ARMAモデル

1. AR, MA,　ARMAモデル
2. 次数と因果性
3. システム同定
4. 可同定性とPE性
5. 極（pole）

はじめに

- ARモデルは，U.Yule（統計学者，英）が太陽黒点の分析（1927）に導入したことが嚆矢とされる。しばらく陽の目を見なかったが，1960年代に入り，H.Akaike（統計学者，日本）やE.Parzen（統計学者，米）によって再発見，再評価されるようになった。さらに，G.Box and G.Jenkinsらが1960年後半から70年代に発表したARIMA（非定常過程），SARIMA（周期的過程（季節性））の有効性が認められ，経済データや気象データのように入力を観測できない時系列データに盛んに適用されるようになった。
- 一方，工学（電気，機械，土木，建築など）の分野では数式（特に微分方程式や偏微分方程式）で表される物理モデルを用いた同定，制御，対象システムの解析と分析などが数学に立脚して，よく発展した。さらにコンピュータを用いたディジタル制御やディジタル信号処理に関する理論やツールが発展し，この流れに沿って，システム同定理論が発展した。
- 現在では，システム同定が数学に立脚した確率論，z変換に基づく解析，およびスペクトル解析の発展と共に，ARMAモデルの性質や解析の方法が良く研究されているので，モデルの見方や性質については，こちらの説明に基づくものとする。
- ただし，システム同定は入力を与えられることを前提としている。一方，statsmodelsは入力を観測できないことが前提であるので，この点に関しての留意点は述べる。

離散時間システムの表現

離散時間データ

- y(k)は離散時間データ，　y(t)は連続時間データ

動的システムの表現は

- 動的システムの離散化で見たように，遅延演算があればよい
- 例：y(k)+2 y(k－1)=1
- k＜0 で，変数は全て0とする。k≧0で，y(k) = －2y(k－1)+1 を考えて，
  y(0)=1，y(1)=－1，y(2)＝3，y(3)＝－5，・・・・・・

遅延演算子の導入

- z^{-1} = z ^{－1}
- z^{-1} y(k) = y (k－1)
- 離散時刻kが一つ進むと蓄えたデータを出力することから，一時的にデータを蓄えるメモリともみなせる
- 上式は右と同じ　$(1 + 2\,z^{-1})\,y(k) = 1$

記号zを使う理由

- 本書では触れないが，遅延演算子に記号z^{-1}を用いた理由は，離散時間システムの解析で広く知られているz変換の演算子z(z = exp (jω))と同じにしたためである。z変換により，後に述べる，ARモデルやMAモデルの安定性や性質を解析的に知ることができる。

離散時間系の差分形式の見方と遅延演算子

$$y(k) = -2y(k-1) - y(k-2) + u(k) + 0.5u(k-1)$$

k	$y(k) =$	$-2y(k-1)$	$-y(k-2)$	$+u(k)$	$+0.5u(k-2)$
0	1	0	0	1	0
1	−1.5	−2	0	0	0.5
2	2	3	−1	0	0
3	−2.5	−4	1.5	0	0

図 8.9: 差分方程式の振舞い

ディジタル回路の基本
AR, MAの代わりに
FIR, IIRという名称を
用いている。

遅延演算子（delay operator）z^{-1} を導入：

$$z^{-1}y(k) = y(k-1),\ z^{-2}y(k) = y(k-2),\ \cdots,\ z^{-p}y(k) = y(k-p)$$

$$y(k) + 2z^{-1}y(k) + z^{-2}y(k) = u(k) + 0.5z^{-1}u(k)$$

$$\left(1 + 2z^{-1} + z^{-2}\right) y(k) = \left(1 + 0.5z^{-1}\right) u(k)$$

$$A(z^{-1}) = \left(1 + 2z^{-1} + z^{-2}\right),\ B(z^{-1}) = \left(1 + 0.5z^{-1}\right)$$

伝達関数

$$A(z^{-1})y(k) = B(z^{-1})u(k)$$

$$y(k) = \frac{B(z^{-1})}{A(z^{-1})}u(k)$$

AR，ARMA，　ARIMAモデル

モデル

- AR model: AutoRegressive，自己回帰モデル
- ARMA model: AutoRegressive Moving Average，自己回帰移動平均モデル
- ARIMA model: AutoRegressive, Integrated and Moving Average，自己回帰和分移動平均モデル，d階差分をとった系列が定常かつ反転可能なARMA(p,q)過程に従う過程に従う次数(p,d,q)のモデルで，ARIMA(p,d,q)と表現する。

AR and ARMA model, see “Time_Series_Analysis_in_Python_with_statsmodels.pdf”（フォルダDataScience内）

予測の他の手法

- Statsmodels Examples, http://statsmodels.sourceforge.net/devel/examples/
- ここの” Prediction”を見る。－> “Prediction (out of sample)”はいい例ですので読んでください。
  - Artificial dataは「人工データ」という意味で、何らかの関数（数学関数やランダム発生関数）を用いて発生させたデータを言う。開発中のアルゴリズムの有効性を検証するのに良く用いられる。

ARMAモデル

ARMA(p, q)

$$y(k)+a_1y(k-1)+\cdots+a_py(k-p)=b_0u(k)+b_1u(k-1)+\cdots+b_qu(k-q)+w(k)$$

- 制御論やディジタル信号論では，w(k)は外乱または観測雑音を意味する
- 統計やディジタルフィルタ設計では，観測雑音を考慮しなくても良いため，この項が無い。statsmodelsはこれに準じているため，以降の話ではw(k)=0とおき無視して説明します。
- AR（Auto Regressive ；自己回帰）モデル

$$y(k)+a_1y(k-1)+\cdots+a_py(k-p)=u(k)$$

$$\Rightarrow\left(1+a_1z^{-1}+\cdots+a_pz^{-p}\right)y(k)=u(k)\Rightarrow A\left(z^{-1}\right)y(k)=u(k)$$

  - 出力自身の過去の値が現在の出力に影響を及ぼすダイナミクスのあるモデルである。数学的観点やシステム工学の観点から，まさしく自己回帰である。統計分析の回帰モデルとは成り立ちも内容も異なることに注意されたい。$A(z^{-1})$はモデルの挙動や安定性に大きく影響を与える。

- MA（Moving Average；移動平均）モデル

$$y(k)=b_0u(k)+b_1u(k-1)+\cdots+b_qu(k-q)$$

$$\Rightarrow y(k)=\left(b_0+b_1z^{-1}+\cdots+b_qz^{-q}\right)u(k)\Rightarrow y(k)=B\left(z^{-1}\right)u(k)$$

  - 入力の過去値が現在の出力に影響を及ぼすモデルである。平均の意味は，入力u(k)の単純平均や加重平均の表現が$B(z^{-1})$の形式で表されることに注意されると，$B(z^{-1})$が平均操作を表すことがわかる。しかも，時刻kとともに，この平均区間は移動する。これらから移動平均と名付けられた。

- ARの次数をp，MAの次数をqとしたとき，ARMA(p, q)という表現を用いる

$$A\left(z^{-1}\right)y(k)=B\left(z^{-1}\right)u(k)$$

ARMAモデル

次数と因果性

- 因果性を考えると　$p \geqq q$ である。これを制御論では**プロパー**(proper)と呼ぶ。
- このうち，$p > q$ を厳密なプロパー(strictly proper)と呼ぶ。
- これは，連続時間で考えた線形な動的システムの場合と同じである。
- この専門用語は無視しても，一般には　$p > q$とした方がシステムとして扱いやすい。

アドバンス：ARMAモデル

b0項の問題

- これは，現時刻kの入力u(k)が直接伝達するため，直達項とも言われる
- ダイナミクスシステムでは，ステップ応答を見たように，入力の影響は必ず遅れて現れるためb0項は無いものとして考える。電気・機械系，化学系・医学系，自然現象系（気象，海洋，天体など）など。
- b0項が考えられるのは取引市場（金融など），都市問題（人口），ディジタルフィルタ設計などがある。
- 一方，入力が観測できない場合には，入力に正規乱数を仮定していることが多い。この場合，出力には，非常に不規則性の高い正規乱数が重畳していることになる。
- b0項があれば，システム同定で言う1段予測はできない。次の式で説明。（注意：予測（制御））

$$\hat{y}(k|k-1) = \left(a_1 z^{-1} + \cdots + a_p z^{-p}\right) y(k) + \left(b_1 z^{-1} + \cdots + b_q z^{-q}\right) u(k)$$

$$\hat{y}(k|k) = \left(a_1 z^{-1} + \cdots + a_p z^{-p}\right) y(k) + \left(b_0 + b_1 z^{-1} + \cdots + b_q z^{-q}\right) u(k)$$

