# 目次

# 第4章

# 時系列データのモデリングと行列計算

## 4.1 概要

この章では時系列データ分析の入門的な話題を紹介するとともに Python で行列計算を行う方法を解説する。

最初にお断りをしておくが，この講義では，洗練された統計学を使った高度な統計分析を行わない。データを可視化したり，データの背後にある傾向を探す作業を体験してもらうのが目的だ。本格的な分析を始める前にデータを眺めるプロセスだと考えてほしい。

本章の理論パートでは，経済理論から離れて，データの見方やデータの背後にある構造を知るためのツールを学ぶ。非常に簡単なモデルだけ紹介するので，より進んだ学習をしたい読者は「時系列分析」をキーワードに参考資料を探すとよい。高校までで学んだ数学や大学初年次の線形代数がどのように役立つかを紹介したい。すぐに何をやっているか分からなくてもいいので，一通り目を通してほしい。

プログラミング・パートでは，時系列モデルのシミュレーションについて説明する。連鎖方式の指数計算で用いたテクニックが使われていることを意識しながら手を動かしてほしい。最後に，データからパラメータを推測する方法を簡単に紹介する。

## 4.2　理論

### 4.2.1　簡単な時系列モデル: 高校数学の復習

読者が日本で高校を卒業していれば，文理問わず時系列モデルの原型を学んでいるはずだ。それは「漸化式」と呼ばれる次のような式だ。

$$a_{n+1} = pa_n + q \tag{4.1}$$

(4.1) をどのように解いたかを思い出してみよう。まず「特性方程式」という次の方程式を立てる。

$$a = pa + q \tag{4.2}$$

そして $a$ について解く。さしあたり $p \neq 1$ として，

$$a = \frac{q}{1-p}$$

である。(4.1) と (4.2) の差を取ると，

$$a_{n+1} - a = p\,(a_n - a)$$

これは，$a_n - a$ という形が繰り返し現れることに注意すると，

$$\begin{aligned} a_n - a &= p\,(a_{n-1} - a) \\ &= p^2\,(a_{n-2} - a) \\ &= p^3\,(a_{n-3} - a) \\ &= \cdots \\ &= p^n\,(a_0 - a) \end{aligned}$$

$a_0$ が既知の情報（初期条件）とすれば，すべての $n = 1, 2, \ldots$ について

$$a_n = a + p^n\,(a_0 - a) \tag{4.3}$$

という公式が得られる。この作業を「漸化式を解く」と学んだ。

ここで，$n \to \infty$ についての極限挙動について確認しておこう。上で計算した $a$ は

$$a = \cdots = a_{n+1} = a_n = a_{n-1} = \cdots$$

が成り立つ数字であった。初期条件がたまたま $a_0 = a$ を満たしていれば，漸化式の解 $a_n$ はずっと $a$ であり続ける。もし $a_0 \neq a$ であっても，$|p| < 1$ であれば $\lim_{n\to\infty} p^n = 0$ すなわち $\lim_{n\to\infty} a_n = a$ が成り立つ。$a$ は漸化式の長期的な挙動を代表する重要な値であり，**定常状態**とか**平衡点**と呼ばれる。$|p| > 1$ であれば，例外的な $a_0 = a$ のケースを除いて，$\lim_{n\to\infty} |a_n| = \infty$ となる。

(4.3) をもう少し変形しておこう。

$$\begin{aligned} a_n &= a + p^n (a_0 - a) \\ &= p^n a_0 + (1 - p^n) a \\ &= p^n a_0 + (1 - p^n) \frac{q}{1-p} \\ &= p^n a_0 + \left(1 + p + p^2 + \cdots + p^{n-1}\right) q \end{aligned} \tag{4.4}$$

$a_n$ の決定には 2 つの要因が絡んでいる。

1. 1 つは初期値 $a_0$ の影響である。$p^n$ は「$n$ 期の過去の情報が現在に与える影響」を取り出す効果があるので，$p^n a_0$ は「($n$ 期から見て $n$ 期前であるところの) 0 期の情報 $a_0$ が $a_n$ に与える影響」となっている。
2. もう 1 つは，毎期毎期新たに加算される $q$ の影響である。こちらは 0 期前 ($p^0 = 1$)，1 期前 ($p^1 = p$)，$\ldots$，$(n-1)$ 期前の $q$ がすべての $a_n$ に効いている。

なお，具体的な $a_n$ の値を求めたいだけなら漸化式を解く必要はない。式 (4.1) を順番に計算していけばよいからだ。しかし，「漸化式を解く」ことで数値計算によらずとも長期的な振る舞いを理解できるようになる。収束・発散の境界は等比数列の拡大率 $p$ の絶対値が 1 のところにある。

### 4.2.2　AR(1) 過程

時系列分析の文脈では，

$$y_t = a_0 + a_1 y_{t-1}$$

という方程式をベースとしたモデルがよく用いられる。なんらかの変数（$y$）の時点 $t$ における観測値 $y_t$ が前期の観測値 $y_{t-1}$ に依存することを表している。$a_0$,$a_1$ は定数，$y_1, y_2, \ldots$ が分析対象である。形式的に (4.1) と同じであることが分かるだろう。上の方程式に，事前には予測できない不確実性（$\varepsilon_t$）を加えたものが時系列分析の基礎モデルである。

$$y_t = a_0 + a_1 y_{t-1} + \varepsilon_t \tag{4.5}$$

方程式 (4.5) で表現される $y$ は**「AR(1) 過程**に従う」と言われる。単に $y$ は AR(1) 過程である，と言ってもよい。AR は autoregressive の略で「自己回帰」と訳す。過去の自分自身のデータ（$y_{t-1}$）を説明変数，今の自分（$y_t$）を被説明変数とした回帰方程式になっている。

