Modeling and Analysis of Time Series Data
Chapter 5: Parameter estimation and model identification for ARMA models

STATS 531, Winter 2026

Edward Ionides

Background on likelihood-based inference

- For any data $y^*_{1:N}$ and any probabilistic model $f_{Y_{1:N}}(y_{1:N};\theta)$ we define the likelihood function to be
$$L(\theta) = f_{Y_{1:N}}(y^*_{1:N};\theta).$$

- It is often convenient to work with the logarithm to base e of the likelihood, which we write as
$$\ell(\theta) = \log L(\theta).$$

- Using the likelihood function as a statistical tool is a very general technique, widely used since Fisher (1922) (Wikipedia: Likelihood function).

- Time series analysis involves various situations where we can, with sufficient care, compute the likelihood function and take advantage of the general framework of likelihood-based inference.

- Computation of the likelihood function for ARMA models is not entirely straightforward.

- Computationally efficient algorithms exist, using a state space model representation of ARMA models that will be developed later in this course.

- For now, it is enough that software exists to evaluate and maximize the likelihood function for a Gaussian ARMA model. Our immediate task is to think about how to use that capability.

- Before evaluation of the ARMA likelihood became routine, it was popular to use a method of moments estimator called **Yule-Walker** estimation [2, Section 3.5]. This is nowadays mostly of historical interest.
- For massively long time series data and big ARMA models, it can be computationally infeasible to work with the likelihood function. However, we focus on the common situation where we can (with due care) work with the likelihood.
- Likelihood-based inference (meaning statistical tools based on the likelihood function) provides tools for parameter estimation, standard errors, hypothesis tests and diagnosing model misspecification.
- Likelihood-based inference often (but not always) has favorable theoretical properties. Here, we are not primarily concerned with the underlying theory. On any practical problem, we can check the properties of a statistical procedure by simulation experiments.

The maximum likelihood estimator (MLE)

- A maximum likelihood estimator (MLE) is

$$\widehat{\theta}(y_{1:N}) = \arg\max_{\theta} f_{Y_{1:N}}(y_{1:N};\theta),$$

  where $\arg\max_\theta g(\theta)$ means a value of argument θ at which the maximum of the function g is attained, so $g(\arg\max_\theta g(\theta)) = \max_\theta g(\theta)$.
- The maximum likelihood estimate (also known as the MLE) is

$$\begin{aligned}\widehat{\theta} &= \widehat{\theta}(y^*_{1:N}) \\ &= \arg\max_{\theta} L(\theta) \\ &= \arg\max_{\theta} \ell(\theta).\end{aligned}$$

- If there are many values of θ giving the same maximum value of the likelihood, then an MLE still exists but is not unique.

Question. Why are $\arg\max_\theta L(\theta)$ and $\arg\max_\theta \ell(\theta)$ the same?

- We can write $\hat{\theta}_{\text{MLE}}$ to denote the MLE if we are considering various alternative estimation methods. However, in this course, we will most often be using maximum likelihood estimation so we let $\hat{\theta}$ correspond to this approach.

Standard errors for the MLE

- As statisticians, it would be irresponsible to present an estimate without a measure of uncertainty!
- Usually, this means obtaining a confidence interval, or an approximate confidence interval.
- It is good to say **approximate** when you present something that is not exactly a confidence interval with the claimed coverage. For example, remind yourself of the definition of a 95% confidence interval.
- Saying "approximate" reminds you that there is some checking that could be done to assess how accurate the approximation is in your particular situation.

Three ways to quantify statistical uncertainty in an MLE

1. Fisher information. This is computationally quick, but works well only when $\hat{\theta}(Y_{1:N})$ is well approximated by a normal distribution.
2. Profile likelihood estimation. This is a bit more computational effort, but generally is preferable to the Fisher information.
3. A simulation study, also known as a bootstrap.

Standard errors via the observed Fisher information

- We suppose that $\theta \in \mathbb{R}^D$ and so we can write $\theta = \theta_{1:D}$.
- The **Hessian matrix** of a function is the matrix of its second partial derivatives. The Hessian matrix of the log likelihood function is $\nabla^2 \ell(\theta)$, a $D \times D$ matrix with (i, j) element

$$[\nabla^2 \ell(\theta)]_{ij} = \frac{\partial^2}{\partial \theta_i \partial \theta_j} \ell(\theta).$$

- The **observed Fisher information** is $\hat{I} = -\nabla^2 \ell(\hat{\theta})$.

