Lecture 1

INTRO

- The last part of the course is **Chapter 6: State Space Modelling**
- Often, with this course, I take a subset of what remains in the lecture notes
- There is updating of material and, often, new case studies and notation
- I would advise, then, for the last weeks to focus attention on what's in the slides rather than the lecture notes

- This recording is a high level overview of what we are going to be looking at
- We have already studied linear methods pretty completely from a mathematical, statistical, computational and practical viewpoints
- There are many non-linear approaches
- We will be keeping ideas of bias-variance trade-off in mind all the time
- We focus on a number of examples of *state space* methods
- We will also have some readings from the end of Chapter 5 which allow comparison between linear and non-linear models

- We will just be introducing each of a number of topics:
- State space models - this is common to all
- Bayesian methods
- Insurance case study
- Kalman Filter: forecast and control
- State space models and non-constant volatility models

State Space Models

Definition.6.1.1 (State space model)

- If $\{Y_t\}$ is an (observed) time series with (unobserved) state process $\{\theta_t\}$.
- In a state space model the dependence between these is can defined by the graphical model shown in Fig. 1

Figure: State space structure

Bayesian Methods

- We shall assume that we have a model for the data $f(x|\theta)$ where we want to learn about the (vector) of parameters θ.
- In Bayesian statistics everything is considered a random variable
- All statements about uncertainty are made using the language and calculus of probability theory.
- Move from *prior distribution* to *posterior distribution* as we update with data
- Illustrate with *Insurance case study* using a state space approach

Kalman Filter

- We look at the Kalman filter
- This is a recursive algorithm which can run in 'real time' – that is at the same rate that the data is collected
- In a general state space model we have an (unobserved) state variable at time s and a set of observations $y_1, \ldots, y_t$
- We subdivide problems by:
  (i) *filtering* is the case where (s=t),
  (ii) *state prediction* is the case when $s > t$ and
  (iii) *smoothing* is the case $s < t$.
- If we are filtering we might was to do the state prediction in a time which is faster than the rate that new data arrives

- The 'real time' behaviour is critical for *control* applications
- One of the reasons we forecast is to be able to make good control decisions
- With real time control problems you typically need to be able to compute predictions faster than you are gathering data.
- We will look at theory, links to Bayesian methods and computation

Non-constant volatility models

Example.7.0.1 (The Bollerslev-Ghysel benchmark dataset)

- The data in Fig. 7.1 from Bollerslev and Ghysels (1996) is the daily percentage nominal returns for the Deutschemark-Pound exchange rate.
- We see the non-constant volatility and volatility clustering that is common in such financial data.
- We also can see from the marginal plots that, where the dynamics is excluded, we get non-normal behaviour.
- The data is 'heavier-tail' than a constant variance Gaussian process.

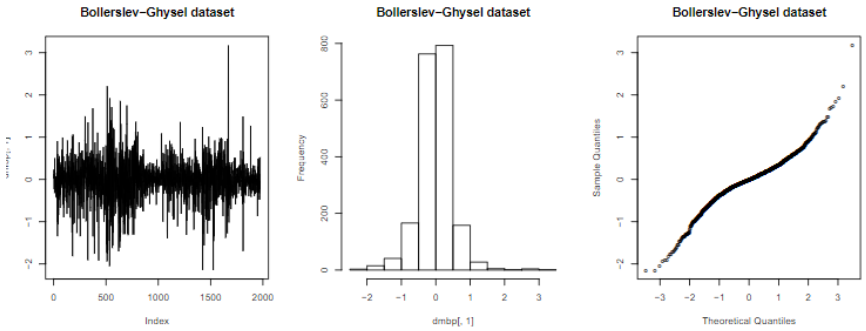


Figure: 7.1 Bollerslev-Ghysel benchmark dataset

Lecture 2

Intro

- This set of slides introduces the state space structure
- These give very useful, non-linear, models.
- Inference is more complex
- Bayesian methods can be useful and link to the Kalman filter and control methods

State Space models

Definition.6.1.1 (State space model)

- If $\{Y_t\}$ is an (observed) time series with (unobserved) state process $\{\theta_t\}$.
- In a state space model the dependence between these is can defined by the graphical model shown in Fig. 1