上で導入した $\varepsilon_t$ について，「時点 $t-1$ にいる分析者は何の情報も持っていない」ことを前提とする。もし情報を持っているならそれはモデルを改善するために使っているはずだからだ。この「時点 $t-1$ にいる分析者は何の情報も持っていない」ということを，数学的には

$$\mathbb{E}_{t-1}[\varepsilon_t] = 0$$

と表現する。$\mathbb{E}_{t-1}[\cdot]$ は条件付き期待値を表す記号である。確率論にアレルギーを持っている人もいるかも知れないが，あまり難しく考えなくてもよい。モデル外の情報（$\varepsilon_t$）は $y_t$ を増やすのか減らすのかも分からないはずだから，ゼロにならなければならないというだけのことだ。通常 $\varepsilon_t$ は，$\varepsilon_t$ と $\varepsilon_k$ $(t \neq k)$ が独立で，平均ゼロ，$\mathrm{Var}(\varepsilon_t)$ = 一定が仮定される（ホワイト・ノイズの仮定）。次のことに注意する。

$$\mathbb{E}_{t-1}[y_{t-1}] = y_{t-1}, \quad \mathbb{E}_{t-1}[a_0] = a_0$$

$t-1$ 時点にいる観測者は $y_{t-1}$ をすでに観測しているし，定数 $a_0$ についてはずっと前から知っているのだ。(4.5) の全体に $\mathbb{E}_{t-1}[\cdot]$ を付けると[1]，

$$\begin{aligned}\mathbb{E}_{t-1}[y_t] &= \mathbb{E}_{t-1}[a_0 + a_1 y_{t-1} + \varepsilon_t] \\ &= \mathbb{E}_{t-1}[a_0] + a_1 \mathbb{E}_{t-1}[y_{t-1}] + \mathbb{E}_{t-1}[\varepsilon_t] \\ &= a_0 + a_1 y_{t-1}\end{aligned}$$

となる。つまり，AR(1) モデルは，今期 $(t-1)$ の情報を用いて次期 $(t)$ に観察されそうな値（期待値）を決定するモデルである。

### 4.2.3　AR(2) 過程

AR(1) があれば AR(2) も AR(3) もある。一般に AR($p$) モデルと呼ばれる時系列モデルは

$$y_t = a_0 + a_1 y_{t-1} + \cdots + a_p y_{t-p} + \varepsilon_t \tag{4.6}$$

と書ける。以下では，AR(2) 過程と漸化式の関係，行列形式の表現を見ていこう。

#### 高校時代の解きかた（線形代数を使わない）

次の形式の漸化式を解く方法を覚えているだろうか。もう忘れてしまった人もいるかもしれないが，これも高校数学で履修済みのはずだ。

$$a_{n+2} = p a_{n+1} + q a_n + r$$

定数項 $r$ が鬱陶しいので，最終的に定常状態に収束したらどうなるかな，と考えて次の方程式を解く。

$$a = pa + qa + r \Longrightarrow a = \frac{r}{1-p-q}$$

[1] 期待値の線形性 $\mathbb{E}_{t-1}[\alpha x + \beta y] = \alpha \mathbb{E}_{t-1}[x] + \beta \mathbb{E}_{t-1}[y]$ となることは認めよう。心配な読者は適当な確率論のテキストを参照してほしい。

である。$b_n = a_n - a, n = 0, 1, 2, 3, \ldots$ として，

$$b_{n+2} = pb_{n+1} + qb_n$$

という定数項のない漸化式になる。これを解くときに，誰かが次のような形に変形することを閃いたらしい。

$$b_{n+2} - \alpha b_{n+1} = \beta(b_{n+1} - \alpha b_n)$$

これを満たす $\alpha, \beta$ を見つけ出そうという訳である。少し変形すると，

$$b_{n+2} - (\alpha + \beta)b_{n+1} + \alpha\beta b_n = 0$$

これは

$$p = \alpha + \beta, \quad q = \alpha\beta$$

なのだから，2 次方程式

$$x^2 - px + q = 0$$

の解によって $\alpha, \beta$ を決定できる（俗に「解と係数の関係」と呼ばれる関係）[2]。したがって，最初の 2 項 $b_0, b_1$ が既知であれば，

$$b_{n+1} = \alpha b_n + \beta^n(b_1 - \alpha b_0)$$

のようにして，一般の $b_n$ あるいは $a_n$ を決定できる。$\lim_{n\to\infty} |b_n|$ が $|\beta|$ に関係していることは明白だろう。しかし，右辺 1 項目 $b_{n+1} = \alpha b_n + \cdots$ に注目すると，$|\alpha|$ も無関係ではなさそうだ。実際，$|\alpha| < 1$ かつ $|\beta| < 1$ が成り立っているときに，$\lim_{n\to\infty} |b_n| = 0$ が成り立つ。これは線形代数学の知識を使うと明快に示すことができる。

[2] ここでは $\alpha, \beta$ が実数で $\alpha \neq \beta$ を満たすことを想定している。

**線形代数を使う**

上の解法は多次元に一般化することは難しい。より実用的な方法を理解するには線形代数学の知識が必要になる。

$$a_{n+2} = pa_{n+1} + qa_n + r$$

次の行列型の漸化式に変形する。

$$\begin{bmatrix} a_{n+2} \\ a_{n+1} \end{bmatrix} = \begin{bmatrix} p & q \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_{n+1} \\ a_n \end{bmatrix} + \begin{bmatrix} r \\ 0 \end{bmatrix},$$

ここの行列漸化式は次の連立方程式と同等である。

$$\begin{cases} a_{n+2} = pa_{n+1} + qa_n + r \\ a_{n+1} = a_{n+1} \end{cases}$$

さらに表現を簡略化するために次のような記法を導入する。

$$\boldsymbol{a}_n = \begin{bmatrix} a_{n+1} \\ a_n \end{bmatrix}, \boldsymbol{A} = \begin{bmatrix} p & q \\ 1 & 0 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} r \\ 0 \end{bmatrix}$$