- A standard asymptotic approximation to the distribution of the MLE for large N is

$$\hat{\theta}(Y_{1:N}) \approx N[\theta, \hat{I}^{-1}],$$

where θ is the true parameter value. This asserts that the MLE is asymptotically unbiased, with variance asymptotically attaining the Cramer-Rao lower bound.

- Since the MLE attains the Cramer-Rao lower bound, under regularity conditions, it is **asymptotically efficient**.
- We can interpret $\approx$ in the above normal approximation to mean "one could write a limit statement formally justifying this approximation in a suitable limit." Almost equivalently, $\approx$ can mean "this approximation is useful in the finite sample situation at hand."
- A corresponding approximate 95% confidence interval for θ_d is $\hat{\theta}_d \pm 1.96([\hat{I}^{-1}]_{dd})^{1/2}$. The Python ARIMA fitting function computes standard errors for the MLE of an ARMA model in this way.
- In practice, we have one time series, with some fixed N, and so we cannot take $N \to \infty$. When our time series model is non-stationary it may not be clear what it would mean to take $N \to \infty$. Asymptotic results are a nice mathematical reason to consider computing an MLE, but not a substitute for checking how the MLE behaves for our model and data.

Confidence intervals via the profile likelihood

- We consider the problem of obtaining a confidence interval for θ_d, the dth component of $\theta_{1:D}$.
- The **profile log likelihood function** of θ_d is defined to be

$$\ell_d(\theta_d) = \max_{\phi \in \mathbb{R}^D : \phi_d = \theta_d} \ell(\phi).$$

In general, the profile likelihood of one parameter is constructed by maximizing the likelihood function over all other parameters.

- Check that $\max_{\theta_d} \ell_d(\theta_d) = \max_{\theta_{1:D}} \ell(\theta_{1:D})$. Maximizing the profile likelihood $\ell_d(\theta_d)$ gives the MLE, $\hat{\theta}_d$.
- An approximate 95% confidence interval for θ_d is given by

$$\{\theta_d : \ell(\hat{\theta}) - \ell_d(\theta_d) < 1.92\}.$$

- This is known as a profile likelihood confidence interval.

Where does the 1.92 cutoff come from

- The cutoff 1.92 is derived using **Wilks's theorem**, which we will discuss in more detail when we develop likelihood ratio tests.
- Note that $1.92 = 1.96^2/2$.
- The asymptotic justification of Wilks's theorem is the same limit that justifies the Fisher information standard errors.
- Profile likelihood confidence intervals tend to work better than Fisher information confidence intervals when the log likelihood function is not close to quadratic near its maximum. This is more common when N is not large.

A simulation study, also called bootstrap

- If done carefully and well, this can be the best approach.
- A confidence interval is a claim about reproducibility. You claim, so far as your model is correct, that on 95% of realizations from the model, a 95% confidence interval you have constructed will cover the true value of the parameter.
- A simulation study can check this claim directly.
- The simulation study takes time to develop and debug, time to explain, and time for the reader to understand and check what you have done. We usually carry out simulation studies to check our main conclusions only.

Bootstrap methods for constructing standard errors and confidence intervals

- Suppose we want to know the statistical behavior of the estimator $\widehat{\theta}(y_{1:N})$ for models in a neighborhood of the MLE.
- In particular, let's consider the problem of estimating uncertainty about θ_1, the first component of the vector θ.
- We use simulation to assess the behavior of the maximum likelihood estimator, $\widehat{\theta}_1(y_{1:N})$, and possibly the coverage of an associated confidence interval estimator, $[\widehat{\theta}_{1,\mathrm{lo}}(y_{1:N}), \widehat{\theta}_{1,\mathrm{hi}}(y_{1:N})]$.
- The confidence interval estimator could be constructed using either the Fisher information method or the profile likelihood approach.

- We can design a simulation study to address the following goals:

(A) Evaluate the coverage of a proposed confidence interval estimator, $[\hat{\theta}_{1,\mathrm{lo}}, \hat{\theta}_{1,\mathrm{hi}}]$,

(B) Construct a standard error for $\hat{\theta}_1$,

(C) Construct a confidence interval for θ_1 with exact local coverage.

