SARIMA Model Diagnostics and Selection

Jared Fisher

Lecture 09a

Announcements

Announcements

- In-class vote: should Homework 5 be due tomorrow/Wednesday April 7 by 11:59pm,
- Or Friday April 9 by 11:59pm? (there's no homework next week to intrude upon)
- Midterm 2 is next week on April 15, and will contain two separate parts:
  - Timed multiple choice section on Gradescope (like Midterm 1, but timed at 20 minutes)
  - Written portion that you'll have the full 24 hours to complete, though it's intended to take 30 minutes or so. You'll upload this to Gradescope like a homework assignment. You may hand-write or type (LaTeX, Markdown, etc.)

Recap

So many names??

Everything fits under SARIMA:

$$
\begin{aligned}
SARIMA: &\quad \Phi(B^S)\phi(B)\nabla_S^D\nabla^d V_t = \Theta(B^S)\theta(B)W_t \\
MSARMA: &\quad \Phi(B^S)\phi(B)X_t = \Theta(B^S)\theta(B)W_t \\
SARMA: &\quad \Phi(B^S)X_t = \Theta(B^S)W_t \\
ARIMA: &\quad \phi(B)\nabla^d V_t = \theta(B)W_t \\
ARMA: &\quad \phi(B)X_t = \theta(B)W_t
\end{aligned}
$$

Note: X_t is stationary, and V_t is stationary after differencing.

Definition: ARIMA

A process V_t is said to be ARIMA(p, d, q) if $X_t = (I - B)^d V_t$ is ARMA(p, q) with mean μ.

In other words:

$$\phi(B)(X_t - \mu) = \theta(B)W_t,$$

where $\{W_t\}$ is white noise.

Definition: Seasonal ARMA

The doubly infinite sequence $\{X_t\}$ is said to be a seasonal ARMA(P, Q) process with period S if it is stationary and if it satisfies the difference equation $\Phi(B^S)X_t = \Theta(B^S)W_t$ where $\{W_t\}$ is white noise and

$$\Phi(B^S) = 1 - \Phi_1 B^S - \Phi_2 B^{2S} - \cdots - \Phi_P B^{PS}$$

and

$$\Theta(B^S) = 1 + \Theta_1 B^S + \Theta_2 B^{2S} + \cdots + \Theta_Q B^{QS}.$$

Definition: MSARMA

The Multiplicative Seasonal Autoregressive Moving Average Model ARMA$(p, q) \times (P, Q)_S$ is defined as the stationary solution to the difference equation:

$$\Phi(B^S)\phi(B)X_t = \Theta(B^S)\theta(B)W_t,$$

for some white noise process $\{W_t\}$.

Definition: SARIMA

A process V_t is said to be ARIMA$(p,\, d,\, q) \times (P,\, D,\, Q)_S$, if after differencing d times and seasonal differencing D times, it follows a multiplicative seasonal ARMA model, that is, if it satisfies the difference equation:

$$\Phi(B^S)\phi(B)\nabla_S^D\nabla^d V_t = \Theta(B^S)\theta(B)W_t.$$

Problem

The Next Problem

- Thus far, we have learned about various models for time series data, how to forecast, and more.
- We've discussed some diagnosis strategies for choosing p/q/etc. from ACF/PACF plots. But there's two holes in this technique.
- First, if p>0 and q>0, it's often hard to use the plots alone to know what ARMA model is represented.
- Second, real-world data is messy and often doesn't fit nicely into a certain p/q/P/Q/S.
- So, how should we decide which model to use?
- Today we will discuss a few common techniques to answer this question.

Example

Let's look at the price of salmon and form two candidate models, using the *sarima()* function

(to the code!)

Questions we could ask

- Does the model fit well (in-sample)?
- Does the model fit better than other models (in-sample)?
- Does the model forecast better than other models (out-of-sample)?

Does the model fit well?

Definition: Ljung-Box-Pierce test

- Fix a maximum lag k (typically $k = 20$).
- Reject the hypothesis that data $x_1, \ldots, x_n$ was generated from a causal and invertible ARMA(p,q) model if

$$\tilde{Q}(x_1, \ldots, x_n) > q_{1-\alpha},$$

- where $q_{1-\alpha}$ denotes the $(1-\alpha)$-quantile of the χ^2 distribution with $k - p - q$ degrees of freedom.

What is this Q?