Figure: State space structure

- In a general state space model we have an (unobserved) state variable at time s and a set of observations $y_1, \ldots, y_t$
- We subdivide problems by:
  (i) *filtering* is the case where (s=t),
  (ii) *state prediction* is the case when $s > t$ and
  (iii) *smoothing* is the case $s < t$.
- If we are filtering we might was to do the state prediction in a time which is faster than the rate that new data arrives
- One reason we want to forecast is to be able to control uncertain systems

Definition.6.1.1 (State space model)

- The relationship in the graphical model defines the conditional independence relations:
  (i) θ_t is a Markov chain
  (ii) conditionally on θ_t for $t = 0, 1, \ldots, t$ then Y_t are independent
- So that the joint distribution of $(Y_1, \ldots, Y_t) \,|\, (\theta_1, \ldots, \theta_t)$ is

$$Lik(\theta) := \prod_{i=1}^{t} f(y_t|\theta_t).$$

- State space models are also called *hidden Markov models*.

Bayesian Methods

- We shall assume that we have a model for the data $f(x|\theta)$ where we want to learn about the (vector) of parameters θ.
- Note that when θ is high-dimensional frequentist methods have problems, but MCMC Bayesian methods can work – see case study
- In Bayesian statistics everything is considered a random variable
- All statements about uncertainty are made using the language and calculus of probability theory.
- Move from *prior distribution* to *posterior distribution* as we update with data
- Illustrate with *Insurance case study*

- $Pr(\theta)$ which we shall call the *prior distribution*
- What we know about θ *before* we see the data
- To see what we know about θ *after* we see the data we apply Bayes theorem to give

$$Pr(\theta|\text{ data }) = \frac{Pr(\text{ data }|\theta)Pr(\theta)}{Pr(\text{data})}.$$

- In other words

$$\text{Posterior}(\theta) \propto Lik(\theta) \times \text{prior}(\theta)$$

Example. Consider a simple normal example, where we have a model $X_i \overset{i.i.d.}{\sim} N(\theta, 1)$ with a sample size of n, so that the likelihood function based on the observed data $x_1, \ldots, x_n$ is

$$\begin{aligned} Lik(\theta) &= Pr(data|\theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{(x_i-\theta)^2}{2}\right] \\ &= \left(\frac{1}{\sqrt{2\pi}}\right)^n \exp\left[-\sum_{i=1}^{n} \frac{(x_i-\theta)^2}{2}\right] \\ &\propto \exp\left[\theta(\sum_{i=1}^{n} x_i) - \frac{n}{2}\theta^2\right] \end{aligned}$$

This is then the key tool that we need to merge with the prior in order to get the posterior.

Example. Suppose that the prior distribution for θ was in the Normal family i.e.

$$\theta \sim N(\theta; \mu_{\text{prior}}, \sigma^2_{\text{prior}}) \propto \exp\left[-\frac{(\theta-\mu_{\text{prior}})^2}{2\sigma^2_{\text{prior}}}\right]$$

where μ_{prior} and σ^2_{prior} are considered known constants. Then the posterior can be written as

$$\begin{aligned} Pr(\theta|x_1, \ldots, x_n) &\propto Lik(\theta) \times \text{prior}(\theta) \\ &= \exp\left[-\frac{1}{2}\frac{\theta^2 - 2\frac{\left(\mu_{\text{prior}}+\sigma^2_{\text{prior}}(\sum_{i=1}^{n} x_i)\right)}{(1+n\sigma^2_{\text{prior}})}\theta}{\sigma^2_{\text{prior}}/\left(1+n\sigma^2_{\text{prior}}\right)}\right] \end{aligned}$$

Example. This gives

$$Pr(\theta|x_1,\ldots,x_n) \sim N\left(\frac{\left(\mu_{\text{prior}} + \sigma^2_{\text{prior}}(\sum_{i=1}^n x_i)\right)}{\left(1+n\sigma^2_{\text{prior}}\right)}, \frac{\sigma^2_{\text{prior}}}{\left(1+n\sigma^2_{\text{prior}}\right)}\right)$$

- The basic rule is Posterior(θ) $\propto$ Likelihood(θ) $\times$ prior(θ). So we need to integrate to find the normalising constant
- In this example we only got a neat solution by a *'careful'* choice of the prior
- In high-dimensional problems this is not possible
- We use instead a numerical method, MCMC, which does not involve integration.