すると,

$$\boldsymbol{a}_{n+1} = \boldsymbol{A}\boldsymbol{a}_n + \boldsymbol{c} \tag{4.7}$$

と書ける。自明な等式を導入することで表現をシンプルにすることができた。式 (4.4) と同じスピリットで一般の $\boldsymbol{a}_n$ を表現してみよう。

$$\begin{aligned} \boldsymbol{a}_n &= \boldsymbol{A}\boldsymbol{a}_{n-1} + \boldsymbol{c} \\ &= \boldsymbol{A}\left(\boldsymbol{A}\boldsymbol{a}_{n-2} + \boldsymbol{c}\right) + \boldsymbol{c} \\ &= \boldsymbol{A}^2\boldsymbol{a}_{n-2} + \left(\boldsymbol{A} + \boldsymbol{I}\right)\boldsymbol{c} \\ &= \cdots \\ &= \boldsymbol{A}^n\boldsymbol{a}_0 + \left(\boldsymbol{A}^{n-1} + \boldsymbol{A}^{n-2} + \cdots + \boldsymbol{A} + \boldsymbol{I}\right)\boldsymbol{c} \end{aligned} \tag{4.8}$$

この表現は一般の $\boldsymbol{A}$,$\boldsymbol{c}$ に対して正しい公式になっているが，極限挙動 $\lim_{n\to\infty} \boldsymbol{a}_n$ については何も教えてくれない。$\boldsymbol{A}^n$ の極限での振る舞いについて調べる必要があるのだが，これには行列の対角化，ジョルダン標準化という考え方を使う。詳細は省略するが，

$$
\begin{aligned}
&\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^{-1} \\
&\boldsymbol{D} = \text{対角成分に } \boldsymbol{A} \text{ の固有値が並ぶ上三角行列} \\
&\quad = \begin{bmatrix} \alpha & * \\ 0 & \beta \end{bmatrix}
\end{aligned}
$$

という形式で表現できて，$\boldsymbol{D}^n$ の成分は比較的簡単に計算できる。$\alpha$,$\beta$ は前節のものと同じと考えてもよいが，実数に限らず複素固有値も許容される。もし $\boldsymbol{A}$ のすべての固有値の絶対値が 1 未満であれば，$\lim_{n\to\infty} \boldsymbol{D}^n = \boldsymbol{0}$。さらにそこから $\lim_{n\to\infty} \boldsymbol{A}^n = \boldsymbol{0}$ を導くことができる。固有値に絶対値が 1 より大きいものがあれば，$\boldsymbol{A}^n$ は発散する。固有値に絶対値が 1 と等しいものがある場合には少し難しいのだが，発散せずに一定の大きさを保ち続けるケースと発散するケースがある。極限での振る舞いに基づいて，固有値を分類しておこう。

- 絶対値が 1 未満の固有値を**安定固有値**という。
- 絶対値が 1 より大きい固有値を**不安定固有値**という。
- 絶対値が 1 に等しい固有値を**中心固有値**という。

高校数学で $b_{n+2} - \alpha b_{n+1} = \beta\,(b_{n+1} - \alpha b_n)$ を考えたのは，固有値に対応する固有空間という概念に対応している。$-1 < \beta < 1$ のときには $n \to \infty$ の極限で $(b_n, b_{n+1})$ は $y = \alpha x$ を満たす直線に漸近する。この直線が固有空間の 1 つである（固有値 $\alpha$ に対応する固有空間）。行列表現した上記の $\boldsymbol{V}$ に関連しているのだが，きちんとした説明にはそれなりの準備が必要なので，関心のある読者は適当な線形代数学や力学系の教科書で勉強してほしい。

重要な性質を命題の形でまとめておこう。

**命題 4.1.** 行列 $\boldsymbol{A}$ が安定固有値のみをもつとき，

$$\lim_{n\to\infty} \boldsymbol{A}^n = \boldsymbol{0}$$

が成り立つ。 □

なお，行列の極限は成分ごとに考えればよい。

安定固有値のみを持つ行列については，無限等比級数の公式（$|\rho| < 1$)

$$\sum_{n=0}^{\infty} \rho^n = 1 + \rho + \rho^2 + \cdots = \frac{1}{1-\rho}$$

と同等の公式が成り立つ。

**命題 4.2.** 行列 $\boldsymbol{A}$ が安定固有値のみをもつ場合，行列 $(\boldsymbol{I} - \boldsymbol{A})$ は逆行列を持ち，次の式が成り立つ。

$$\sum_{n=0}^{\infty} \boldsymbol{A}^n = \lim_{n\to\infty} \left( \boldsymbol{A}^{n-1} + \boldsymbol{A}^{n-2} + \cdots + \boldsymbol{A} + \boldsymbol{I} \right) = (\boldsymbol{I} - \boldsymbol{A})^{-1}$$

証明. 証明するには

$$(\boldsymbol{I} - \boldsymbol{A}) \left[ \lim_{n\to\infty} \left( \boldsymbol{A}^{n-1} + \boldsymbol{A}^{n-2} + \cdots + \boldsymbol{A} + \boldsymbol{I} \right) \right] = \boldsymbol{I}$$
$$\left[ \lim_{n\to\infty} \left( \boldsymbol{A}^{n-1} + \boldsymbol{A}^{n-2} + \cdots + \boldsymbol{A} + \boldsymbol{I} \right) \right] (\boldsymbol{I} - \boldsymbol{A}) = \boldsymbol{I}$$

