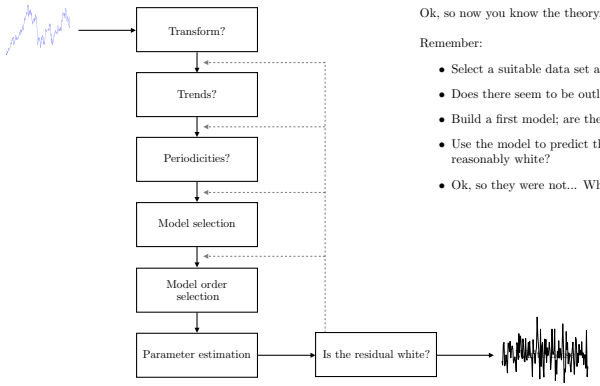
Wrapping it all up

Andreas Jakobsson

Reflections



Ok, so now you know the theory, lets sort out the project...

Remember:

- Select a suitable data set as your modelling data
- Does there seem to be outliers? Need for transformations?
- Build a first model; are the model residuals reasonably white?
- Use the model to predict the validation data; are the prediction residuals reasonably white?
- Ok, so they were not... What to do next?

Reflections

Some suggestions and thoughts:

- Are all coefficients significant? Can you reduce the model? Should you add some seemingly too weak component?
- Should you replace that ∇ with a $(1 - az^{-1})$ and estimate the a?
- Is the periodicity not perfectly aligned with the samples? Maybe worth testing replacing the ∇_s with

$$(1 - a_{s-1}z^{-s+1} - a_s z^{-s} - a_{s+1}z^{-s-1})$$

  This will allow the periodicity to be close to s, but does not require it to be precisely s.

Reflections

Some suggestions and thoughts:

- Are all coefficients significant? Can you reduce the model? Should you add some seemingly too weak component?
- Should you replace that ∇ with a $(1 - az^{-1})$ and estimate the a?
- Is the periodicity not perfectly aligned with the samples? Maybe worth testing replacing the ∇_s with

$$(1 - a_{s-1}z^{-s+1} - a_s z^{-s} - a_{s+1}z^{-s-1})$$

  This will allow the periodicity to be close to s, but does not require it to be precisely s.
- Do not oversimplify your model; do not restrict it to be AR to allow you to use RLS. This is unlikely to work well... Do not expect complex phenomena to be well modelled with almost no parameter...
- Are your predictions ok, but lower than you expect? Did you remember to add the predicted input?
- Does it seem difficult to get rid of all dependencies? Is the model really stationary? Test using your recursive predictor.
- Always, always, always... Compare to a naive predictor. If your model is not better than that, it is not worth much!

Reflections

More suggestions and thoughts:

- Does it make sense to transform the data? Look at the Box-Cox plot. How much higher is really that peak?
- Simulate a process, with lots of data, and test your code. Test your predictions using the model you used to simulate the data. Check if your Kalman filter converge to the true parameters if initialised wrong. Everything should work really well; if it does not, something is wrong. Check your code!
- Do you suffer from outliers? Why not compute the TACF and compare it to the ACF - if you are, these estimates will likely differ.
- Remember that many of the results only holds asymptotically - and typically relies on several assumptions that might not be met. Be careful with your trust!

Reflections

More suggestions and thoughts:

- Does it make sense to transform the data? Look at the Box-Cox plot. How much higher is really that peak?
- Simulate a process, with lots of data, and test your code. Test your predictions using the model you used to simulate the data. Check if your Kalman filter converge to the true parameters if initialised wrong. Everything should work really well; if it does not, something is wrong. Check your code!
- Do you suffer from outliers? Why not compute the TACF and compare it to the ACF - if you are, these estimates will likely differ.
- Remember that many of the results only holds asymptotically - and typically relies on several assumptions that might not be met. Be careful with your trust!
- It is often wise to plot your ACF without $\hat{\rho}_y(0)$ to better see the details.
- Always plot predictions in the correct domain - do not show the transformed differentiated predictions; show them in the domain you work in!
- If you add an input, do not retain your old model - use the input maximally before modelling the residual!
- Look at the predictions and residuals; do they make sense? Are they in the correct domain?
- Be careful to use Matlab commands that you do not fully understand.
- Multiple inputs? Use the one with the strongest correlation first, then use the next on the resulting residual.
- Is it good enough? Then, you are done!