Summary

- This short set of slides has give basic introduction to state space models and the basics of Bayesian statistics
- We are not aiming to get all the details of the Bayesian method
- We need enough to understand some approaches to dealing with complex real world examples
- Also to motivate what we see later with the Kalman filter

Lecture 3

Introduction

- This set of slides introduces the idea of Markov Chain Monte Carlo
- Your emphasis here should be the interpretation of the output, not the algorithm itself
- The aim it to be able to understand a complex case study that is in the next talk
- We see later links to the Kalman filter and control methods

Representing Probability Distributions

- The usual way that we think about defining a probability distribution (which defines the properties of a random variable X) is through a mathematical formula
- This might be through the density function, distribution function, generating function etc
- But numerically there is another way of doing this at least approximately through a **large** sample of X i.e.

$$\{x_1, x_2, \cdots, x_{N.sample}\}$$

where $N.sample$ is very large

Definition.(MCMC)

- If we have a complex distribution $P(\theta)$ that is known up to a constant then it would seem that we would need to integrate to compute the normalising constant
- However, rather than working directly with the distribution, we can compute most statistical functions – such as means, variances, marginal distributions – using a large sample

$$\theta_1, \theta_2, \ldots, \theta_{N.sample}$$

- This is a Markov chain whose stationary (equilibrium) distribution is proportional to $P(\theta)$ and does not need any integration

Example.(Bayesian regression) Suppose we have a model

$$y_i|\boldsymbol{x}_i \sim N(\boldsymbol{x}_i\beta, \sigma^2)$$

with semi-conjugate independent priors

$$\begin{aligned} \beta &\sim MultiN_{p+1}\left(b_0, B_0^{-1}\right) \\ \sigma^2 &\sim InvGamma(c_0/2, \nu_0, d0^2/2) \end{aligned}$$

then the posterior is could be calculated but it is easier to use MCMC.

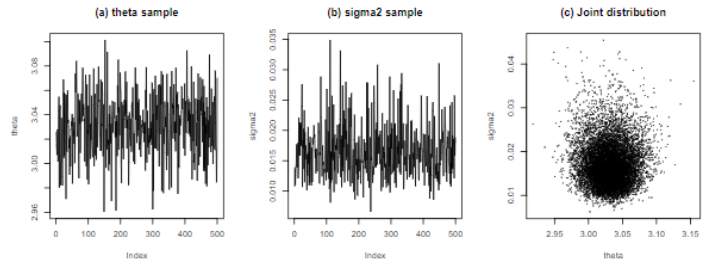


Figure: MCMC output for Example (a) θ sample, (b) σ^2 sample, (c) joint distribution

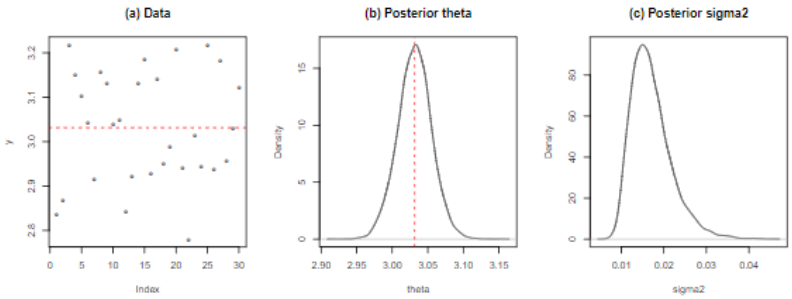


Figure: MCMC output

Introduction

- In this set we look again at the multicollinearity issue from the Bayesian point of view
- We also look at the penalty methods for model selection can be thought of as Bayesian methods
- We look quickly at how to make the MCMC methods operational in practice
- We start to look at the way that Bayesian methods can be used in state space models.

Bayesian Inference

***Example* (2.3.1 revisited).** We can return to the house price example of Chapter 2 and fit the model using MCMC.

```
> out <- MCMCregress(Y ~X1+X2+X3+X4+X5+X6+X7+X8+X9, data=house.price)
> summary(out)

                Mean       SD  Naive SE Time-series SE
(Intercept) 15.22298 6.45155 0.0645155      0.0645155
X1           1.93971 1.12235 0.0112235      0.0112235
X2           6.87051 4.67551 0.0467551      0.0480019
X3           0.13595 0.53313 0.0053313      0.0053274
X4           2.80602 4.75816 0.0475816      0.0475816
X5           2.04593 1.49712 0.0149712      0.0149712
X6          -0.50892 2.60459 0.0260459      0.0260459
X7          -1.29140 3.70783 0.0370783      0.0370783
X8          -0.03864 0.07188 0.0007188      0.0007188
X9           1.69108 2.09140 0.0209140      0.0209140
sigma2      10.25968 4.46910 0.0446910      0.0725849
```

- We get very similar estimates to the OLS solution.
- But now, since we are getting a joint posterior for all the parameters, we can see something interesting about the multicollinearity issue seen before.
- Recall that we found it odd that the estimate for β_6 was negative
- was explained due to dependence in the explanatory variables.
- In the Bayesian analysis this dependence flows very naturally into the posterior as shown in Fig. 3.