を確かめればよい。 □

**命題 4.3.** 行列 $\boldsymbol{A}$ が安定固有値のみをもつ場合，*(4.7)* は定常状態に収束する。

証明. 上の命題 4.1,4.2, 式 (4.8) を用いると，

$$\boldsymbol{a}_n = \boldsymbol{A}^n \boldsymbol{a}_0 + \left( \boldsymbol{A}^{n-1} + \boldsymbol{A}^{n-2} + \cdots + \boldsymbol{A} + \boldsymbol{I} \right) \boldsymbol{c}$$
$$\xrightarrow[n\to\infty]{} (\boldsymbol{I} - \boldsymbol{A})^{-1} \boldsymbol{c}$$

が分かる。 □

式 (4.7) のようにかけるモデルを線形差分方程式と呼ぶ。線形差分方程式が極限でよい振る舞いをするかどうかは，係数行列の固有値を調べればよい。これは非常に重要な性質なので記憶しておこう。

**問題 4.1.** 3 項間漸化式を 2 次ベクトルの漸化式に書き換えることができたのだから，一般の $(p+1)$ 項間漸化式を $p$ 次ベクトルの漸化式に書き換えることができるはずだ。やってみよう。

$$b_n = \alpha_0 + \alpha_1 b_{n-1} + \alpha_2 b_{n-2} + \cdots + \alpha_{n-p} b_{n-p}$$

**Answer**

線形代数を使うと，次数の大きさに関わらずに同じ表現とアルゴリズムを使うことができるという大きなメリットがある。線形代数学を学ぼう。

### AR(2) 過程

次の形式の時系列モデルを AR(2) 過程という。

$$y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + \varepsilon_t \tag{4.9}$$

$y_0$,$y_1$ が観測されていれば,

$$
\begin{aligned}
y_2 &= a_0 + a_1 y_1 + a_2 y_0 + \varepsilon_2 \\
y_3 &= a_0 + a_1 y_2 + a_2 y_1 + \varepsilon_3 \\
&\vdots \\
&\vdots
\end{aligned}
$$

を順に計算できる。もちろん，$\varepsilon_t$ は $\mathbb{E}_{t-1}[\varepsilon_t] = 0$ ($\mathbb{E}[\varepsilon_t] = 0$) ,$\mathrm{Var}(\varepsilon_t) \equiv \sigma^2$ を満たすホワイトノイズで観察のたびにランダムに決まる。

次の行列表現と同じであることを確かめよう。

$$
\begin{aligned}
\boldsymbol{x}_t &= \boldsymbol{A}\boldsymbol{x}_{t-1} + \boldsymbol{B}\tilde{\varepsilon}_t \\
y_t &= \boldsymbol{C}\boldsymbol{x}_t,
\end{aligned}
$$

ただし,

$$
\boldsymbol{x}_t = \begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix}, \quad \boldsymbol{A} = \begin{bmatrix} a_1 & a_2 \\ 1 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \tilde{\varepsilon}_t = \varepsilon_t + a_0.
$$

上で用いた行列表現とは違って，中間変数 $\boldsymbol{x}_t$ を導入した。このようなモデル表現を「状態空間表現」というが，今はあまり気にしなくてよい。係数行列の固有値は確率的な変動が入る場合にも役に立つ。行列 $\boldsymbol{A}$ が安定固有値のみを持つことは，時系列モデルの**弱定常性**と呼ばれる性質に関連している。弱定常性を持つ過程は

- すべての時点の変数 $y_t$ が同じ期待値と分散を持つ。すなわち，ある定数の周りをおおよそ同じ振れ幅で振動するような挙動を示す。
- 過去のデータが現在のデータに与える影響は，現在と過去の時間差のみに依存する。例えば，3 日前のデータ今日のデータに与える影響の強さは，2 日前のデータが明日のデータに与える影響の強さと同じである。

AR 過程が 1 つ目の性質を持つことは次のようにして分かる。まず，$\boldsymbol{x}_t$ が下のように表

現できることを確認する。

$$\boldsymbol{x}_t = \boldsymbol{A}^n \boldsymbol{x}_{t-n} + \sum_{k=0}^{n-1} \boldsymbol{A}^k \boldsymbol{B} \tilde{\varepsilon}_{t-k}, \quad n = 1, 2, \ldots$$

であることに注意する。両辺の期待値を取ると，（$\mathbb{E}[\tilde{\varepsilon}_t] = a_0$ に注意）

$$\begin{aligned}
\mathbb{E}[\boldsymbol{x}_t] &= \boldsymbol{A}^n \mathbb{E}[\boldsymbol{x}_{t-n}] + \sum_{k=0}^{n-1} \boldsymbol{A}^k \boldsymbol{B} \mathbb{E}[\tilde{\varepsilon}_{t-k}] \\
&\xrightarrow[n\to\infty]{} \sum_{k=0}^{\infty} \boldsymbol{A}^k \boldsymbol{B} a_0 \\
&= (\boldsymbol{I} - \boldsymbol{A})^{-1} \boldsymbol{B} a_0
\end{aligned}$$

したがって，

$$\mathbb{E}[y_t] = \boldsymbol{C} (\boldsymbol{I} - \boldsymbol{A})^{-1} \boldsymbol{B} a_0 \equiv \mu$$

