t = \max\{p, q\} + 1, \dots, T$  を  $WN(0, \sigma^2)$  とする
- $a_1, \dots, a_p, b_1, \dots, b_q$  を定数とする.  $X_1, \dots, X_{\max\{p, q\}}$  が初期値として与えられたとき,

$$X_t = a_1 X_{t-1} + \dots + a_p X_{t-p} + b_1 \epsilon_{t-1} + \dots + b_q \epsilon_{t-q} + \epsilon_t, \\ t = \max\{p, q\} + 1, \dots, T$$

で帰納的に定まる確率過程のモデルを**次数  $(p, q)$  の自己回帰平均移動モデル**もしくは**ARMA( $p, q$ ) モデル**と呼ぶ

- ARMA( $p, 0$ ) モデルを単に AR( $p$ ) モデルと呼び, ARMA( $0, q$ ) モデルを単に MA( $q$ ) モデルと呼ぶ
- ARMA モデルは単純な形ながら異なる時点間の観測データの従属構造を柔軟に記述できるため, 基本的な時系列モデルとして広く利用されている

## (弱) 定常性

- 確率過程  $X_t$ ,  $t = 1, \dots, T$  が次の 2 つの性質をもつとき, **(弱) 定常**であるという:
  - (i)  $X_t$  の平均は時点  $t$  によらない.
  - (ii)  $X_t$  と  $X_{t+h}$  の共分散は時点  $t$  によらず時差  $h$  のみで定まる. 特に,  $X_t$  の分散は時点  $t$  によらない ( $h = 0$  の場合を考えればよい).
- 定常でない確率過程は**非定常**であるという. 前節で説明した確率過程の定常性は以下のようにまとめられる.
  - ▶ **定常過程** ホワイトノイズ, MA モデル
  - ▶ **非定常過程** トレンドのあるホワイトノイズ, ランダムウォーク
  - ▶ **定常にも非定常にもなりうる確率過程** AR モデル, ARMA モデル

## (弱) 定常性

- 非定常過程は平均や分散といった基本的な統計量が時間によって変動してしまうため扱いが難しい
- そのため, 非定常過程の分析の際には対数変換や階差をとる変換等によって定常過程とみなせるように変換したあと分析を実行することが多い

## 自己共分散・自己相関

- $X_t$ ,  $t = 1, \dots, T$  が定常過程の場合, その定義から  $X_t$  と  $X_{t+h}$  の共分散は時点  $t$  によらずラグ  $h \geq 0$  のみで定まる
  - ▶ この共分散をラグ  $h$  での**自己共分散**と呼ぶ
- また, 分散も時点によらないので,  $X_t$  と  $X_{t+h}$  の相関も時点  $t$  によらずラグ  $h \geq 0$  のみで定まる
  - ▶ この相関をラグ  $h$  での**自己相関**と呼ぶ
- 通常, ラグ  $h$  での自己共分散を観測データ  $X_1, \dots, X_T$  から推定するには, 標本自己共分散

$$\frac{1}{T} \sum_{t=1}^{T-h} (X_t - \bar{X})(X_{t+h} - \bar{X})$$

を用いる. ただし,  $\bar{X} = \frac{1}{T} \sum_{t=1}^T X_t$  は標本平均である

## 自己共分散・自己相関

- 同様に, ラグ  $h$  での自己相関を観測データ  $X_1, \dots, X_T$  から推定するには, 標本自己相関

$$\frac{\sum_{t=1}^{T-h} (X_t - \bar{X})(X_{t+h} - \bar{X})}{\sum_{t=1}^T (X_t - \bar{X})^2}$$

を用いる

- 自己共分散および自己相関は, 異なる時点間での観測データの従属関係を要約するための最も基本的な統計量である
- R では関数 `acf()` によって計算できる