A simulation study

1. Generate J independent Monte Carlo simulations,

$$Y_{1:N}^{[j]} \sim f_{Y_{1:N}}(y_{1:N}; \hat{\theta}) \text{ for } j \in 1:J.$$

2. For each simulation, evaluate the maximum likelihood estimator,

$$\hat{\theta}^{[j]} = \hat{\theta}(Y_{1:N}^{[j]}) \text{ for } j \in 1:J,$$

and, if desired, the confidence interval estimator,

$$[\hat{\theta}_{1,\mathrm{lo}}^{[j]}, \hat{\theta}_{1,\mathrm{hi}}^{[j]}] = [\hat{\theta}_{1,\mathrm{lo}}(Y_{1:N}^{[j]}), \hat{\theta}_{1,\mathrm{hi}}(Y_{1:N}^{[j]})].$$

3. For large J, the coverage of the proposed confidence interval is well approximated, for models in a neighborhood of $\hat{\theta}$, by the proportion of the intervals $[\hat{\theta}_{1,\mathrm{lo}}^{[j]}, \hat{\theta}_{1,\mathrm{hi}}^{[j]}]$ that include $\hat{\theta}_1$.

4. The sample standard deviation of $\{\hat{\theta}_1^{[j]}, j \in 1:J\}$ is a natural standard error to associate with $\hat{\theta}_1$.

Model selection for ARMA models

- **Likelihood ratio tests** are appropriate for **nested hypotheses**.
- The whole parameter space on which the model is defined is $\Theta \subset \mathbb{R}^D$. Suppose we have two nested hypotheses

$$\begin{aligned} H^{\langle 0 \rangle} &: \theta \in \Theta^{\langle 0 \rangle}, \\ H^{\langle 1 \rangle} &: \theta \in \Theta^{\langle 1 \rangle}, \end{aligned}$$

defined via two nested parameter subspaces, $\Theta^{\langle 0 \rangle} \subset \Theta^{\langle 1 \rangle}$, with respective dimensions $D^{\langle 0 \rangle} < D^{\langle 1 \rangle} \leq D$.

- We consider the log likelihood maximized over each of the hypotheses,

$$\begin{aligned} \ell^{\langle 0 \rangle} &= \sup_{\theta \in \Theta^{\langle 0 \rangle}} \ell(\theta), \\ \ell^{\langle 1 \rangle} &= \sup_{\theta \in \Theta^{\langle 1 \rangle}} \ell(\theta). \end{aligned}$$

- A useful approximation is that, under the hypothesis $H^{\langle 0 \rangle}$,

$$\ell^{\langle 1 \rangle} - \ell^{\langle 0 \rangle} \approx (1/2)\chi^2_{D^{\langle 1 \rangle} - D^{\langle 0 \rangle}},$$

  where χ^2_d is a chi-squared random variable on d degrees of freedom and $\approx$ means "is approximately distributed as."

- We will call this the **Wilks approximation**.
- The Wilks approximation can be used to construct a hypothesis test of the null hypothesis $H^{\langle 0 \rangle}$ against the alternative $H^{\langle 1 \rangle}$.
- This is called a **likelihood ratio test** since a difference of log likelihoods corresponds to a ratio of likelihoods.
- When the data are iid, $N \to \infty$, and the hypotheses satisfy suitable regularity conditions, this approximation can be derived mathematically and is known as **Wilks's theorem**.
- The chi-squared approximation to the likelihood ratio statistic may be useful, and can be assessed empirically by a simulation study in situations not covered by a known theorem.

Using a likelihood ratio test to construct profile likelihood confidence intervals

- Recall the duality between hypothesis tests and confidence intervals: $\theta^{\langle 0 \rangle}$ is in a 95% confidence interval for θ if and only if we fail to reject a null hypothesis of $\theta = \theta^{\langle 0 \rangle}$ at the 5% level.
- We can check what the 95% cutoff is for a chi-squared distribution with one degree of freedom:

```
print("Chi-square 95% cutoff:",
    f"{stats.chi2.ppf(0.95, df=1):.3f}")
```

```
Chi-square 95% cutoff: 3.841
```

- The Wilks approximation suggests a confidence interval constructed from parameter values having a profile likelihood within 1.92 log units of the maximum.

Akaike's information criterion (AIC)

- Likelihood ratio tests provide an approach to model selection for nested hypotheses, but how about when models are not nested?
- A more general approach is to compare likelihoods of different models by penalizing the likelihood of each model by a measure of its complexity.
- Akaike's information criterion **AIC** is given by

$$\text{AIC} = -2 \times \ell(\hat{\theta}) + 2D$$

  "Minus twice the maximized log likelihood plus twice the number of parameters."
- We are invited to select the model with the lowest AIC score.
- AIC was derived as an approach to minimizing prediction error. Increasing the number of parameters leads to additional **overfitting** which decreases predictive skill of the fitted model.