- statsmodelsでは用意していないが，ARMAモデルを用いて，電力需要の数時間先予測に見られる，d段予測（d>0となる自然数）$\hat{y}(k+d|k)$は可能である（他書を参照）。
- statsmodelsでは，b0 =1とおいている。この事実は，例えば，
  - statsmodels.tsa.arima_model.ARMAResults.plot_predict(http://www.statsmodels.org/dev/generated/statsmodels.tsa.arima_model.ARMAResults.plot_predict.html)の[source]（http://www.statsmodels.org/dev/_modules/statsmodels/tsa/arima_model.html#ARMAResults.plot_predict）を見ると
  - def _fit_start_params_hr(self, order, start_ar_lags=None):
  - def _forecast_error(self, steps):
  - これらの中で，MAパラメータのb0の位置に相当する場所に1を与えている。

システム同定

パラメータと次数を求める

- $A(z^{-1})$と $B(z^{-1})$の**パラメータが未知**であって，これを求めることをいう。パラメータ同定，パラメータ推定ともいう。
- さらに，同定という行為には次も含む
  - 次数の選定
  - サンプリング時間の選定
- システム同定では，入力u(k)はこちらで選定でき観測できるという場合がある。しかし，市場や気象に関する時系列データでは，一般にu(k)の観測は不可である。しかし，システム同定の要件を知るには，当面，u(k)の備えるべき性質も知るべきである。

同定とは

- 同定（identification)は，もともと，生物学において，生物の種名を定めることを言った。例えば，外見の特徴を分類して，さらに，解剖して体内の構造の特徴を分類する。これらの特徴量を用いて種名を定める。この行為を同定と称する。
- 制御工学の分野で同定とは，数理モデル構造を決定する行為を指し，本来は，数式そのもの選定行為を意味し，システム同定ともいう。ここでは，ARMAモデル構造に限定し，同定はパラメータ推定と次数の選定に留める。

アドバンス：可同定性とPE性の条件

パラメータを推定できる条件

- $A(z^{-1})$と $B(z^{-1})$の次数は既知，パラメータは未知とする。
- 全てのパラメータを推定できるためには，入力の周波数成分が重要である。
- これを測る指標に　PE性（persistently exciting）がある。
  - 詳細な説明は他書に譲るとして，ポイントだけ述べると
  - 正弦波入力 u(k)= Asin ω k　は次数2のPE性であると言われる。この次数は，単一の角周波数ωのもとで，振幅と位相の二つの自由度を示す。この場合，ARMA(1,1)（b0項は無いとして）の a_1とb_1を同定できる。

PE性の条件

- パラメータの数（p+q）を同定するならば，入力信号は次数（p+q）（b0項を含むならばp+q+1）のPE性信号でなければならない。（システム同定の他書を参照）
- 要は，パラメータの数だけ周波数成分が豊富な入力信号を与えなさい，ということを述べている。

さらに次の条件が必要

- 対象システムは安定である。
- 多項式A(z-1)，B(z-1)は共通因子（共通の根）を持たない。
- B(z-1)の係数はすべてゼロでない。

より詳しく知るには

- 確率論の確率極限，一致推定量，不偏推定量に基づき，パラメータが求められるための幾つかの条件や定理を知ればよい。

入力信号の候補

白色雑音（white noise）

- 全周波数領域にわたり一定値をとる確率変数で，全ての周波数を含んだ光が白色であることが名前の由来である。連続時間信号ではパワー無限大であるから架空上の信号。
- ちなみに，周波数により，信号のパワーが変化するものは有色雑音（colored noise）と呼ばれる。
- このパワースペクトル密度関数を$P(\omega) = \text{const.}$（一定値）となる。離散時間信号の場合は，$P(\exp(i\omega)) = \text{const.}$（一定値）というように，定数で表されることとなる。
- 白色雑音の自己相関関数は，ウィーナーヒンチン定理（ Wiener–Khinchin theorem）より，ラグが0のときだけある分散値をとり，これ以外で自己相関関数は0となる。これは，ディラック（Dirac）のデルタ関数で表される。離散時間信号の場合は，クロネッカー（Kronecker）のデルタ関数となる。
- パワースペクトル密度関数が定数，自己相関関数がデルタ関数で表現されると，数式で追及する解析の取り扱いが便利になるため，白色雑音はよく用いられる。
- しかし，白色雑音は現実には存在しないので，現実的な入力信号の作成は後に述べる。
- 連続時間，離散時間のどちらのデルタ関数で表現されても，これは無相関（uncorrelated）な信号である。したがって，無相関なランダム信号は白色雑音である。
- 備考：確率論より，独立ならば無相関は成立するが，この逆は必ずしも成立しない。
- 無相関なランダム信号の種類は多数ある。正規乱数（正規分布に従う確率変数）は無相関であるため，白色雑音の一種である。また，解析で取り扱いやすいため，理論を展開する場合や，シミュレーションではよく同定入力として用いられる。

入力信号の候補

実存するシステム

- 電気・機械系，化学系，自然科学系など現実に存在するシステムの多くは，正規乱数を加えると，負担が大きすぎて支障が生じる。この場合，PE性の高いものとして，次の入力を用いることが多い
  - M系列信号(Maximum length sequence, https://en.wikipedia.org/wiki/Maximum_length_sequence)
  - PBS(Pseudorandom binary sequence, https://en.wikipedia.org/wiki/Pseudorandom_binary_sequence)
  - 通常入力に小さな分散の正規乱数を重畳させる

入力信号が観測できないシステム

入力が観測できないシステムとは

- 金融データ，気象量（気温，湿度，風速など），自然現象（川の流量，天体観測量など），社会データ（人数に関する量，市場に関する量など）は入力信号が観測できない
- しかし，ARMAモデルを用いる以上，何らかの入力はあると考える。
- このため，仮想入力は正規乱数であると仮定されることが多い。このため，入力のPE性はあるとされている。
- しかし，保証されているか否かは不明である，というのが本当のところである。

ARモデル

解説

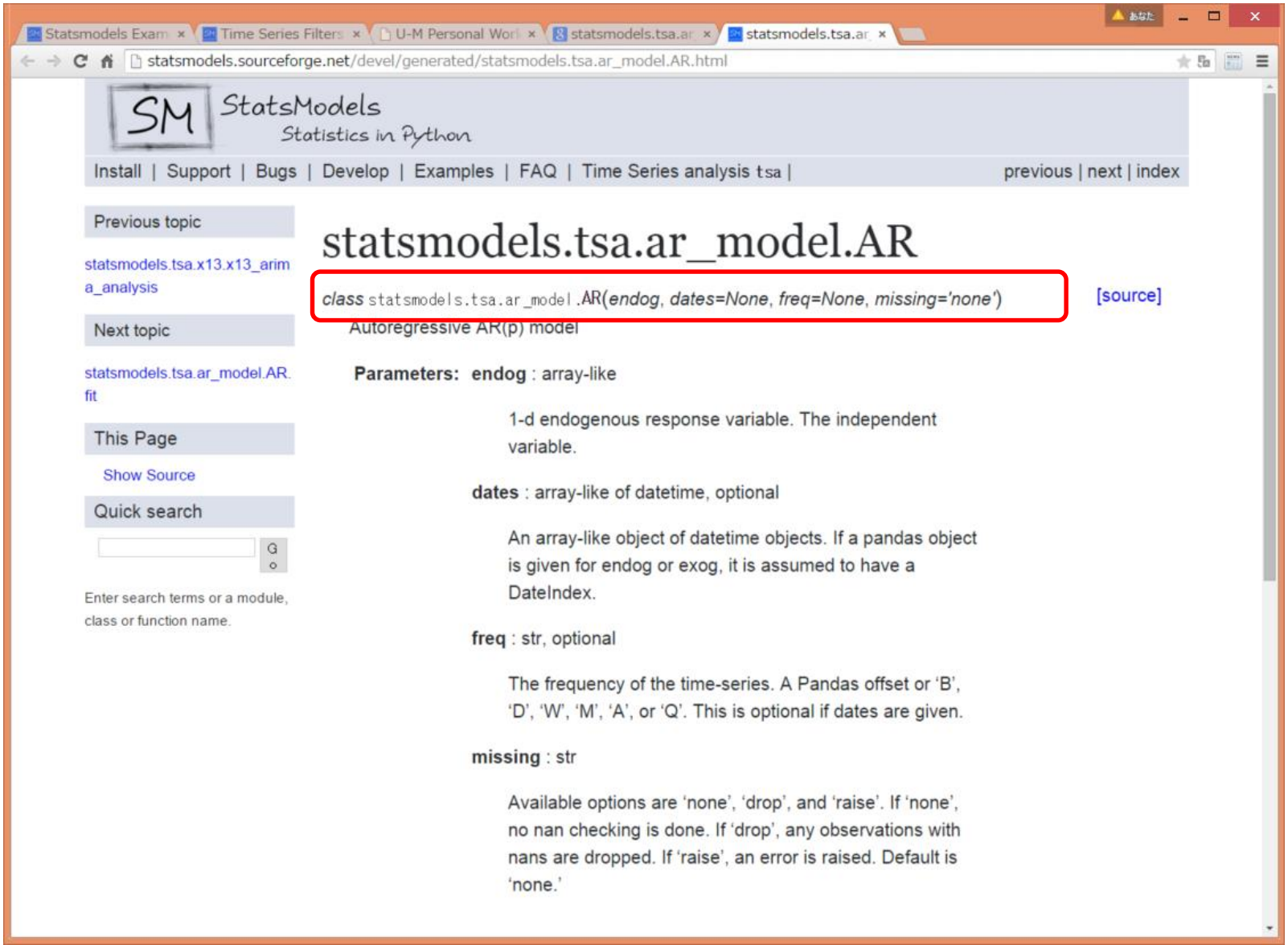
- “Hashimoto-TSA.pdf”