となり，すべての時点で 平均が同じになる。

**問題 4.2.** AR(2) 過程 (4.9) に対して $\mu$ を計算しなさい。

**Answer**

すべての $t$ について分散が同一であることは以下のように示せる。この計算では

$a_0 = 0$（したがって，$\mu = 0$）と置くが，これによって一般性を失うことはない。

$$
\begin{aligned}
\mathrm{Var}(y_t) &= \boldsymbol{C}\mathbb{E}\left[\boldsymbol{x}_t\boldsymbol{x}_t^\top\right]\boldsymbol{C}^\top \\
\mathbb{E}\left[\boldsymbol{x}_t\boldsymbol{x}_t^\top\right] &= \mathbb{E}\left[\boldsymbol{A}\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}^\top\boldsymbol{A}^\top + (\text{独立性によりゼロになる項}) + \boldsymbol{B}\boldsymbol{B}^\top\varepsilon_t^2\right] \\
&= \boldsymbol{A}\mathbb{E}\left[\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}^\top\right]\boldsymbol{A}^\top + \boldsymbol{B}\boldsymbol{B}^\top\sigma^2 \\
&= (\boldsymbol{A})^2\,\mathbb{E}\left[\boldsymbol{x}_{t-2}\boldsymbol{x}_{t-2}^\top\right]\left(\boldsymbol{A}^\top\right)^2 + \left(\boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^\top\boldsymbol{A}^\top + \boldsymbol{B}\boldsymbol{B}^\top\right)\sigma^2 \\
&= \cdots \\
&= (\boldsymbol{A})^n\,\mathbb{E}\left[\boldsymbol{x}_{t-n}\boldsymbol{x}_{t-n}^\top\right]\left(\boldsymbol{A}^\top\right)^n + \left[\boldsymbol{A}^n\boldsymbol{B}\boldsymbol{B}^\top\left(\boldsymbol{A}^\top\right)^n + \cdots + \boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^\top\boldsymbol{A}^\top + \boldsymbol{B}\boldsymbol{B}^\top\right]\sigma^2
\end{aligned}
$$

$\boldsymbol{A}$ の安定性より，$n \to \infty$ で収束して，

$$
\begin{aligned}
\mathbb{E}\left[\boldsymbol{x}_t\boldsymbol{x}_t^\top\right] &= \sum_{n=0}^{\infty}\boldsymbol{A}^n\boldsymbol{B}\boldsymbol{B}^\top\left(\boldsymbol{A}^\top\right)^n\sigma^2 \equiv \boldsymbol{\Sigma}, \\
\mathrm{Var}(y_t) &\equiv \boldsymbol{C}\boldsymbol{\Sigma}\boldsymbol{C}^\top
\end{aligned}
$$

$\boldsymbol{\Sigma}$ の具体的な値を求めるには，行列方程式

$$
\boldsymbol{\Sigma} = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^\top + \boldsymbol{B}\boldsymbol{B}^\top\sigma^2 \tag{4.10}
$$

を $\boldsymbol{\Sigma}$ について解けばよい。式 (4.10) のような方程式は**離散時間リアプノフ方程式**と呼ばれている。

**問題 4.3.** 行列方程式 (4.10) の未知行列 $\boldsymbol{\Sigma}$ を

$$
\boldsymbol{\Sigma} = \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix}
$$

と成分表示する。方程式

$$
\begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix}\begin{bmatrix} a_1 & 1 \\ a_2 & 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \end{bmatrix}\sigma^2
$$

を $\alpha, \beta, \gamma$ について解き，

$$\mathrm{Var}(y_t) = \frac{(1 - a_1^2)\sigma^2}{1 - a_2 - (a_1^2 + a_2^2) - a_2(a_1^2 - a_2^2)}$$

であることを示しなさい。

**Answer**

異なる時点の変数間の共分散（自己共分散）を計算しよう。これも $a_0 = 0$ として一般性を失うことはない。

$$\begin{aligned}
\mathbb{E}\left[\boldsymbol{x}_t \boldsymbol{x}_{t-k}^\top\right] &= \mathbb{E}\left[\left(\boldsymbol{A}^k \boldsymbol{x}_{t-k} + \sum_{j=0}^{k-1} \boldsymbol{A}^j \boldsymbol{B} \varepsilon_{t-j}\right) \boldsymbol{x}_{t-k}^\top\right] \\
&= \boldsymbol{A}^k \boldsymbol{\Sigma} + \sum_{j=0}^{k-1} \boldsymbol{A}^j \mathbb{E}\left[\boldsymbol{B} \underbrace{\varepsilon_{t-j}}_{t-j>t-k} \boldsymbol{x}_{t-k}^\top\right] \\
&= \boldsymbol{A}^k \boldsymbol{\Sigma}
\end{aligned}$$

となる。最後の等式は，ホワイトノイズの仮定から得られる「過去の $\boldsymbol{x}$ と未来の $\varepsilon$ が独

立である」という性質によって成り立つ。$y$ については

$$\mathrm{Cov}(y_t, y_{t-k}) = \boldsymbol{C}\boldsymbol{A}^k\boldsymbol{\Sigma}\boldsymbol{C}^\top$$

である。一般に，$\mathrm{Cov}(y_t, y_{t-k})$ は $t$,$k$ に依存するが，弱定常 AR 過程の自己共分散は $t$ にはよらず時間差 $k$ のみで決定できる。自己共分散を $k$ の関数と見て，

$$\phi(k) = \mathrm{Cov}(y_t, y_{t-k}) = \boldsymbol{C}\boldsymbol{A}^k\boldsymbol{\Sigma}\boldsymbol{C}^\top$$

を定義する。$\phi(k)$ を**自己共分散関数**という。

自己共分散が時間差のみに依存するという性質は応用上極めて重要である。$\mathrm{Cov}(y_t, y_{t-k})$ の定義に含まれる期待値は，本来，時系列の生成を何度も繰り返して計算した結果の平均（アンサンブル平均）によって推定されるべきものである。しかし，1 回だけ生成した長時間時系列の中には同じ時間差を持つデータがたくさん入っているので（$y_t$ と $y_{t-k}$,$y_{t+1}$ と $y_{t-k+1}$,$y_{t+2}$ と $y_{t-k+2}$ など），これらの実データの平均値を使うことで共分散の推定が可能になる。アンサンブル平均が長時間平均と一致するという性質は**エルゴード性**として知られている。実データを用いる時系列分析では，1 回だけ生成された時系列しか使えないことがほとんどなので，エルゴード性を持つモデルは大変都合がよいのだ。