A caution for using AIC

- Viewed as a hypothesis test, AIC may have weak statistical properties.
- It is a mistake to interpret AIC by making a claim that the favored model has been shown to provide a superior explanation of the data.
- However, viewed as a way to select a model with reasonable predictive skill from a range of possibilities, it is often useful.

Comparing AIC with likelihood ratio tests

Question. Suppose we are in a situation in which we wish to choose between two nested hypotheses, with dimensions $D^{\langle 0 \rangle} < D^{\langle 1 \rangle}$. Suppose the Wilks approximation is valid. Consider the strategy of selecting the model with the lowest AIC value, and view this model selection approach as a formal statistical test.

(A) Find an expression for the size of this AIC test (i.e., the probability of rejecting the null hypothesis, $H^{\langle 0 \rangle}$, when this null hypothesis is true).

(B) Evaluate this expression for $D^{\langle 1 \rangle} - D^{\langle 0 \rangle} = 1$.

Likelihood-based inference for ARMA models in Python

- The Great Lakes are an important resource for leisure, agriculture and industry in this region.
- A past concern has been whether human activities such as water diversion or channel dredging might be leading to a decline in lake levels.
- A current concern has been high levels leading to coastal erosion.
- Are lake levels affected by climate change?
- The physical mechanisms are not always obvious: for example, evaporation tends to be highest when the weather is cold but the lake is not ice-covered.
- We look at monthly time series data on the level of Lake Huron, which is essentially the same as Lake Michigan.

Reading in the data

The file `huron_level.csv` gives monthly water level, in meters, for Lakes Michigan and Huron from 1860 to 2025.

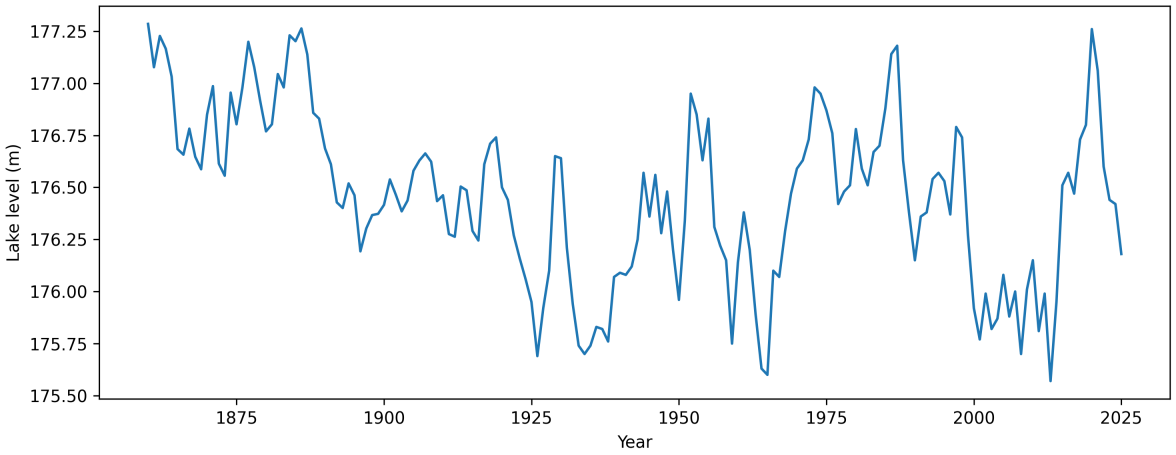
```
dat = pd.read_csv("huron_level.csv", comment='#')
dat.columns = dat.columns.str.strip()
print(dat.iloc[:2, :6])
```

```
   Year      Jan      Feb      Mar      Apr      May
0  1860  177.285  177.339  177.349  177.388  177.425
1  1861  177.077  177.105  177.224  177.254  177.382
```