- Assume that the data $X_1, \ldots, X_n$ is generated from an invertible ARMA(p,q) model with parameters ϕ, θ.
- By invertibility:
$$X_t = -\sum_{j \geq 0} \pi_j X_{t-j} + W_t$$
- Hence, the best linear prediction of X_t based on$X_{t-1}, X_{t-2}, \ldots$ is given by
$$\hat{X}_t(\phi, \theta) = -\sum_{j \geq 0} \pi_j X_{t-j}.$$
- The residuals $R_t = \hat{X}_t(\phi, \theta) - X_t = W_t$ coincide with the white noise process $\{W_t\}$.

What is this Q?

- Recall that sample acf of the residuals R_t $(r_1, \ldots, r_k)$ for some maximal lag k, are approximately i.i.d. N(0,1/n)
- Thus, we create Q which follows a chi-square distribution with k degrees of freedom:

$$Q = n \sum_{i=1}^{k} r_i^2 \sim \chi_k^2$$

- When the true parameters ϕ, θ are replaced by appropriate estimates $\hat{\phi}, \hat{\theta}$, the respective estimated residuals

$$\hat{R}_t = \hat{X}_t(\hat{\phi}, \hat{\theta}) - X_t$$

should still be approximately white noise.

What is this Q?

- The *Box-Pierce* test statistic is

$$\hat{Q} = n \sum_{i=1}^{k} \hat{r}_i^2$$

- Under an ARMA(p,q) model, one can show that for n large enough $\hat{Q}$ is approximately chi-square distributed with $k - p - q$ degrees of freedom

$$\hat{Q} \to \chi^2_{k-p-q} \quad \text{for } n \to \infty.$$

- In practice, one often considers a slightly modified version of the statistic $\hat{Q}$, namely

$$\tilde{Q} = n(n+2) \sum_{i=1}^{k} \frac{\hat{r}_i^2}{n-i},$$

which is denoted as the *Ljung-Box-Pierce* test statistic.

Definition: Ljung-Box-Pierce test

- Fix a maximum lag k (typically $k = 20$).
- Reject the hypothesis that data $x_1, \ldots, x_n$ was generated from a causal and invertible ARMA(p,q) model if

$$\tilde{Q}(x_1, \ldots, x_n) > q_{1-\alpha},$$

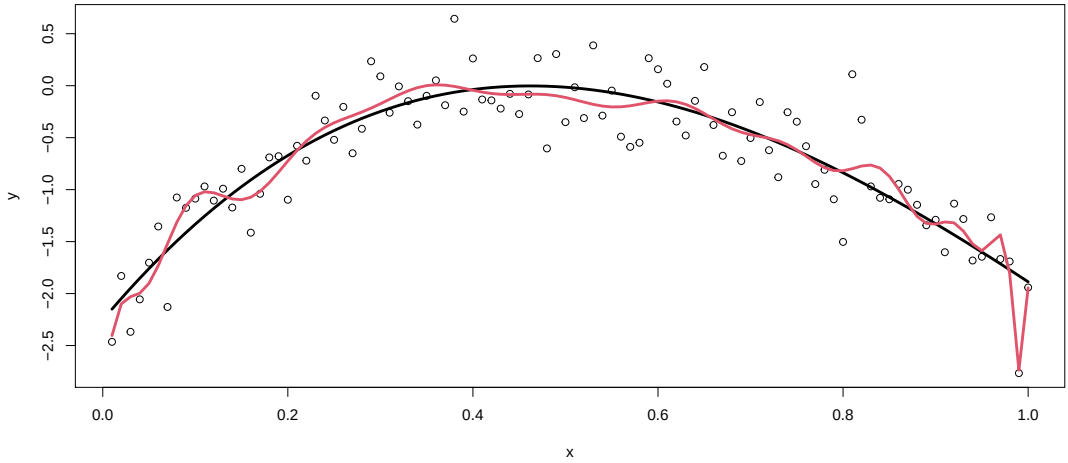
- where $q_{1-\alpha}$ denotes the $(1-\alpha)$-quantile of the χ^2 distribution with $k - p - q$ degrees of freedom.

Which model fits best in-sample?

Overfit

- The Ljung-Box-Pierce test provides a strategy to evaluate, for given p, q, whether or not an ARMA(p,q) model is appropriate for data $x_1, \ldots, x_n$.
- But how should we choose the parameters p and q in the first place?
- Clearly every ARMA(p,q) model can be arbitrarily-well approximated by an ARMA(p', q') model with $p\prime > p$ and $q' > q$.
- Like with a polynomial in linear regression, the larger chosen degree of the polynomial, the better the fit.