#### 自己相関のグラフ

$\boldsymbol{A}$ が安定固有値のみをもつ場合，自己共分散関数

$$ACF_k = \boldsymbol{C}\boldsymbol{A}^k\boldsymbol{\Sigma}\boldsymbol{C}^\top, \quad k = 0, 1, 2, \ldots$$

は $k \to \infty$ に伴って減衰（ゼロに漸近）する挙動を示す。これは弱定常 AR 過程の重要な特徴である。

さらに，偏自己相関関数（partial autocorrelation function, PACF) という重要な概

念があるので，ここで紹介しておこう。AR (2) のケースでは，

$$\begin{aligned} y_t &= a_1 y_{t-1} + a_2 y_{t-2} + (*) \\ &= a_1(a_1 y_{t-2} + (*)) + a_2 y_{t-2} + (*) \end{aligned}$$

なので（議論に関係ない部分は (*) としている），$y_{t-2} \longrightarrow y_t$ の影響は，$y_{t-1}$ を経由した間接的な影響（$a_1^2 y_{t-2}$）と，$y_1$ を介さない直接的な影響（$a_2 y_{t-2}$）の 2 つの部分に分解できる。間接的な影響を取り除いて直接的な影響のみを取り出そうとするものが PACF である。AR 過程の場合は係数が PACF に相当するので，AR(2) では $k = 2$ のことろでするどいカットオフが生じる。

$$PACF_1 = a_1, \quad PACF_2 = a_2, \quad PACF_3 = PACF_4 = \cdots = 0$$

与えられたデータから ACF や PACF を推定[3]，可視化することができるので，ここで説明したような定性的な性質を理解しておけば，手元の時系列データが AR 過程で表現できそうか，また，次数がどれくらいかを見積もることができる。詳細は**?** を参照。

## 4.3　プログラミング：シミュレーション

データからモデルやモデルのパラメータを決定することを**推定**という。逆に，1 つのモデルパラメータからデータを生成することを**シミュレーション**という。この節ではシミュレーションについて説明し，次節で推定について簡単に紹介する。実データを使った分析例は次節で紹介する。（予定。まだない）

```
import numpy as np
import matplotlib.pyplot as plt
```

[3] 標本 ACF，標本 PACF は線形回帰モデルを用いて推定する。

### 4.3.1　時系列データの表現

**2??**章では，ベクトル時系列データを行列形式で表現した。つまり，$T$ 個の $d$ 次元ベクトル

$$\boldsymbol{x}_t = \begin{bmatrix} x_{t,1} \\ x_{t,2} \\ \vdots \\ x_{t,d} \end{bmatrix}, \qquad t = 0, 1, 2, \ldots, T-1$$

からなる時系列データを $T \times d$ 行列（2 次元配列）を用いて，

$$\begin{bmatrix} x_{0,1} & x_{0,2} & \cdots & x_{0,d} \\ x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{t,1} & x_{t,2} & \cdots & x_{t,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{T-1,1} & x_{T-1,2} & \cdots & x_{T-1,d} \end{bmatrix}$$

のように表現した。このような表現が最も基本的であるのは間違いないのだが，**NumPy** で「$t$ 期のデータ（行）を取り出す」という操作をするときに，取り出したベクトルには縦横の情報が設定されていない状況（1 次元配列）になってしまう。行列とベクトルの計算するときに間違いが起こりやすく，エラーの温床になってしまう。そこで，少し方針を変えて次のような愚直な理解をしよう。

ベクトル時系列 = $T$ 個の $d \times 1$ 行列を並べたもの

つまり，シェイプが $(T, d, 1)$ であるような **NumPy** 配列（3 次元配列）を用いて $T$ 期間 $d$ 変数のデータを表現する。このように作った配列から $t$ 期の情報を取り出すと，シェイプ　$(d, 1)$ の配列（2 次元配列）となるので，いつでも「列ベクトル」になっている。

実際のコードを見てみよう。配列 **x** は 3 期間，2 変数のデータである。ゼロで初期化している。

表 4.1: ベクトル時系列の表現

| | 2??章の表現 | この章の表現 |
|---|---|---|
| 全体のシェイプ | $(T, d)$ | $(T, d, 1)$ |
| $t$ 期のベクトルのシェイプ | $(d,)$ | $(d, 1)$ |

```
T, d = 3, 2
x = np.zeros((T, d, 1))
x
```

```
array([[[0.],
        [0.]],

       [[0.],
        [0.]],

       [[0.],
        [0.]]])
```

初期値を代入しよう。

```
x[0] = np.array([[1],
                 [2]])
x
```

```
array([[[1.],
        [2.]],

       [[0.],
        [0.]],

       [[0.],
        [0.]]])
```

$t-1$ 期のデータから $t$ 期のデータに変換する行列 $A$ を次のように設定する。

```
A = np.array([[0.5, -0.2],
              [1.0, 0.0]])
A
```

```
array([[ 0.5, -0.2],
       [ 1. ,  0. ]])
```

ここでは

$$x_t = Ax_{t-1}, \quad t = 1, 2$$

という単純な差分方程式の解を逐次的に計算してみよう。「行列 × ベクトル」の演算は Python ではアットマーク@ を用いる。

```
for t in range(1, T):
    x[t] = A @ x[t-1]


x
```

```
array([[[ 1.  ],
        [ 2.  ]],

       [[ 0.1 ],
        [ 1.  ]],

       [[-0.15],
        [ 0.1 ]]])
```

計算が終わった後には，表形式でデータを持っておきたいと考えるかもしれない。これには シェイプを上書きするだけでよい。

```
x.shape = (T, d)
x
```

```
array([[ 1.  ,  2.  ],
       [ 0.1 ,  1.  ],
       [-0.15,  0.1 ]])
```