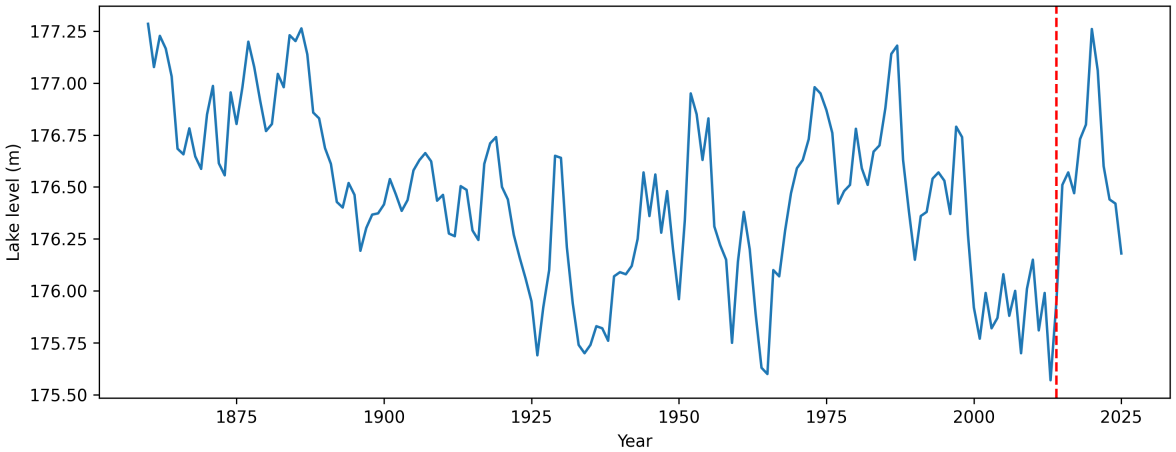
For now, we avoid monthly seasonal variation by considering an annual series of January depths. We will investigate seasonal variation later in the course, but sometimes it is best avoided.

```
huron_level = dat['Jan'].values
year = dat['Year'].values
plt.figure(figsize=(10, 4))
plt.plot(year, huron_level, '-')
plt.xlabel('Year')
plt.ylabel('Lake level (m)')
```

```
plt.tight_layout()
plt.show()
```



- Until the recent surge in water level, there was concern about a long-run decline in lake level due to dredging or water diversion or climate change.
- We put ourselves back in 2014 and temporarily ignore subsequent data.



Fitting an ARMA model

- Later, we will consider hypotheses of trend. For now, let's start by fitting a stationary ARMA(p, q) model under the null hypothesis that there is no trend. This hypothesis, which asserts that nothing has substantially changed in this system over the last 160 years, is not entirely unreasonable from looking at the data.
- We seek to fit a stationary Gaussian ARMA(p,q) model with parameter vector $\theta = (\phi_{1:p}, \theta_{1:q}, \mu, \sigma^2)$ given by

$$\phi(B)(Y_n - \mu) = \theta(B)\epsilon_n,$$

where

$$\begin{aligned} \mu &= \mathrm{E}[Y_n] \\ \phi(x) &= 1 - \phi_1 x - \cdots - \phi_p x^p, \\ \theta(x) &= 1 + \theta_1 x + \cdots + \theta_q x^q, \\ \epsilon_n &\sim \text{iid } N[0, \sigma^2]. \end{aligned}$$

We need to decide where to start in terms of values of p and q, for which it is helpful to tabulate AIC values.

```
def aic_table(data, P, Q):
    table = np.zeros((P+1, Q+1))
    for p in range(P+1):
        for q in range(Q+1):
            try:
                model = ARIMA(data, order=(p, 0, q))
                results = model.fit()
                table[p, q] = results.aic
            except:
                table[p, q] = np.nan
    df = pd.DataFrame(table,
        index=[f'AR{p}' for p in range(P+1)],
        columns=[f'MA{q}' for q in range(Q+1)])
    return df

huron_aic_table = aic_table(huron_level, 4, 5)
print(huron_aic_table.round(2))
```

```
        MA0     MA1     MA2     MA3     MA4     MA5
AR0  166.78   46.98    7.71  -13.70  -17.62  -24.89
AR1  -37.25  -36.62  -34.74  -33.13  -33.14  -31.18
AR2  -36.52  -37.41  -35.89  -32.42  -31.81  -29.85
AR3  -34.79  -35.88  -31.71  -31.21  -32.04  -29.61
AR4  -33.19  -32.62  -29.81  -32.61  -29.67  -29.83
```

Question. What do we learn by interpreting the results in the above table of AIC values?

Question. In what ways might we have to be careful not to over-interpret the results of this table?

We proceed to inspect the ARMA(2,1) model with lowest AIC.