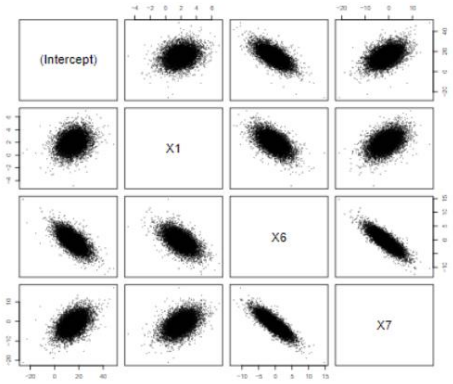


Figure: MCMC output for Example 2.3.1

Theorem.(Bayes and ridge regression) The Lasso estimate for linear regression parameters can be interpreted as a Bayesian posterior mode estimate when the priors on the regression parameters are independent double - exponential (Laplace) distributions.

Proof.
See Park and Casella (2008). □

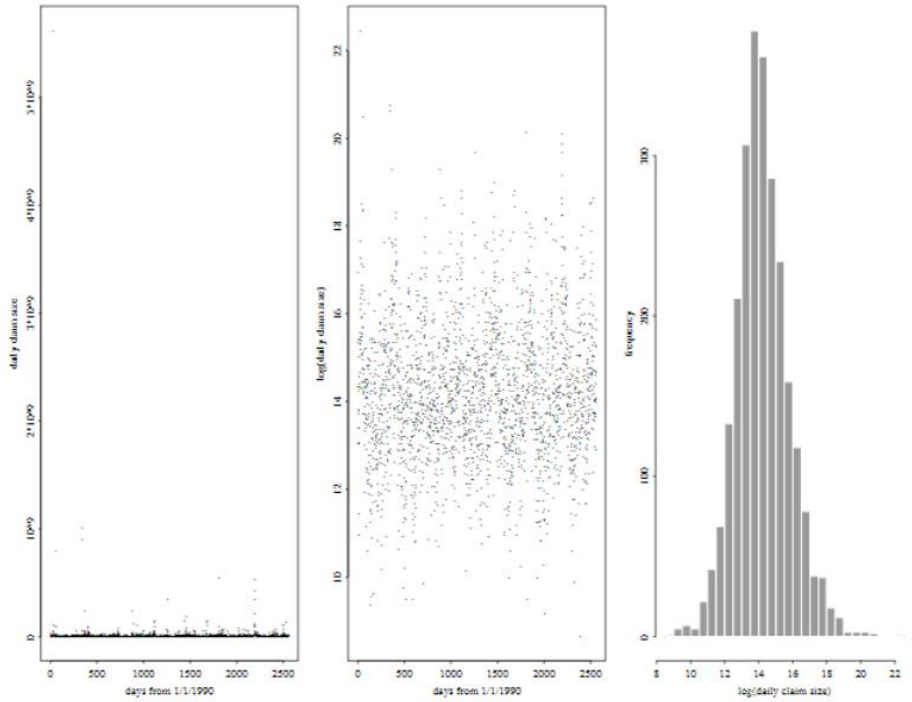
Lecture 4: Case Study

- Commercial insurance claim data study
- Complete set of commercial insurance claims taken over seven years in the UK
- Of particular importance is the investigation of the impact of very severe events on the overall record,
- Understanding of aggregated behaviour of the claims over a period of several days
- This is basically a complex, state space time series model. Used Bayesian methods to fit the data