参照　Statsmodels Examples, http://statsmodels.sourceforge.net/devel/examples/

- Prediction (out of sample)
  http://www.statsmodels.org/0.6.1/examples/notebooks/generated/predict.html
- Autoregressive Moving Average (ARMA): Sunspots data
  http://www.statsmodels.org/dev/examples/notebooks/generated/tsa_arma_0.html
- statsmodels.tsa.ar_model.AR　の取説
  http://statsmodels.sourceforge.net/devel/generated/statsmodels.tsa.ar_model.AR.html （次のページで解説）

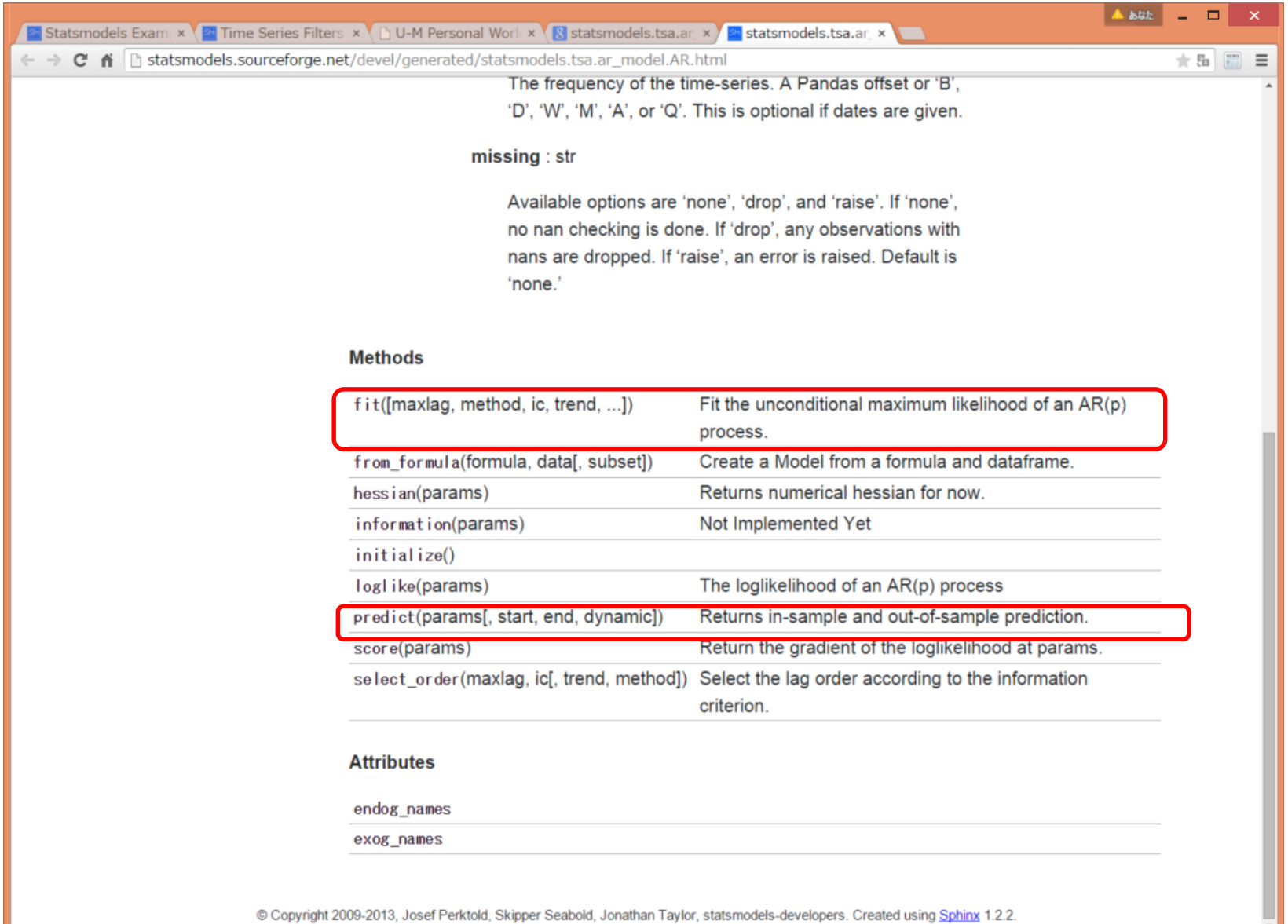
ARモデル

- http://statsmodels.sourceforge.net/devel/generated/statsmodels.tsa.ar_model.AR.html



ARモデル

- http://statsmodels.sourceforge.net/devel/generated/statsmodels.tsa.ar_model.AR.html



Statsmodels Exam × | Time Series Filters × | U-M Personal Worl × | statsmodels.tsa.ar × | statsmodels.tsa.ar ×

statsmodels.sourceforge.net/devel/generated/statsmodels.tsa.ar_model.AR.html

The frequency of the time-series. A Pandas offset or 'B', 'D', 'W', 'M', 'A', or 'Q'. This is optional if dates are given.

missing : str

Available options are 'none', 'drop', and 'raise'. If 'none', no nan checking is done. If 'drop', any observations with nans are dropped. If 'raise', an error is raised. Default is 'none.'

Methods

fit([maxlag, method, ic, trend, ...])	Fit the unconditional maximum likelihood of an AR(p) process.
from_formula(formula, data[, subset])	Create a Model from a formula and dataframe.
hessian(params)	Returns numerical hessian for now.
information(params)	Not Implemented Yet
initialize()	
loglike(params)	The loglikelihood of an AR(p) process
predict(params[, start, end, dynamic])	Returns in-sample and out-of-sample prediction.
score(params)	Return the gradient of the loglikelihood at params.
select_order(maxlag, ic[, trend, method])	Select the lag order according to the information criterion.

Attributes

endog_names
exog_names

© Copyright 2009-2013, Josef Perktold, Skipper Seabold, Jonathan Taylor, statsmodels-developers. Created using Sphinx 1.2.2.

次数の選定

【FPE と AIC を用いた次数の選定】 観測データ数を N，ARX モデルまたは ARMAX の次数を (n, m) として，次数の選定の指標として次の二つがある。

$$\mathrm{FPE(n,m)} = \frac{N + (n+m+1)}{N - (n+m+1)} \hat{\sigma}_\varepsilon^2 \tag{8.16}$$

$$\mathrm{AIC(n,m)} = N \log \hat{\sigma}_\varepsilon^2 + 2(n+m) \tag{8.17}$$

ここに，$\hat{\sigma}_\varepsilon^2$ は 1 段予測誤差の分散である。すなわち，

$$\hat{\sigma}_\varepsilon^2 = \frac{1}{N} \sum_{i=1}^{N} \left(\varepsilon(k) - \frac{1}{N} \sum_{i=1}^{N} \varepsilon(k) \right)$$

このとき，最少となる FPE(n,m) または AIC(n,m) となる (n,m) が最適次数である。ただし，FPE は 1 段予測誤差が白色である場合にしか適用できない。

左はb0項を含んでいません

この FPE と AIC を求めるのに，変数が n と m の二つあるため，少し実験は大変であるが，$n \geq m$ の制約条件を忘れないようにする。また，1 段予測誤差の白色性の検定は 8.6.4 項で述べる。