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  155
Model:                 ARIMA(2, 0, 1)   Log Likelihood                23.707
Date:                Wed, 28 Jan 2026   AIC                            -37.414
Time:                        14:14:23   BIC                            -22.197
Sample:                             0   HQIC                           -31.233
                                - 155
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const        176.4595      0.114   1554.123      0.000     176.237     176.682
ar.L1         -0.0558      0.060     -0.926      0.355      -0.174       0.062
ar.L2          0.7935      0.061     13.080      0.000       0.675       0.912
ma.L1          0.9995      0.769      1.301      0.193      -0.507       2.506
sigma2         0.0422      0.032      1.313      0.189      -0.021       0.105
===================================================================================
Ljung-Box (L1) (Q):                   0.06   Jarque-Bera (JB):                 0.63
Prob(Q):                              0.81   Prob(JB):                         0.73
Heteroskedasticity (H):               1.95   Skew:                             0.15
Prob(H) (two-sided):                  0.02   Kurtosis:                         2.90
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

Examining the AR and MA roots

- We can examine the roots of the AR polynomial:

```
ar_pars = huron_arma21.arparams
ar_poly = np.polynomial.Polynomial(np.r_[1,-ar_pars])
print(f"AR roots: {ar_poly.roots()}")
```

```
AR roots: [-1.08801273  1.1583089 ]
```

- The roots are just outside the unit circle, suggesting we have a stationary causal fitted ARMA.
- However, let's check the MA root:

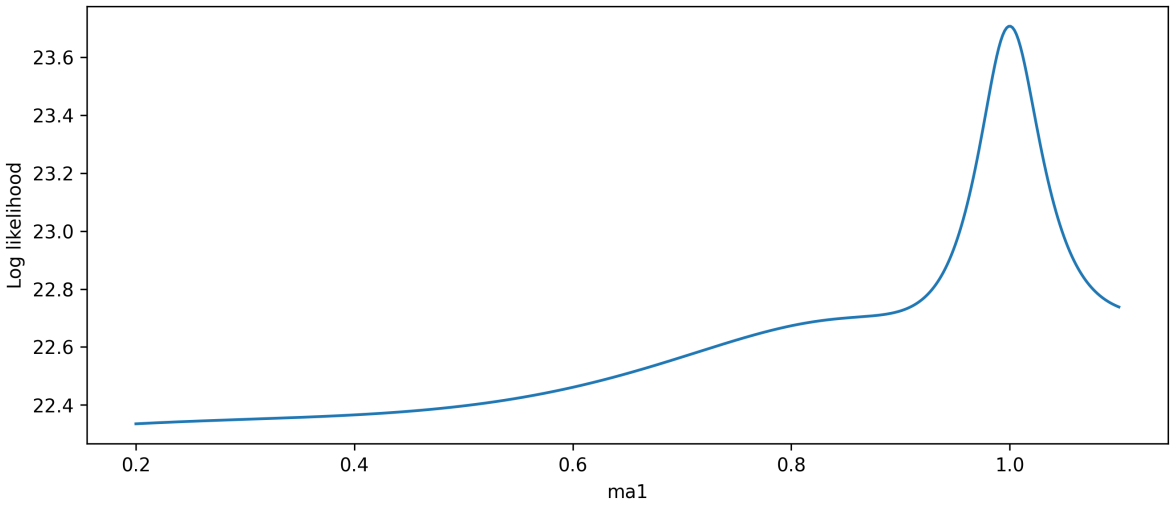
```
ma_pars = huron_arma21.maparams
ma_poly = np.polynomial.Polynomial(np.r_[1,ma_pars])
print(f"MA root: {ma_poly.roots()}")
```

```
MA root: [-1.00046719]
```

The fitted model is close to the threshold of non-invertibility.

- Do we have a non-invertibility problem? We investigate this using profile and bootstrap standard error to check the Fisher Information approach.
- First, we compute the profile likelihood confidence interval. To do this, we maximize the ARMA likelihood while fixing the MA1 coefficient at a range of values.

```
import shelve
K = 500
ma1_vals = np.linspace(0.2, 1.1, K)
with shelve.open('cache/profile') as cache:
    if "profile_loglik" not in cache:
        profile_loglik = np.zeros(K)
        for k in range(K):
            model = ARIMA(huron_level, order=(2,0,1))
            with model.fix_params({'ma.L1': ma1_vals[k]}):
                results = model.fit()
                profile_loglik[k] = results.llf
        cache["profile_loglik"] = profile_loglik
    else:
        profile_loglik = cache["profile_loglik"]
```



Question. Interpret the profile likelihood plot for θ_1.

Question. What do you conclude about the Fisher information confidence interval?