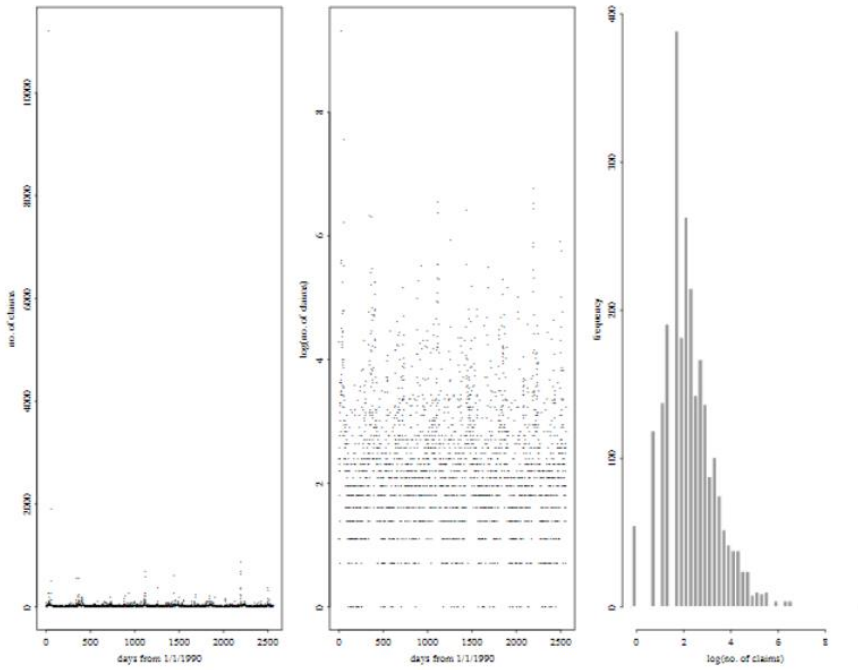
- *Temporal effects.* Is there any significant variation across the years in the observed data, apart from that caused by inflation? Any such variation should be identified and described.
- *Aggregation.* There is interest in the aggregation in the data, usually over three day periods. Suitable models for such summary statistics need to be found.
- *Extremes.* The data set contains two 'extreme' events. These were the storms in January and February 1990. The modelling should be able to predict return periods for extreme events such as these.

- For each claim the following information is available:
- the date of the reported claim incidence,
- the settlement date
- and the claim amount.

- Totalling all the claims occurring on a particular day creates a time series of total daily claims.
- or, look at time series created which contains the number of claims



: The total daily claim sizes from 1/1/1990 until 31/1



The number of claims on each day from 1/1/1990

- The physical cause of the damage claims will always be some metrological event, for example a storm or flood.
- These events are unobserved (unrecorded)
- We therefore model these metrological events as an unobserved state variable

- Let N_t denote the number of claims on day t.
- We shall consider the distribution of N_t conditionally on both the weather for that day and the number of policies at risk
- We have

$$N_t \sim \text{Poisson}(\lambda_t).$$

Note that the information about the number of policies acts as a multiplier for the rate parameter λ_t.

- We select

$$\log \lambda_t = H_t + \epsilon_t,$$

where H_t is a deterministic function of time which represents the fixed temporal effects, while ϵ_t represents the random weather effects

- A simple model for hidden state is

$$\epsilon_t = \rho \epsilon_{t-1} + \nu_t$$

where $\nu_t \sim N(0, \sigma_t^2)$.

- Deterministic part as

$$H_t = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 f^m(t) + \sum_{i=1}^{7} (\beta_i d_i(t)) + \sum_{i=1}^{12} (\gamma_i m_i(t))$$

where f^w is an indicator function which is 1 if t is the first day of the month, and 0 otherwise. While m_i is the indicator function for the i^{th} month of the year and d_i for the i^{th} day of the week.

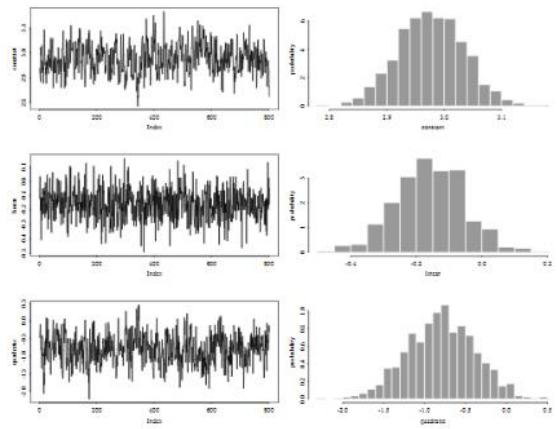


Figure: The trend effect: left hand panel shows output from MCMC algorithm, right hand panel showing marginal distribution.