# ARモデルのあてはめ

- Rには、与えられた時系列データに適切な定常ARモデルをあてはめるための関数 `ar()` が用意されている
- 実行例 `ar-estimation.r`

# ARMA モデルのあてはめ

- 指定された次数の定常 ARMA モデルを与えられた時系列データにあてはめるための関数として `arima()` が用意されている
- ただし, AR モデルをあてはめるための関数 `ar()` と異なり, 関数 `arima()` には適切な次数を決定する機能は備わっていない
- そのため, 次数の決定は試行錯誤で行うか, パッケージ `forecast` の関数 `auto.arima()` を利用するとよい
- 実行例 `arma-estimation.r`

# 予測

- 推定されたモデルを使って数期先の時系列データの値を予測するには関数 `predict()` を使う
- $n$  期先までのデータを予測する場合, オプション `n.ahead` に  $n$  を指定すればよい
- 実行例 `ts-predict.r`

# クラスター分析

## ● クラスター分析 (cluster analysis)

- ▶ 主成分分析と並ぶ教師なし学習の代表的な手法の一つ
- ▶ 多数の個体に対するいくつかの共変量 (特徴量) の観測データが与えられたとき, それらの個体の間に隠れているクラスター構造 (グループ構造) を共変量の値に基づいて発見することを目的とする分析手法
- ▶ 同じクラスターに属する個体どうしは (なんらかの意味で) 近い性質をもち, 異なるクラスターに属する個体どうしは異なる性質をもつような少数のクラスターを見いだすことで, さらなるデータ解析やデータの可視化に役立てる目的で利用される

# クラスター分析

- クラスター分析には大きく分けて以下の2つのアプローチがある:
  - ▶ **階層的方法**  
データ点およびクラスターの間 (共変量から定まる) 距離 (非類似度) を定義し, 近いものから順にクラスターを形成, もしくは近いものどうしがクラスター内に残るように分割しながら, グループ化していく方法
  - ▶ **非階層的方法**  
クラスターの定め方の「良さ」を評価するための損失関数を定め (値が小さいほど良いとする), その損失関数を最小化するようにクラスターを形成して変数をグループ化していく方法
- 本講義では非階層的手法の代表的方法である  **$k$ -平均法 ( $k$ -means clustering)** について説明
- 階層的クラスタリングの一例としては関数 `hclust()` がある (ヘルプファイルを参照のこと)

## k-平均法

- $p$  個の変数  $X_1, X_2, \dots, X_p$  を  $n$  個の個体について観測した観測データ  $x_{i1}, x_{i2}, \dots, x_{ip}$  ( $i = 1, 2, \dots, n$ ) が与えられているとする
- $i$  番目の個体に対する観測データに対応するベクトルを

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})$$

とする

- クラスターの定め方は、各個体番号  $i = 1, 2, \dots, n$  に対してその個体が属するクラスター番号  $C(i)$  を定める対応  $C$  として定式化できる
- 非階層的クラスタリングでは、このような対応  $C$  について、
  - ▶ 対応  $C$  の「良さ」を評価する損失関数を観測データ  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$  を決める
  - ▶ その損失関数を  $C$  に関して最小化するようにクラスターを定めるという手順を踏むことで実行できる

## k-平均法

- k-平均法では、最終的に得たいクラスターの個数  $k$  をあらかじめ指定する
- また、2つの個体  $i, i'$  の「近さ」を共変量の観測データ間のユークリッド距離の二乗

$$\|\mathbf{x}_i - \mathbf{x}_{i'}\|^2 := \sum_{j=1}^p (x_{ij} - x_{i'j})^2$$

で評価する

- そして、同じクラスターに属する個体どうしが近いほど値が小さくなるように、 $C$  の損失関数  $W(C)$  を定める

# k-平均法

- 具体的には

$$W(C) := \sum_{l=1}^k \frac{1}{n_l} \sum_{i:C(i)=l} \sum_{i':C(i')=l} \|\mathbf{x}_i - \mathbf{x}_{i'}\|^2$$