- FPE: Final Prediction Error, あまり使えません。
- AIC: Akaike's Information Criterion, 考案した赤池氏は，あくまでも目安ですよ，と述べている。

モデルの評価

実際のシステムに自由に入力を加えられるならば，ステップ応答などを通してモデル評価が行いやすい。しかし，自由に入力を与えられない場合も多い。この場合，得られたモデルが妥当であるかの評価項目は次数の評価と残差の評価がある。

初めに次数の評価を考える。実際の例では，FPE や AIC で選定した次数は多めになることがある。そこで，同定したモデルの極・零点の位置を計算する。ここに，

$$A(z) = 1 + a_1 z^{-1} + \cdots + a_n z^{-n} \tag{8.34}$$

の根が極であり，

$$B(z) = b_1 z^{-1} + \cdots + b_m z^{-m} \tag{8.35}$$

の根が零点である。

b0項を入れていません

用いた次数が真の次数より大きい場合には極・零点の一部が一致するの，これを相殺しなければいけない。この理由は，同定をフィードバック制御で用いる場合，内部不安定につながることがあるためである。

しかし，実システムを同定する場合，極と零点の配置には不確かさが存在すため，完全な相殺は起こらない。したがって，極と零点を複素平面（または，z 平面）にプロットして，視察により相殺が予想される極・零点の対を取り除くのが原始的であるが堅実な方法であろう。

実際には，p≧q として，pを3～9程度から始めてから，試行錯誤する

モデルの評価

次に定義される残差の解析を考える。

$$e(k) = H^{-1}(z)\{y(k) - G(z)u(k)\} \tag{8.36}$$

モデルが同定対象を正確に表しているならば，残差は正規性で，入力と独立になる。そのため，残差の正規性検定は広く利用される。この検定法は，5.2 節にすでに示したとおりである。

pandasを用いたARMA同定

ARMAモデルを求める意義

- 推定と予測，または，対象システムの性質を分析する，制御を行う（ここでは扱わない）
- statsmodelsのARMAを用いて推定や予測を行うには，pandasを用いた時間系列も与えなければならない。

推定

- ARMAモデルを考える。
- 現在時刻k－1において，{y(i)}，(i=k－1，k－2，,,,)が観測できているとする。
- 入力系列{u(i)}は観測できない。もし，i=k－1，k－2,,,,が推定できて，かつ，i=kにおけるu(k)を予測できるならば，y(k)を予測できることになる。

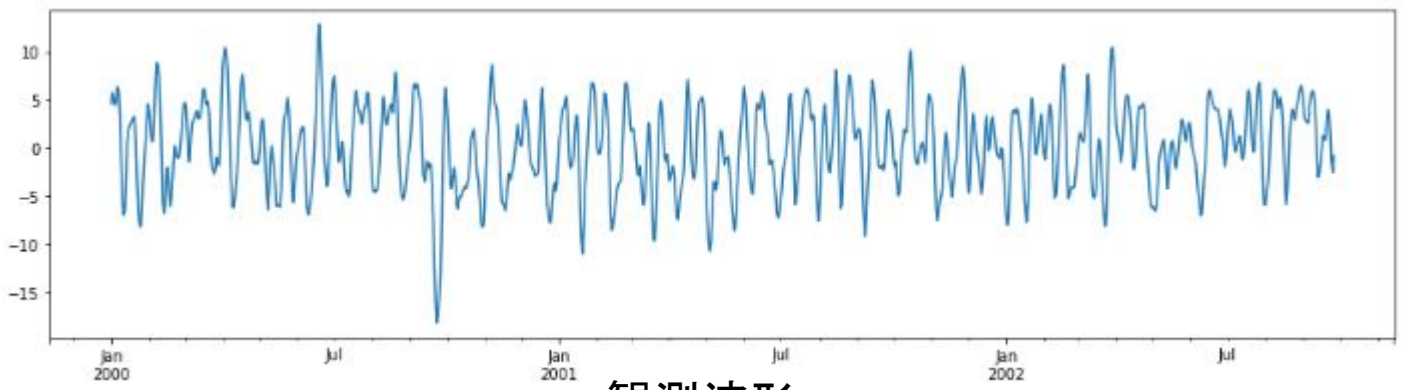
推定と予測

ARMA_Prediction

シミュレーション例

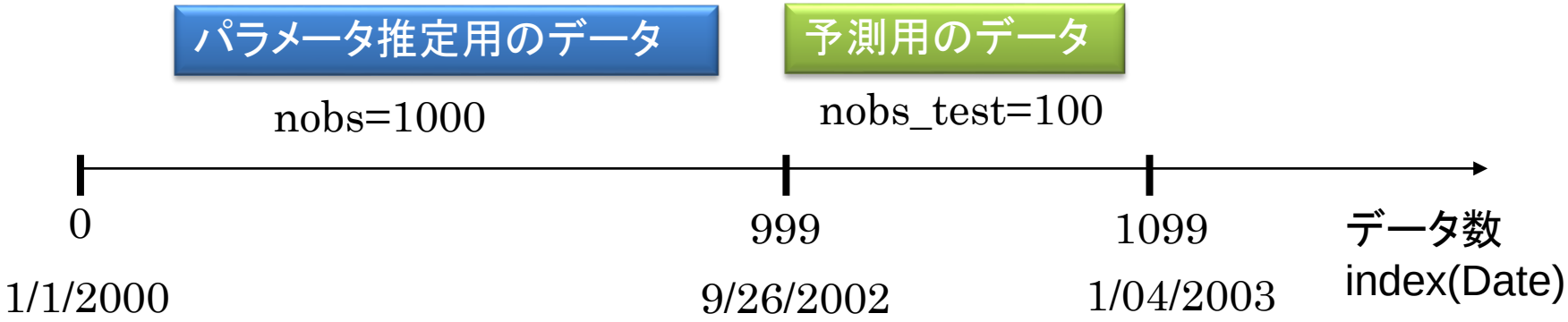
- システムパラメータ, ar = [1, -1.5, 0.7], ma = [1.0, 0.6]
- データ数：nobs = 1000 （the number of observation), パラメータ推定（モデルの同定）用
- データ数：nobs_test, 予測区間, 実データと予測値の比較を行うためのもの
- 入力：平均値0, 分散1の正規性乱数
- date rangeは'1/1/2000'から 時間間隔（frequency）は1日（freq = 'D'）とした

```
nobs = 1000
nobs_test = 100
nobs_all = nobs + nobs_test
ar = [1, -1.5, 0.7]
ma = [1.0, 0.6]
dist = lambda n: np.random.randn(n)  # 正規分布,  引数 n はダミー
y_all = arma_generate_sample(ar, ma, nobs_all, sigma=1, distrvs=dist, burnin=500)
```