Overfit



Overfit

- In the extreme case, when we have a 100 observations and fit a polynomial of degree 99 the fit will be perfect.
- However, such a model is likely overfitting the data and might be useless to predict future values.
- Model selection: we want the number of model parameters to be large enough, so that it can fit the data well. At the same time the number of model parameters should not be too large, which would result in overfitting: fitting the data and not the true underlying process.
- A solution: measure in-sample fit and penalize for model size/complexity.

But First: Likelihood

- The likelihood is a function of the parameters, the same function as the density of the data

$$L(\textit{parameters}) = f(\textit{data}|\textit{parameters})$$

- For our models this typically looks like

$$L(\mu, \phi, \theta, \sigma^2) = f(x_1, ..., x_n|\mu, \phi, \theta, \sigma^2)$$

Akaike's information criterion (AIC)

- Used in various areas, not just in time series analysis.
- As a model selection criterion, choose the model that minimizes

$$AIC = -2\log(\text{likelihood}) + 2k$$

- k denotes the number of parameters in the model.
- We want to maximize the likelihood, a measure of the in-sample fit of the model (i.e., the model performance on the given data set).
- The term $2k$ serves as a penalty function which penalizes models for each parameter used

AIC Notes

- Example: for ARMA(p, q) model with a non-zero mean μ and noise variance σ^2, we have $k = p + q + 2$.
- When you fit a model in R with the *arima()* function it will also output *aic*.
- So if you want to choose between a collection of different models, that is values for p and q you can chose the one where the AIC value is smallest.
- Using the likelihood makes sense, but why the $2k$ for the penalty?

Bayesian Information Criterion (BIC)

$$BIC = -2\log(\text{likelihood}) + k \log n.$$

- Note that the penalty above is larger than that of AIC.
- Consequently, BIC selects more parsimonious (simple) models compared to AIC.

Bias Corrected AIC (AICc),

$$AICc = AIC + \frac{2k(k+1)}{n-k-1}$$

- Corrected for small sample sizes, where AIC tends to select too many parameters
- Similar to BIC, AICc selects more parsimonious models compared to AIC, but the difference between AIC and AICc vanishes as n increases.

Back to the code

To see the diagnostic plots and IC's.

How do we compare models out-of-sample?

Cross Validation

- Very popular technique for model selection and choice of tuning parameters, in general.
- General idea: we want to know how the model will perform out-of-sample, so let's reserve some of our data to check this out!
- Basic Idea:
  - Divide dataset into "training" and "testing" subsets
  - Fit all candidate models with the training data
  - Evaluate the performance of each model on the testing data
  - Repeat as needed
  - Select the model that performed the best

Cross Validation Example Steps

- Disclaimer: there are many ways to do cross validation for time series. This is one.
- Suppose we have monthly data for m years $x_1, \ldots, x_n$ where $n = 12m$ and the objective is to predict the data for the next year.
- Suppose we have ℓ competing models $M_1, \ldots, M_\ell$ for the dataset. We can use cross-validation in order to pick one of these models in the following way:

Cross Validation Example Steps

1. Fix a model M_i. Fix $k < m$.
2. Fit the model M_i to the data from the first k years.
3. Using the fitted model, predict the data for the $(k+1)$st year.
4. Calculate the sum of squares of errors of prediction for the $(k+1)$st year.
5. Repeat these steps for $k = k_0, \ldots, m-1$ where k_0 is an arbitrary value of your choice.
6. Average the sum of squares of errors of prediction over $k = k_0, \ldots, m-1$. Denote this value by CV_i and call it the Cross Validation score of model M_i.
7. Calculate CV_i for each $i = 1, \ldots, \ell$ and choose the model with the smallest Cross-Validation score.

Cross Validation Example Steps - Specific

For monthly stock data from 2001 to present, with your model and my model. The psuedo code for this:

```
for(M in Models){
  for(k in 2011:2018){
    fitted.model = model(M, data = 2001 to year (k-1))
    predictions = predict year k using fitted.model
    accuracy_k = sum(([year k data] - predictions)^2)
  }
  CV_M = sum(accuracy_k)/8
}
```

Then, choose the model with smallest CV_M