と定義する

- ▶  $n_l$  は  $l$  番目のクラスターに属する個体の総数
- いま,  $l$  番目のクラスターに属する個体の特徴量の共変量の平均を

$$\bar{\mathbf{x}}_l := \frac{1}{n_l} \sum_{i:C(i)=l} \mathbf{x}_i$$

で定める

## k-平均法

- このとき, 簡単な計算によって  $W(C)$  は次のように書き直せる:

$$W(C) = 2 \sum_{l=1}^k \sum_{i:C(i)=l} \|\mathbf{x}_i - \bar{\mathbf{x}}_l\|^2$$

- 従って,  $W(C)$  を最小化するように  $C$  を定めることは, クラスタ内変動の総和が最小になるようにクラスタを定めることと同等

- 我々の目的は  $W(C)$  が最小になるように  $C$  を定めること
- $C$  の取り方は  $k^n$  通りで有限個のパターンしかないので, 原理的にはこの  $k^n$  通り全てのパターンについて  $W(C)$  の値を計算し比較することで,  $W(C)$  を最小化する  $C$  が決定できる
- しかし, サンプル数  $n$  が十分小さくない限り, この方法は計算量の観点から現実には実行が不可能
- そのため, 現実的な計算量で実行可能であるような  $W(C)$  の最小化のためのアルゴリズムがいくつか提案されているが, 代表的なものとして **Lloyd-Forgy のアルゴリズム**がある
- Lloyd-Forgy のアルゴリズムでは,  $\bar{x}_l$  が

$$\sum_{i:C(i)=l} \|x_i - \mu_l\|^2$$

を最小化するような  $p$  次元ベクトル  $\mu_l$  と一致することに着目する

# k-平均法: Lloyd-Forgy のアルゴリズム

## ● Lloyd-Forgy のアルゴリズム

$C$  と  $\mu_1, \mu_2, \dots, \mu_k$  の更新を以下の手順で繰り返すことで  $W(C)$  を最小化する  $C$  を求める:

1.  $p$  次元ベクトルの初期値  $\mu_1, \mu_2, \dots, \mu_k$  を与える
2. 各データ点  $i = 1, 2, \dots, n$  について,  $\|x_i - \mu_l\|$  を最小化するような  $l$  を  $i$  が所属するクラスター番号  $C(i)$  として定める
3. 各  $l = 1, 2, \dots, k$  について, ベクトル  $\mu_l$  を

$$\mu_l = \frac{1}{n_l} \sum_{i:C(i)=l} x_i$$

によって更新する

4. 平均ベクトルが更新前と更新後で変化しなかった場合計算を終了する。そうでなければステップ 2 に戻る

## k-平均法: Lloyd-Forgy のアルゴリズム

- Lloyd-Forgy のアルゴリズムの成否は初期値のベクトル  $\mu_1, \mu_2, \dots, \mu_k$  の選び方に依存する
- 応用上は複数の初期値の候補をランダムに試して,  $W(C)$  の値を最も小さくする解を最終的な解として採用するということが行われる

## k-平均法: Rでの実行

- Rにはk-平均法を実行するための関数 `kmeans()` が実装されている。
  - ▶ クラスターの数  $k$  はオプション `centers` で指定する。
  - ▶ オプション `algorithm` では  $W(C)$  を最適化するために利用するアルゴリズムが指定できる。デフォルトではLloyd-Forgyのアルゴリズムの改良版であるHartigan-Wongのアルゴリズムを利用する。
  - ▶ オプション `nstart` では試す初期値の候補の数が指定できる。
- なお,  $W(C)$  の定義から明らかのように, 共変量のうちの1つを定数倍 (例えば測定値の単位を変更) すると, クラスタリングの結果が変わりうることに注意する必要がある
- すべての共変量を同じスケールで評価したい場合は, 主成分分析の場合と同様に, 実行前にデータを標準化すればよい
- 実行例 `kmeans.r`