観測波形

データ生成の考え方



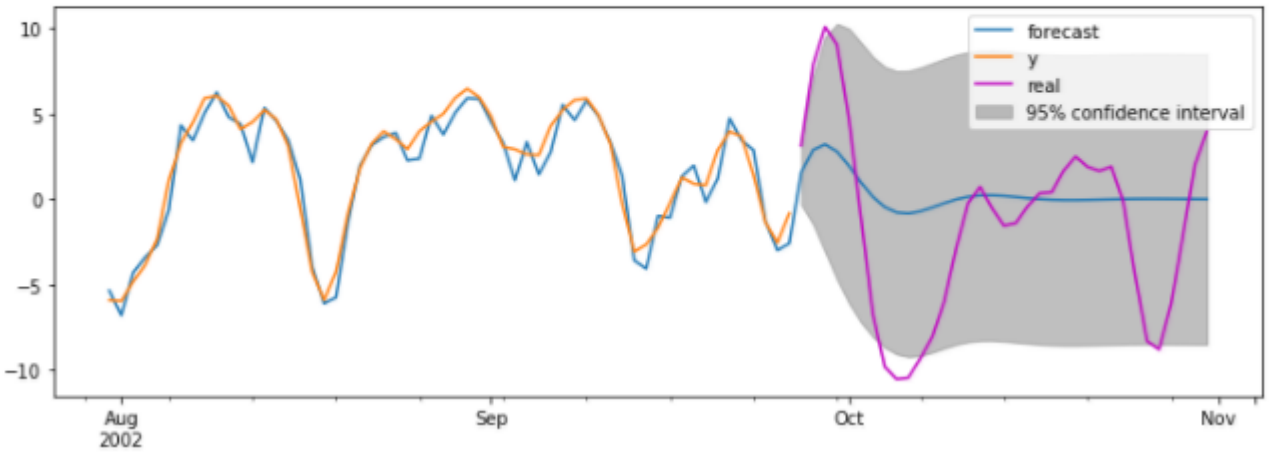
nobs：the number of observation

結果の考察

推定と予測

- ARMAモデルの出力は青色である。予測をforecastと言っている。
- データは 2002-09-26まであり，この範囲内ではk時刻においてy(k)の“推定値”$\hat{y}(k)$を求めている
- このデータの範囲外では“予測値”(forecast)を求めている。ARの次数分だけは予測していて，その先はゼロとならざるを得ない。
- 実際の値(グラフでrealと表示)と比較して，値は一致しない(当然！)，上がるか下がるかという定性的評価では，数ステップ間だけは，この例では(あくまでも！)予測があたっていると言える。
- ARの次数を大きくすれば，その分，先の時刻までの予測は行えるが，予測誤差の分散は，その分だけ大きくなる。注意：予測値の誤差とは言っていない。ARMAはあくまでも確率論で話が成り立っているので，1個1個の値でなく，確率・統計量(分散などをいう)で議論が進めてきたことに注意されたい。

```
1  fig, ax = plt.subplots(figsize=(12,4))
2  fig = arma_result.plot_predict(start='2002-07-31', end='2002-10-31', ax=ax)
3  y_test['2002-09-27':'2002-10-31'].plot(color='m', label='real')
4  legend = ax.legend(loc='upper right')
```



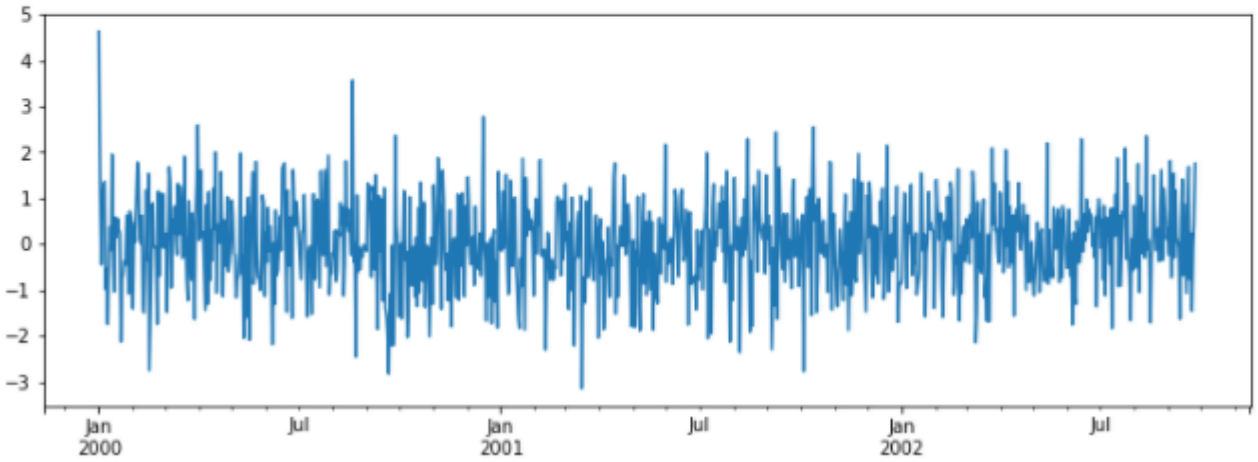
アドバンス：結果の考察

残差系列の評価

- 入力が正規分布に従うならば，という条件下では残差も正規分布に従う。このため，正規分布の検定(test)を行うと，p値は　pvalue=0.06256　，若干高めの感がある。

```
resid = arma_result.resid # residual sequence
resid.plot(figsize=(12,4))
from scipy import stats
print(stats.normaltest(resid))
```

```
NormaltestResult(statistic=5.5432545565920144, pvalue=0.062560119142447537)
```

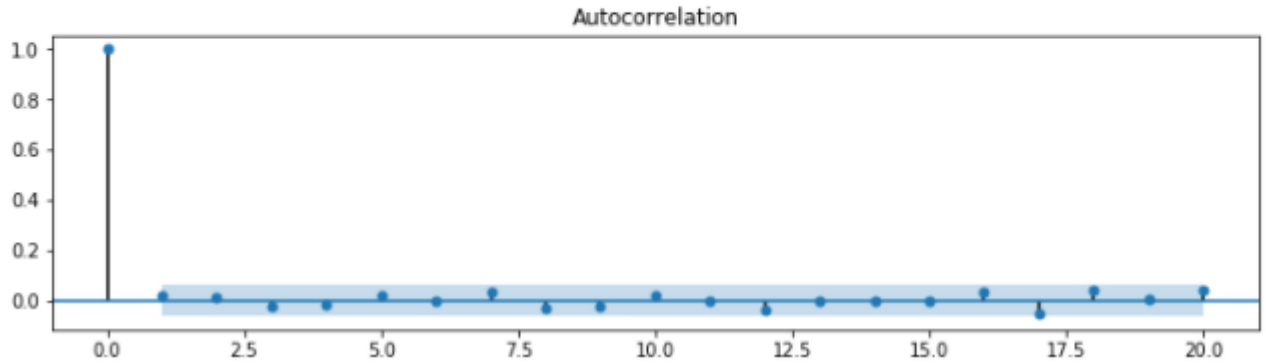


アドバンス：結果の考察

残差系列の検定

- 青色の帯は，有意水準5%（sig_val = 0.05）とすることで，95 %の信頼区間を示している。
- 自己相関関数は，ほぼこの信頼区間に入っている。

```
: 1 fig = plt.figure(figsize=(12,3))
  2 sig_val = 0.05 # 有意水準
  3 ax1 = fig.add_subplot(111)
  4 fig = sm.graphics.tsa.plot_acf(resid.values.squeeze(), lags=20, alpha=sig_val, ax=ax1)
  5 plt.show()
```



おわりに

次数の選定

- AIC, BICは目安程度。赤池氏による導出は，もともと，尤度に基づく残差系列の分散を基にしていた。このとき，パラメータ数が多いほど分散が小さくなる傾向があった。これでは，パラメータ数が多いほど良い，という話になるので，ケチの原理を導入して，パラメータ数が少ない方が良いだろう，という程度の意味であった。ケチの原理を導入したのは，この理論が1970年代というコンピュータ能力（処理速度と記憶容量）が今と比べると格段に低い時代であったことも要因であろう。また，データ数が数十～数百点では，パラメータ同定の信頼性が低いことを示したように，誤差分散を用いるAICやBICの信頼性は，AICやBICが確率変数であることを考えると，試行毎に変化すると考えた方が良い。
- コンピュータ能力がはるかに向上した現在は，ケチになる必要はなくてもよいと考える。
- 時系列データの予測だけならば，残差分散の小さい方が良い方が out of data rangeでの予測は，一般に小さい。
- 制御やディジタルフィルタのように，入力を与えられる場合には，極・零相殺の評価を行うべきである。これは，モデルやシステムを含んだフィードバックループでの内部不安定の危険性を回避するためである。
- また，対象システムの数式で表現される物理モデルがわかっている場合，それに合わせた次数にするべきであろう。その場合，観測データ数が少ないと，同定だけで良いパラメータ値が得られない。この場合，物理的または工学的考察を通して，パラメータ値を同定とは別の方法でチューニングしてもよい。
- AIC, BICを基にした次数選定は総当たり戦が必要。これを行う関数に
  statsmodels.tsa.stattools.arma_order_select_ic
  がある。この使用法は各自で確かめられたい。

1. 赤池、中川：ダイナミックシステムの統計的解析と制御、サイエンス社、1972
2. 赤池：情報量規範AICとは何か、数理科学, vol.14, no.3, pp.5-11, 1976]
3. 赤池記念館, 統計数理研究所, http://www.ism.ac.jp/akaikememorial/index.html
4. J.S.ベンダット, A.G.ピアソル著、得丸、他共訳：ランダムデータの統計的処理、培風館、1976
5. 相良、中溝、秋月、片山：システム同定、計測自動制御学会、1987
6. 足立：ユーザのためのシステム同定理論、計測自動制御学会、1993
7. 足立：システム同定の基礎、東京電機大学出版局、2009

SARIMAの参考文献

1. Seasonal ARIMA Models, Eberly College of Science, PennState https://onlinecourses.science.psu.edu/stat510/node/50, Chap.4.1, ペンシルベニア州立大学（同校はペン・ステート（Penn State）という愛称で親しまれている） Eberly 科学学部, Robert E. Eberly氏の名前に由来する。
2. SARIMAX: Introduction http://www.statsmodels.org/dev/examples/notebooks/generated/statespace_sarimax_stata.html
3. 初出は「Box, G. E. P., Jenkins, G. M. and Reinsel, G. C. (1976) Time Series Analysis, Forecasting and Control. Third Edition. Holden-Day. Series G.」