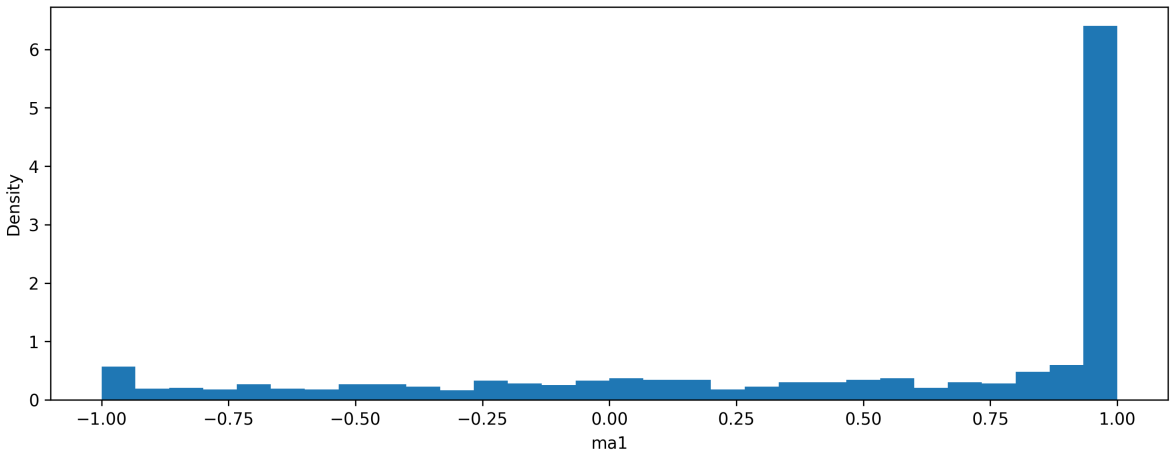
Question. In what situations is the Fisher information confidence interval reliable?

Question. Is this profile likelihood plot, and its statistical interpretation, reliable? How could you support your opinion on this?

A simulation study

```
np.random.seed(578922)
J = 1000
params = huron_arma21.params
ar = params[1:3]; ar_poly = np.r_[1, -ar]
ma = params[3:4]; ma_poly = np.r_[1, ma]
intercept = params[0]
sigma = np.sqrt(huron_arma21.scale)
```

```
with shelve.open('cache/simulate') as cache:
    if "theta_list" not in cache:
        theta_list = []
        for j in range(J):
            arma_process = ArmaProcess(ar_poly, ma_poly)
            Y_j = arma_process.generate_sample(
                nsample=len(huron_level),
                scale=sigma) + intercept
            model_j = ARIMA(Y_j, order=(2,0,1)).fit()
            theta_list.append(model_j.params[3])
        cache["theta_list"] = theta_list
    else:
        theta_list = cache["theta_list"]
```



- This seems consistent with the profile likelihood plot.
- Note that statsmodels ARIMA transforms the model to invertibility. Thus, the estimated value of θ_1 can only fall in the interval $[-1, 1]$.

Using parallel computing for simulation studies

- We do a simulation study for which we fit ARMA(2,1) when the true model is AR(1).
- When doing simulation studies, **parallel computing** is helpful. All modern computers have multiple cores.
- In Python, we can use the `joblib` library for parallel computing with a simple interface.

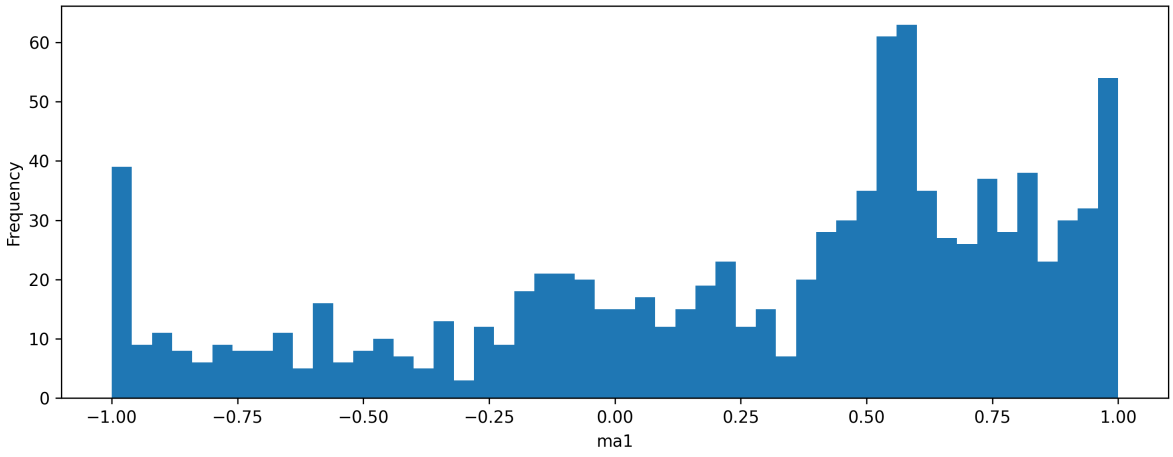
- The `joblib` library provides `Parallel` and `delayed` for easy parallelization of loops. For J parallel replications of `my_function`,

```
results = Parallel(n_jobs=-1)(delayed(my_function)(j)
    for j in range(J))
```

```
J = 1000
huron_ar1 = ARIMA(huron_level, order=(1,0,0)).fit()
params = huron_ar1.params
ar = params[1]; ar_poly = np.r_[1, -ar]
intercept = params[0]; sigma = params[2]
ma_poly = np.array([1])
arma_process = ArmaProcess(ar_poly, ma_poly)
def fit_arma21(j):
    Y_j = arma_process.generate_sample(
        nsample=len(huron_level),
        scale=sigma) + intercept
    model_j = ARIMA(Y_j, order=(2,0,1)).fit()
    return model_j.params[3]  # MA1 coefficient

huron_sim = Parallel(n_jobs=-1)(delayed(fit_arma21)(j)
    for j in range(J))
```

