STAT 436 / 536 - Lecture 15

State Space Models

- Recall the innovations form of the state space model that we discussed last time has the following form:

$$\begin{aligned} y_t &= \widetilde{w}^T \widetilde{x}_t + \epsilon_t \\ \widetilde{x}_t &= F \widetilde{x}_{t-1} + g \epsilon_t, \end{aligned}$$

  where only a single error term, ϵ_t, exists.

- Now, contrast this with a state space model below

ARMA Models

- The ARMA(p,q) model can be written compactly as

- Consider the following specification: $\widetilde{w}^T = (1\ 0)$, $\widetilde{x}_t =$, $v_t = 0$, $F = \begin{bmatrix} \alpha_1 & 1 \\ \alpha_2 & 0 \end{bmatrix}$, $R = (1\ 0)^T$

$$\begin{aligned} y_t &= (1\ 0) \begin{pmatrix} x_{1,t} \\ x_{2,t} \end{pmatrix} \\ \begin{pmatrix} x_{1,t} \\ x_{2,t} \end{pmatrix} &= \begin{bmatrix} \alpha_1 & 1 \\ \alpha_2 & 0 \end{bmatrix} \begin{pmatrix} x_{1,t-1} \\ x_{2,t-1} \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} w_t \end{aligned}$$

- The observation equation is:

- For the state equation, we have:

- The final step is to simplify $x_{1,t}$ by substituting $x_{2,t-1}$. Thus,

- Similarly, an ARMA(2,1) model can be specified as: $\widetilde{w}^T = (1\ 0)$, $v_t = 0$, $F = \begin{bmatrix} \alpha_1 & 1 \\ \alpha_2 & 0 \end{bmatrix}$, $R = (1\ \beta)^T$

$$
\begin{aligned}
y_t &= (1\ 0) \begin{pmatrix} x_{1,t} \\ x_{2,t} \end{pmatrix} \\
\begin{pmatrix} x_{1,t} \\ x_{2,t} \end{pmatrix} &= \begin{bmatrix} \alpha_1 & 1 \\ \alpha_2 & 0 \end{bmatrix} \begin{pmatrix} x_{1,t-1} \\ x_{2,t-1} \end{pmatrix} + \begin{pmatrix} 1 \\ \beta \end{pmatrix} w_t
\end{aligned}
$$

  Watch for a homework question about this!

- Recall (536 students) there were two assumptions for state space models.

1. The state vector and

2. The observed values are

- A major benefit of the state-space model framework is that we can easily integrate the stochastic modeling approach for serial correlation (ARMA) directly with additional covariates.

- For instance consider the following model:

$$
\begin{aligned}
y_t &= (1\ 0\ \ ) \begin{pmatrix} x_{1,t} \\ x_{2,t} \end{pmatrix} \\
\begin{pmatrix} x_{1,t} \\ x_{2,t} \end{pmatrix} &= \begin{bmatrix} \alpha_1 & 1 & 0 \\ \alpha_2 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_{1,t-1} \\ x_{2,t-1} \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} w_t
\end{aligned}
$$

- What is this model? Specifically, what are z_t and β? How could we add dynamics to β?

- The `dlm` package contains functions for fitting general models of this type. Standard MCMC algorithms can also be constructed for these models, but we will also have a 536-specific lab that introduces Sequential Monte Carlo (SMC) methods that can be used for this type of model in general.

- The `auto.arima()` function in the `forecast` package does have the option to include covariates.

Exercise

- Use the `auto.arima()` function to model the pastry count. Assume that we are looking at this through the lens of explanatory inference, in other words you need to explain to a baker what you see in your model.:

1. Change the frequency of the ts object from 1 to 7, what do you see? why are there differences?

2. Include the number of coffee drinks purchased on the same day as a covariate, how does the model change?

3. How do the model using the ARMA structure compare to a standard linear regression model?

4. As in Lab 7, assume our goal was to predict the number pastries consumed tomorrow, how would your model change?

5. What if the baker was interested in both the number of pastries *and* the number of cups of coffee to prepare for tomorrow. How would this be modeled?

```r
library(readr)
library(dplyr)
library(forecast)
bakery_sales <- read_csv('http://math.montana.edu/ahoegh/teaching/timeseries/data/BreadBasket.csv')

pastry_count <- bakery_sales %>%
  filter(Item %in% c('Pastry','Scandinavian','Medialuna','Muffin','Scone')) %>%
  group_by(Date) %>% tally() %>% rename(num_pastry = n)

drink_count <- bakery_sales %>% filter(Item %in% c('Coffee','Tea')) %>%
  group_by(Date) %>% tally() %>% rename(num_drink = n)

ggtsdisplay(ts(pastry_count$num_pastry))
```

ts(pastry_count$num_pastry)