アドバンス

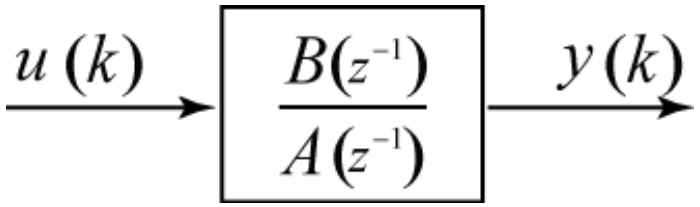
以下は，授業では説明しません。

ただし，ARMAを正しく扱いたいときには，必須の内容ゆえ，
必要な人は学ぶべき内容です。

ARMAモデルの安定性と性質

モデルの見方

- 制御論でいうブロック線図を示す
- このブロックは伝達関数と呼ばれる
- 分母にあるAR項がモデルの安定性を支配する。
- また，モデルの性質に大きな影響を与える。
  ⇒　極配置とインパルス応答で述べる
- AR(z-1)=0とおいたときの根（root）を極(pole)という
- MA(z-1)=0とおいたときの根を零点（zero）という
- 備考：
  - n次の多項式の根はn個である（重根含む）。
  - さらに，実係数（実数の係数）ならば，この根は複素共役か実数に限られている。複素共役はペア（2対）であるから，例えば，3次ならば，実数根が3つ，複素共役のペアが1つ（根は2つ）と実数根が1つ，この2通りしかない。



AR_Coefficient_Root

モデルの安定性と性質

- 安定性は極に依存する
- 性質は極と零点に依存する

モデルの安定性

- 複素平面上（ガウス平面，横軸が実軸（実数軸），縦軸が虚軸（虚数軸））の単位円（unit circle, 中心が原点，半径1の円）を考える
- 単位円内に極があれば安定，無ければ不安定
- 次ページ以降で確かめる

システムの性質を知る

システム構造が未知の場合

- 何らかの入力を与え，その出力を見る。システム構造をよく表す入力とは？
- インパルス応答
  - 時間応答
  - システムの伝達関数またはカーネルを良く表す。
  - しかし，理想的なインパルスは無限大のパワーであるから，離散時間システムでは，単位インパルスを与えることが多い
- 白色雑音
  - システムの伝達関数を良く表す。
  - システムに負荷を与えることがあるため，疑似乱数が与えることが多い。

システム構造が既知の場合

- システムを数式表現して，その解析を行うことが一般的
- 制御や信号処理分野では，その解析法や評価法が充実している

離散時間インパルス応答

- ARMAの単位インパルス応答を見る
  - 遅延演算子を理解する
  - インパルス応答とは何かを理解する
- 次を対象とする

$$y(k) = -\left(a1\ y(k-1) + a2\ y(k-2) \right) + b0\ u(k) + b1\ u(k-1),$$

where b0=1とおいて, $u(k) = 1$ if $k = 0$, $u(k) = 0$ if $k > 0$

これを図で表現すると？

AR_ImpulseResponse

```
num=10
ma = np.array([1.0, 0.0])  # MAモデルのb0, b1 を与える。b0 = 1は単位インパルスを実現するため
ar = np.array([1.0, -0.5]) # ARモデルのa0, a1 を与える。
tresp = arima.arma_impulse_response(ar, ma, nobs=num)
print(tresp)

[ 1.  0.5  0.25  0.125  0.0625  0.03125  0.015625  0.0078125  0.00390625  0.00195312]
```

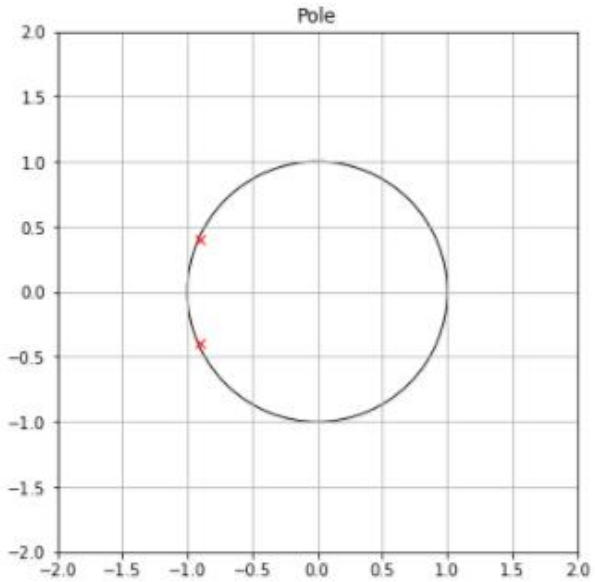
極配置とインパルス応答

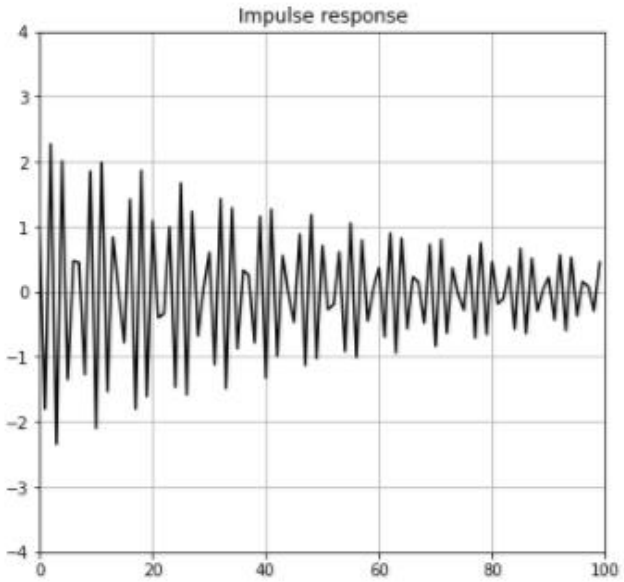
極の配置と応答の関係を調べる

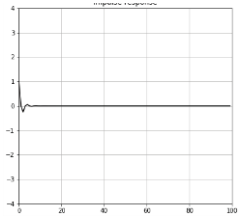
- AR(2), MA = 1 とする。ARの極の影響だけを見たいため。
- プログラムで，幾つかの配置に対する応答がどのようになるかを調べる
- 安定領域内で
  - 左半平面は，実数が負のため，激しい振動的（応答の符号が反転しながら進むため）
  - 右半平面は，実数が正のため，包絡線の減衰は比較して緩やか
  - 原点から縦軸（虚軸）方向に離れるほど，振動成分が大きくなる。
- 不安定領域
  - 発散する

インパルス応答

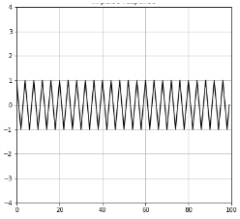
- k = 0で，u(k) = 1，　k > 0 でu(k)=0 を与えたときのy(k)の応答



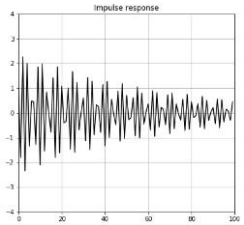




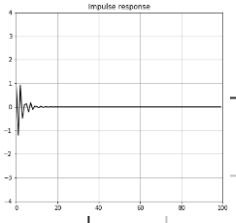
[9]



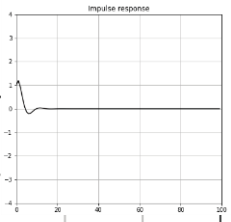
[10]



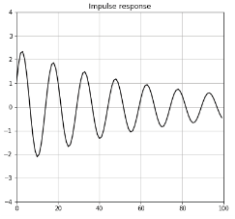
[2]



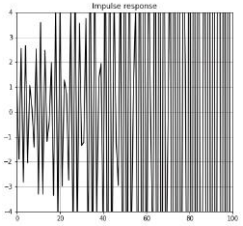
[3]



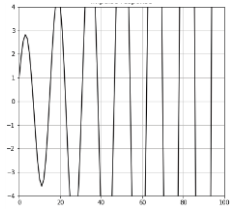
[4]



[5]

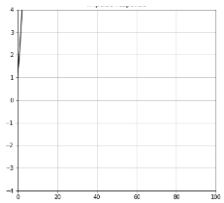


[1]



[6]

[7]



[8]