**問題 4.4.** 最後のコードで，シェイプが (T, d) である 2 次元配列を得た。これをシェイプが (T, d, 1) である 3 次元配列に戻すにはどのようにすればよいか？[4]

**問題 4.5.** この節のコードを 3 次元配列を使わないコードで再現しなさい。

### 4.3.2　行列の安定性

安定行列に関する命題 4.1,4.2 を確認しておこう。

安定性を判別するため数値線形代数用のライブラリをインポートしておく[5]。

```
import scipy.linalg as LA
import matplotlib.pyplot as plt
```

次の行列 $A$ の安定性を確認しよう。

```
A1 = np.array([[0.8, 0.1],
               [0.5, 1.2]])
```

E, V = LA.eig(A) というコードを実行すると，E に固有値，V に固有ベクトルを並べた行列が格納される（E, V という名前は変えてもよい）。今知りたいのは固有値の絶対値なので，np.abs(E) をチェックする。

```
E, V = LA.eig(A1)
np.abs(E)
```

```
array([0.7, 1.3])
```

1 を超えるものがあるので，$A^n$ は発散するはずだ。確認するために各要素 $A = [a_{ij}]$ の絶対値の和 $\sum_{i,j} |a_{ij}|$ の振る舞いを可視化してみよう。（図 4.1）

```
N = 50
x = np.zeros(N)
```

[4] 解答:x.shape = *x.shape, 1 とするのがクリーンだろう。右辺は * でタプルを一旦ばらして，1 を追加したタプルを作っている。こういうものだと考えておけばよい。

[5] numpy.linalg を使ってもよい。

```
An = np.eye(A1.shape[0])
x[0] = np.sum(np.abs(An))        # Σ |a_n|
for n in range(1, N):
    An = An @  A1
    x[n] = np.sum(np.abs(An))    # Σ |a_n|

plt.plot(x)
```

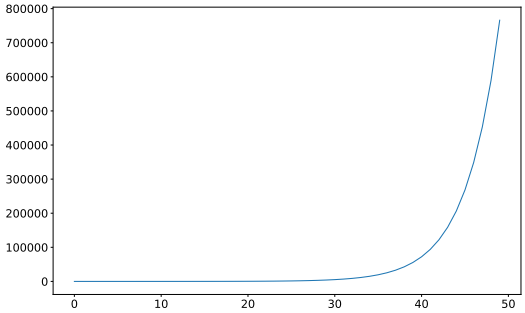


図 4.1: 不安定固有値を持つ行列のべき

次の行列は安定固有値のみを持つようだ。

```
A2 = np.array([[0.7, -0.1],
               [0.3, 1.05]])

E, V = LA.eig(A2)
np.abs(E)
```

```
array([0.85, 0.9 ])
```

可視化の結果は図 4.2 のようになる。コードは省略する。

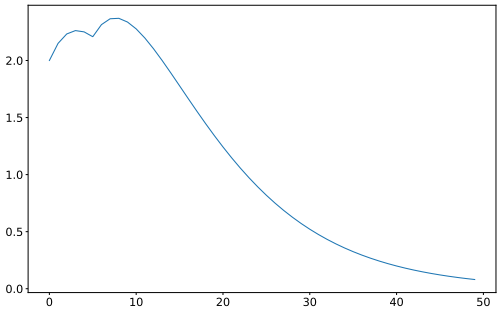


図 4.2: 安定固有値のみを持つ行列のべき

上で定義した `A2` は命題 4.2 の公式

$$(\boldsymbol{I} - \boldsymbol{A})^{-1} = \sum_{n=0}^{\infty} \boldsymbol{A}^n$$

を満たすはずである。極限の計算はできないので，十分大きい $N$ に対して，

$$(\boldsymbol{I} - \boldsymbol{A}) \left( \sum_{n=0}^{N} \boldsymbol{A}^n \right) \approx \boldsymbol{I}$$

が成り立つことを確認しよう。総和の計算には，

$$\sum_{n=0}^{N} \boldsymbol{A}^n = \boldsymbol{I} + \boldsymbol{A} \left( \sum_{n=0}^{N-1} \boldsymbol{A}^n \right)$$

という関係を用いると便利である。

```
I = np.eye(A2.shape[0])
S = I
for n in range(1, 200):
    S = I + A2 @ S

np.allclose((I - A2) @ S, I)
```

```
True
```

### 4.3.3　AR 過程

AR(2) 過程

$$y_t = 0.6y_{t-1} + 0.3y_{t-2} + \varepsilon_t \tag{4.11}$$

をシミュレーションしてみよう。シミュレーションというのは，適当な初期値 $y_0$,$y_1$ と，適当に発生させたランダムな $\varepsilon_t$ を使って $y_t$ を計算してみることである。状態空間表現を用いるとクリーンなコードを書けるので，この表現を用いよう。

$$\begin{aligned} \boldsymbol{x}_t &= \boldsymbol{A}\boldsymbol{x}_{t-1} + \boldsymbol{B}\varepsilon_t \\ y_t &= \boldsymbol{C}\boldsymbol{x}_t, \end{aligned}$$

ただし，

$$\boldsymbol{x}_t = \begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix}, \quad \boldsymbol{A} = \begin{bmatrix} 0.6 & 0.3 \\ 1 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 1 & 0 \end{bmatrix},$$

$$\mathbb{E}[\varepsilon_t] = 0.0, \quad \mathrm{Var}(\varepsilon_t) = 1.0$$

とする。初期値はゼロとしよう。係数行列と初期条件を適当に設定しさえすれば，一般の AR ($p$) 過程をほぼ同じコードで記述できる。

```
A = np.array([[0.6, 0.3],
              [1.0, 0.0]])
```

固有値を確認する。

```
E, V = LA.eig(A)
np.abs(E)
```

```
array([0.9244998, 0.3244998])
```

問題なさそうなので，他の行列も定義しよう。

```
B = np.array([[1.0],
              [0.0]])
C = np.array([[1.0, 0.0]])
x0 = np.array([[0.0],
               [0.0]])
```