- Now, we can look at the resulting estimates of the MA1 component:



- When the true model is AR(1) and we fit ARMA(2,1), it seems that we often obtain a model with estimated MA1 coefficient on the boundary of invertibility.
- Thus, we cannot reject an AR(1) hypothesis for the Huron data, even though the Fisher information based analysis appears to give strong evidence that the data should be modeled with a nonzero MA1 coefficient.
- It may be sensible to avoid fitted models too close to the boundary of invertibility. This is a reason not to blindly accept whatever model AIC might suggest.

Question. What else could we look for to help diagnose, and understand, this kind of model fitting problem? Hint: pay some more attention to the roots of the fitted ARMA(2,1) model.

Assessing the numerical correctness of evaluation and maximization of the likelihood function

- We can probably suppose that the ARIMA fitting function has negligible numerical error in evaluating the likelihood.
- Likelihood evaluation is a linear algebra computation which should be numerically stable away from singularities.
- Possibly, numerical problems could arise for models very close to reducibility (canceling AR and MA roots).
- Numerical optimization is more problematic.
- The likelihood surface can be multimodal and have nonlinear ridges, when AR and MA roots almost cancel.
- No optimization procedure is reliable for maximizing awkward, non-convex functions.
- Evidence for imperfect maximization (assuming negligible likelihood evaluation error) can be found in the AIC table.

	MA0	MA1	MA2	MA3	MA4	MA5
AR0	166.8	47.0	7.7	-13.7	-17.6	-24.9
AR1	-37.2	-36.6	-34.7	-33.1	-33.1	-31.2
AR2	-36.5	-37.4	-35.9	-32.4	-31.8	-29.9
AR3	-34.8	-35.9	-31.7	-31.2	-32.0	-29.6
AR4	-33.2	-32.6	-29.8	-32.6	-29.7	-29.8

Question. How is this table inconsistent with perfect maximization?

- Hint: recall that, for nested hypotheses $H^{\langle 0 \rangle} \subset H^{\langle 1 \rangle}$, the likelihood maximized over $H^{\langle 1 \rangle}$ cannot be less than the likelihood maximized over $H^{\langle 0 \rangle}$.
- Recall also the definition of AIC:

AIC = -2 $\times$ maximized log likelihood + 2 $\times$ number of parameters

Further reading

- Section 3.5 of Shumway and Stoffer [2] and Section 4.2 of Huang and Petukhina [1] give complementary discussions of parameter estimation for ARMA models.
- Section 3.7 of Shumway and Stoffer [2] takes a different perspective on selecting ARMA models, putting less emphasis on likelihood. Section 4.3 of Huang and Petukhina [1] is closer to the approach in these notes. Both perspectives can be valuable.

Acknowledgments

- Compiled on January 28, 2026 using Python.
- Licensed under the Creative Commons Attribution-NonCommercial license. Please share and remix non-commercially, mentioning its origin.
- We acknowledge previous versions of this course.

References

[1] Changquan Huang and Alla Petukhina. *Applied Time Series Analysis and Forecasting with Python*. Springer, 2022.

[2] Robert H Shumway and David S Stoffer. *Time Series Analysis and its Applications: With R Examples*. 4th. Springer, 2017.