零点の影響

安定性には関係しないが，応答の性質に影響する

- 安定零点は単位円内，不安定零点は単位円外，名前と反して安定性には影響しない。
- ただし，制御を行おうとしてフィードフォワード系やフィードバック系を構成すると，MA部の不安定零点が安定性に影響を与えることがある。
- 零点が単位円外にあると，インパルス応答において，過大な振幅を一瞬示す。
- また，零点が右半平面にあると，逆振れが生じやすい。逆振れとは，例えば，出力の初期値が正のとき，次のステップ時刻では負に振れることをいう。
- 過大な振幅や逆振れは，システムに負担をかけることになり，あまり望ましいとは言えない。

ARMAモデルのパラメータ推定

statsmodelsのデータ生成を用いたシミュレーション例

- データを人工的に発生させ，これに対するパラメータ推定を行う
- ここで，データを発生するものをシステム，データからパラメータ推定が施されるものをモデルと称する。
- データの発生は次を用いる。

arma_generate_sample

- from statsmodels.tsa.arima_process import arma_generate_sample
- 引数の説明：
- ar = np.array([a0, a1,,,an])
- ma = np.array([b0, b1,,, bm])
- 注意：AR，　MAともに a0, b0 項を与えている。
- 入力はディフォルトで標準正規分布 N(0,1)を与えている。もし，他の入力を与えたい場合には，次のように，Pythonで定められている無名関数（lambda）を用いて
  - dist = lambda n: np.random.standard_t(3, size=n) # t 分布, n はダミー引数で実際に与えなくてもよい，これを用いて，引数に distrvs = dist を与える。
- burin = nnn，ARMAシステムの各変数y, uの初期値は全て0である。このシステムの駆動時，過渡現象が生じて，定常状態とは異なる様相を示すため，この過渡現象期間のデータを破棄することが必要である。この破棄するデータ数をburinで指定し，この後のデータを出力する。

ARMAモデルのパラメータ推定

- パラメータ推定を行いモデルを作るのが次である
- ARMA()
  - from statsmodels.tsa.arima_model import ARMA
  - model = ARMA(x, order = (n,m)).fit(trend = 'nc')
  - order = (n,m)と与えたとき，出力されるパラメータは {a1, a2, ,,, an}, (b1, b2, ,, bm}であり， a0とb0は1とされて出力されない。
  - このことは，arma_generate_sampleではa0, b0を必要としたことと異なることに注意
  - trend ='nc'はバイアスが無い場合，'c'はバイアスがある場合，よって，trendとは言っているが，定数のバイアスだけを対象としているので，後にデータの前処理で述べるように， 常にtrend ='nc'として構わない。
  - model.paramsで，ARのパラメータ値は符号が反転して出力される。これは，アルゴリズム上の都合である（付録参照）。符号が反転されるのだから，最初から符号反転してデータ生成すればいいだろう，という考えは誤りである。パラメータ値の符号を反転させると，極の位置が異なり，性質の異なるARシステムとなる。

注意点

ARのパラメータは符号を反対にして求める

- http://www.statsmodels.org/ → [Examples] → Topics の[Time Series Analysis] → [ARMA: Artificial Data] （ここの図をクリック），例題プログラムが現れます。ここに，
  - arparams = np.array([.75, -.25])
  - arparams = np.r_[1, -arparams]
- ここに，マイナス演算子を付加して，パラメータの符号を反転してARモデルに与えている。この例題に従って，必ず符号を反転させる，とは思わないでください。
- 符号を反転させると，当然，極の位置が変わり，ARモデルの特性が変わります。このことは，先のインパルス応答のプログラムでご自身で確認してみてください。
- では，なぜ，符号反転を行っているか？
- ARモデルのパラメータを求めるアルゴリズムにおいて
  - y[k] + a_1 y[k-1] + a_2 y[k-2] = MA(z^{-1}) u[k]
- これを次のようにして数値計算を行っている。
  - y[k] = − a_1 y[k-1] − a_2 y[k-2] + MA(z^{-1}) u[k]
- すなわち，− a_1 , − a_2 を符号付きで求めます。このため，ARのパラメータは，見掛け上，求めたパラメータは符号が反転しています。このため，数値計算で求めたARのパラメータは，符号を反転してください。
- MAのパラメータは，アルゴリズムより，符号反転は生じません。アルゴリズムの詳細は他書を参照してください（「システム同定」関連本）。

例 ARMA(2,1)のパラメータ同定

ARMA(2，1)の出力データの生成

ARMA_ParameterEstimation

- 下記より，安定である
- データ数は10000点ある
- ARMAモデルの入力は標準正規分布に従うランダム変数である。

```
nobs = 10000
ar = [1, -1.5, 0.7]
ma = [1.0, 0.6]
dist = lambda n: np.random.randn(n)  # 正規分布，  引数 n はダミー
np.random.seed(123)
y = arma_generate_sample(ar, ma, nobs, sigma=1, distrvs=dist, burnin=500)
```

パラメータ同定の仕方

```
arma21 = ARMA(y, order=(2,1)).fit( trend='nc' )
print('arma21')
print('parameters = ',arma21.params,'  , AIC =', arma21.aic)
```

例 ARMA(2,1)の同定

結果

- ARMA(2,1)モデルのパラメータ同定が，当然ながら，最も真値に近い。ただし，ARの係数は反転して見ること。
- 観測データ数をこれだけ得られない場合には，推定パラメータ値は，モデル間でかなりばらつきがある。
- 入力が観測できれば，nobs=1000でも，推定したパラメータの精度は上がるが，これは望めない以上，同定精度には注意が必要である。
- 同定のアルゴリズムは収束計算を行うため，計算時間を要するだけでなく，数値計算上の計算誤差を推定したパラメータに含むことも念頭におくと良い。
- AICを見ると，ARMA(3,2)であり，真のシステムとは異なる。これは，データ数が高々10000点という有限のデータ数で，かつ，入力が観測できない以上，同定精度が望めないためである。このため，出力誤差（残差）の分散値は試行ごとにばらつく。
- このため，AICの値もばらつく。有限データ長を扱う以上，AICは確率変数であり，その値は揺れる。よって，AICは目安であること考えてよい。

```
arma20
parameters =  [ 1.64234146 -0.82610482]   , AIC = 30935.479229234246
arma21
parameters =  [ 1.49623143 -0.69516878  0.60450373]   , AIC = 28368.324505251763
arma32
parameters =  [ 1.49091126 -0.70283206  0.01274749  0.62190495  0.03152872]   , AIC = 28363.8089374829
arma43
parameters =  [ 2.31756458 -1.82692641  0.42961809  0.06692595 -0.20498307 -0.58968237
 -0.08836438]   , AIC = 28364.815635767445
```

例 ARMA(2,1)の同定結果


```
print('arma21----------summary-------------------')
print(arma21.summary())

arma21----------summary-------------------
                              ARMA Model Results
==============================================================================
Dep. Variable:                      y   No. Observations:                10000
Model:                     ARMA(2, 1)   Log Likelihood              -14180.162
Method:                       css-mle   S.D. of innovations              0.999
Date:                Mon, 05 Mar 2018   AIC                          28368.325
Time:                        07:31:58   BIC                          28397.166
Sample:                             0   HQIC                         28378.087

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1.y        1.4962      0.008    190.755      0.000       1.481       1.512
ar.L2.y       -0.6952      0.008    -89.044      0.000      -0.710      -0.680
ma.L1.y        0.6045      0.009     70.606      0.000       0.588       0.621
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.0762           -0.5295j            1.1994           -0.0728
AR.2            1.0762           +0.5295j            1.1994            0.0728
MA.1           -1.6542           +0.0000j            1.6542            0.5000
-----------------------------------------------------------------------------
```

statsmodels.tsa.arima_model.ARMAResults（http://www.statsmodels.org/dev/generated/statsmodels.tsa.arima_model.ARMAResults.html ）より，z^{^1}の根を求めており，この場合，安定極は単位円外となる。これまでの説明は，zの根であり，この安定極は単位円内である。上記の極は z^{-1}の逆数を求めると，単位円内に極が入る。

モデルの評価

極

- Roots of z：　　[0.74811572+0.36809191j 0.74811572-0.36809191j]　より，安定で振動的なシステムと推定できる。
- また，MAとは共通因子（根）を有さない