ランダムな外乱 $\varepsilon_t$ は正規乱数を用いて生成しよう。平均 0，分散 1 の正規乱数は次のように生成する。loc,scale,size はそれぞれ平均，標準偏差，大きさを表すパラメータである。size にタプルを渡すと，この章の時系列表現と同じような形式で乱数を生成できる。

```
rng = np.random.default_rng(1234)    # 124 は再現性のため
rng.normal(loc=0, scale=1.0, size=(3, 1, 1))
```

```
array([[[-1.60383681]],

       [[ 0.06409991]],

       [[ 0.7408913 ]]])
```

これで準備が整った。300 期間分のシミュレーションをして結果をプロットしてみよう。

```
T, d = 300, 2
x = np.empty((T, d, 1))
y = np.empty((T, 1, 1))
eps = rng.normal(loc=0, scale=1.0, size=(T, 1, 1))

x[0] = x0
y[0] = C @ x[0]

for t in range(1, T):
    x[t] = A @ x[t-1] + B @ eps[t]
```

```
    y[t] = C @ x[t]


y.shape = (T, )
y[:5]
```

```
array([0.         , 0.86374389, 3.43134556, 0.83910714,
       2.47834093])
```

作図のために 1 次元配列に変換していることに注意しよう。

```
plt.plot(y)
```

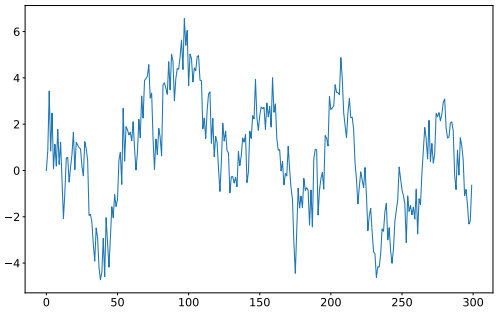


図 4.3: AR(2) のシミュレーション

## 4.4　プログラミング：推定

この節では，Python を使ったデータの取得とモデルパラメータの推定を簡単に紹介する。Python による推定は **statsmodels** というライブラリを用いる。時系列分析（time series analysis）用のツール群を使うためには次のインポート文を実行する。

```
import statsmodels.tsa.api as tsa
```

データとして先程作った y を用いよう。

次のように標本自己相関を可視化できる（図 4.4）。標本自己相関は，間接的な影響も

含めて，過去から現在への影響の強さを見るもので，図の右の方に進むほど「より過去」のデータが現在に与える影響を表している。図 4.4 は標本自己相関が徐々に減衰するという AR 過程の特徴を表現している。

```
tsa.graphics.plot_acf(y)
plt.show()
```

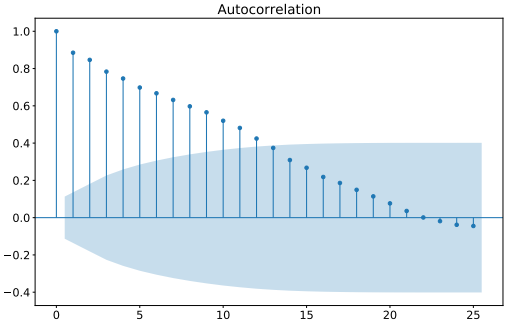


図 4.4: 標本自己相関

標本偏自己相関は次のようにして可視化する（図 4.5）。標本偏自己相関は，間接的な影響を除いて，過去から現在への影響の強さを見るものである。図の右の方に進むほど「より過去」のデータが現在に与える影響を表しているという点では標本自己相関と同じである。図 4.5 は標本偏自己相関がするどくカットオフするという AR 過程の特徴を表現している。

```
tsa.graphics.plot_pacf(y)
plt.show()
```

推定には SARIMAX というメソッドを用いる。これは AR モデルを特殊ケースとして含む SARIMAX モデルの係数を推定するメソッドである。ちなみに，SARIMAX というのは，以下のようなコンポーネントを持つ時系列モデルである。詳しくは時系列分析の専門書を読んでほしい。

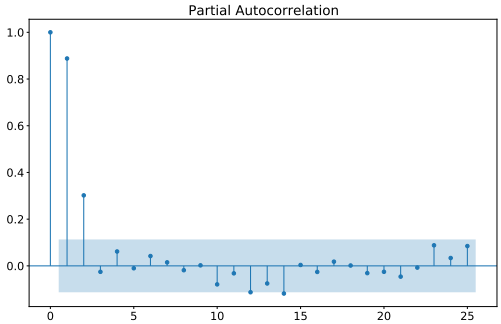


図 4.5: 標本自己相関

- S: Seasonal, 季節変動成分を許す
- AR: 自己回帰
- I: Integral, 和分過程
- MA: Moving Average, 移動平均
- X: eXogenous, 外生変数

データと `order=(2, 0, 0)` だけを指定すると，季節成分も移動平均成分も外生変数を持たない和分過程でない自己回帰モデル，すなわち普通の AR(2) 過程を推定する。

推定は

- モデルの設定
- モデルのフィット

という 2 段階で行われる。

```
mod = tsa.SARIMAX(y, order=(2, 0, 0))
result = mod.fit()
```

結果は `summary` メソッドを使って確認する。ここでは紙面の都合で `params` を見ることにする。

```
result.params
```

```
array([0.61992331, 0.30242164, 1.05848657])
```

得られた結果とデータを生成するときに使ったパラメータと見比べて，数字の意味を確認しておいてほしい。

**問題 4.6.** 次のコードを実行して，表示された内容を確認しなさい。パラメータはどこに書かれているか。他の数字をどのように解釈するか。

```
result.summary()
```

**問題 4.7.** 適当に与えたパラメータに対して AR 過程のシミュレーションを実行しなさい。シミュレーション結果に対して推定のコードを実行し，パラメータを復元できているかどうかを確認しなさい。