残差検定

- 入力は，PE性の条件を満足すればパラメータ同定は行えることを既に述べた。
- しかし，入力は観測できないという条件で説明している。
- 制御やシステム同定の場合，入力は，現場適用可能な信号としていて，観測雑音w(k)を正規性確率変数としていた
- 一方，入力を観測できない分野では，w(k)=0とする代わりに入力u(k)を白色雑音と仮定することが多い。これは，自己相関関数を用いた表現や周波数領域での解析が簡単になるためである。
- 注意：正規性の確率変数とは言っていない。これは理論展開するとき，ウィーナーヒンチン定理より，自己相関関数がデルタ関数になればいいだけのためである。
- また，時刻kにおけるy(k)の推定値 $\hat{y}(k)$を求めるのに，b0項の存在を仮定していたため，u(k)が必要である。このため，次の工夫を行っている
- 残差系列ε(k) = y(k) − $\hat{y}(k)$を1ステップ前のu(k−1)としている。
- 以上の仮定の下で，残差系列ε(k)は白色雑音であろうという推測のもと，この白色性検定を行うことが多い。
- ウィーナーヒンチン定理に基づき，残差系列ε(k)は白色雑音であるならば，その自己相関関数はディラックのデルタ関数になる。
- しかし，実際にはそのような理想状態を得ることはできないため，以下のように考える

モデルの評価

残差系列（residual time series）の白色性検定

- 残差系列が白色性信号ならば，この自己相関関数がディラック関数である
- 参考：足立：ユーザのためのシステム同定理論，計測自動制御学会，1993

プログラムの説明

ARMA_ParameterEstimation

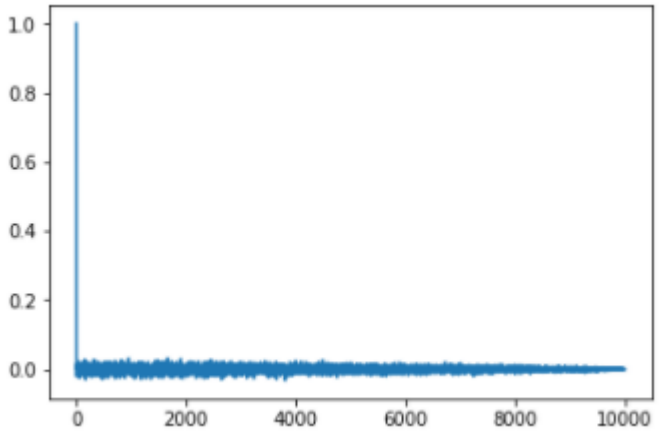
- resid は　residualの省略語
- np.correlateは，両側相関関数を求める（ラグがマイナスの場合も求めるということ）
- auto_corrの配列の中心から右側だけで十分である。
- AutoR[0] = 1（最大値）となるように正規化している。
- AutoR[i], i ≧1 において，次ページ以降で述べる検定を実施する。

```
resid = arma21.resid
auto_corr = np.correlate(resid, resid, mode='full')
center = int(len(auto_corr)/2)
AutoR = auto_corr[center:]/np.max(auto_corr)
plt.plot(AutoR)
count = 0
for i in np.arange(1,len(AutoR)-1):
    if np.abs(AutoR[i]) >= 2.17/np.sqrt(len(AutoR)):
        count += 1
#        print("Warning", i, AutoR[i])
print('count = ', count, ' len(AutoR) = ', len(AutoR), '  rate =', count/len(AutoR))
print('max(AutoR[i], i >= 1)', np.max(AutoR[1:]))
```

結果

検定結果

- 0.77%の自己相関が大きな値を示した。　経験的に2～3%以内ならばよしとする（人為的）。
- その最大値は 0.0315であった。　経験的に0.02～0.03ならよしとする（人為的）。
- 厳格な検定ならば棄却であるが，ほどほど白色性信号とみなすこともある。
- その自己相関関数のグラフ




```
count =  77  len(AutoR) =  10000   rate = 0.0077
max(AutoR[i], i >= 1) 0.0315401145424
```

偏相関関数（PARCOR)

- 残差系列に対して，統計学で言うところの偏相関関数（Partial Correlation Function, よくPARCORと称される）を見ることも考えられる。PARCORは，パラメータ同定のアルゴリズム上で現れるものである。他の評価法で十分であるため，本評価を用いることはあまりない。

J.D.Markel, A.H.Gray(著), 鈴木久喜(約):音声の線形予測, コロナ社, 1981

S.M.Kay and S.L.Marple: Spectrum Analysis – A Modern Perspective, Proc. of the IEEE, vol.69, no.11, pp.1380-1419,1981

金井浩:音・振動のスペクトル解析, コロナ社, 1999

考察

観測データ数

- 今回は人工データゆえ，データ数（nobs）を自由に設定できたので，10000点とした。
- このデータ数でも真のパラメータに完全一致させることは難しい。
- センサを用いた観測では，1万から10万点オーダは普通に得られることが多い。しかし，観測雑音が重畳するので，このオーダのデータ数でも真値に一致させることは難しいことが多い。
- 取引市場データや人口問題のように観測雑音が無い場合，データ数が多くても数千点がせいぜいの場合が多い。
- また，生物系や医薬系の場合には，さらにデータ点数が少ない場合がある。
- よって，推定パラメータが真値に一致しているかどうかを気にしても仕方が無いことである。そよりも，出力の誤差の統計的性質だけを評価するのがよいと考える。
- ただし，入力を観測できる場合で，制御や信号処理を行う場合には，出力誤差分散よりも時間応答や周波数応答がどれだけ良好な近似であるかに評価の重点を置く。

私見

- 入力を観測できないシステム同定では，入力を推定することとなる。
- では，ある時刻kのときのu(k) と $\hat{u}(k)$が一致することはまずなく，その統計的な性質だけを近づけようとしているだけに過ぎない。
- ならば，いっそ，MA項をあきらめて，観測できるy(k)を活用するARモデルとして，この次数を多少大きめにしたモデルでも十分に活用できることになる。この実験的評価は読者の手に委ねる。実験では，ARMA（n,0）とすればよいだけである。

次数の選定，極・零点消去法

対象システム

- ARMA(3,2), ar = [1, -2.0, 1.7, -0.5],　　ma=[1.0, -1.5, 0.685]

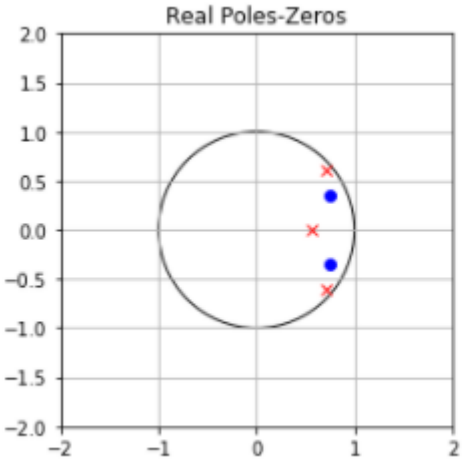
推定モデル

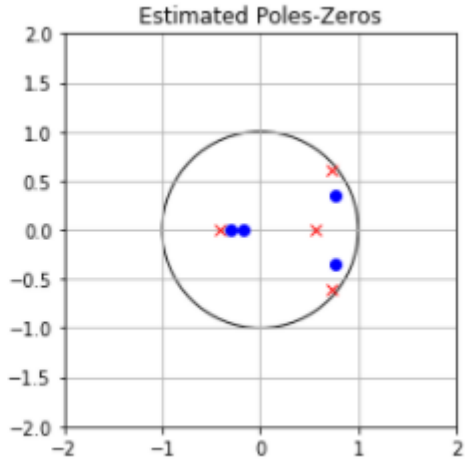
- ARMA(4,4)
- nobs = 10000

ARMA_PoleZeroCancellation

同定結果

- パラメータ値はnotebookを参照
- 極（赤の×），零点（青の〇）を見て，推定した値（右図）で，実軸上の×を一つ，〇を2個が近いと判断して，この三つを除外すると，ARMA（3,2）となる。残った極，零点から改めて係数を求めればよい。





次数の選定

AIC

- この最小値を探索する方法は，人為的判断を含まず定量的に決定できる点で優れているが，次数（p,q）のあらゆる組合せを考慮する必要があるため，計算時間と労力を必要とする。ただし，AICの計算で使用する誤差分散が確率変数であり，かつ，有限データ数の中で求めるわけだから，AICが最小だからといってベストフィットしているわけではないことを念頭においておく必要がある。

極・零点消去法

- 大きめの次数で同定しておいてから，極零相殺を行えばよいので，計算時間と労力の点では魅力的である。しかし，相殺するための近さがどれだけであるかは，人為的判断による恣意性から免れることはできない。また，当然，パラメータ推定の精度が高い，という前提があることを念頭においておく必要がある。

物理モデルの考察

- 工学の分野では，古くから連続時間で表される物理モデルが確立されているものが多い。このモデルの離散化作業を通して次数を物理的観点から決めるというのが，その後の学術的考察を自然に行えるという点から望ましい。