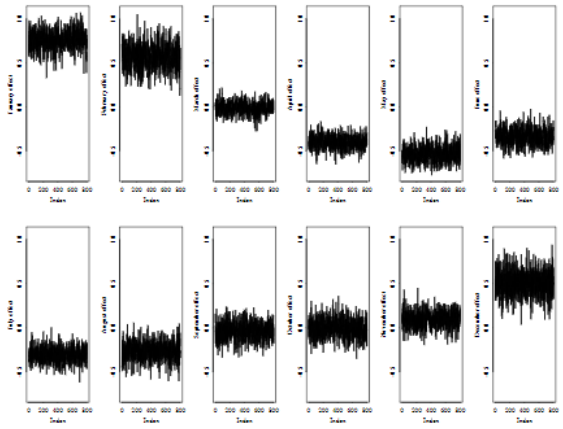


Figure: The seasonal monthly effect

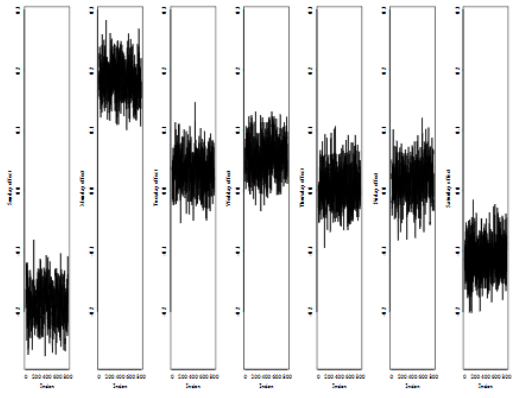


Figure: The weekly effect

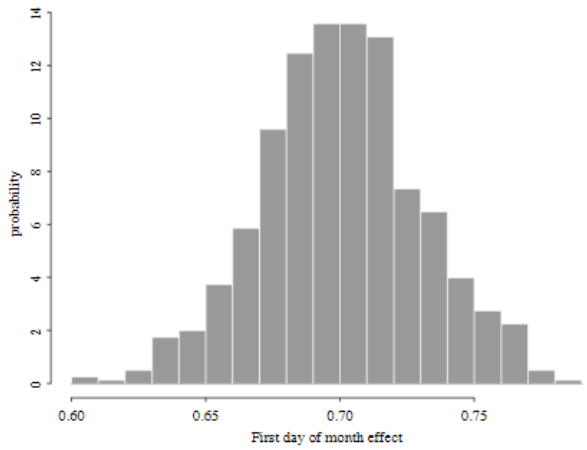


Figure: The first day of the month effect

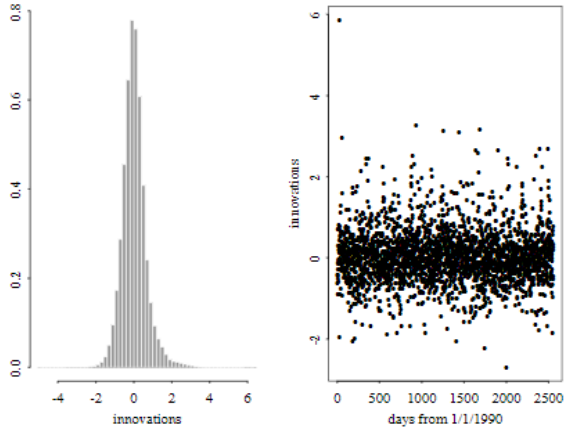


Figure: The innovation process; the left hand panel shows the marginal distribution, the right hand panel the time series structure

- Here just showing part of the analysis; the number of claims as a discrete time series
- Could try other forms of model, perhaps better suited to dealing with the extreme events
- Need to model the claims amounts in terms of number of claims and perhaps other covariates.

Lecture 5

INTRO

- In this set of slides we look briefly at models which can have 'non-constant variance'
- These are commonly used in finance where the data is not consistent with ARMA or ARIMA models
- These have a state space structure where the state is a model of the 'volatility'
- We look at ARCH and GARCH models
- We also look at regime shifting state space models

Example.7.0.1 (The Bollerslev-Ghysel benchmark dataset)

- The data in Fig. 7.1 from Bollerslev and Ghysels (1996) is the daily percentage nominal returns for the Deutschemark-Pound exchange rate.
- We see the non-constant volatility and volatility clustering that is common in such financial data.
- We also can see from the marginal plots that, where the dynamics is excluded, we get non-normal behaviour.
- The data is 'heavier-tail' than a constant variance Gaussian process.

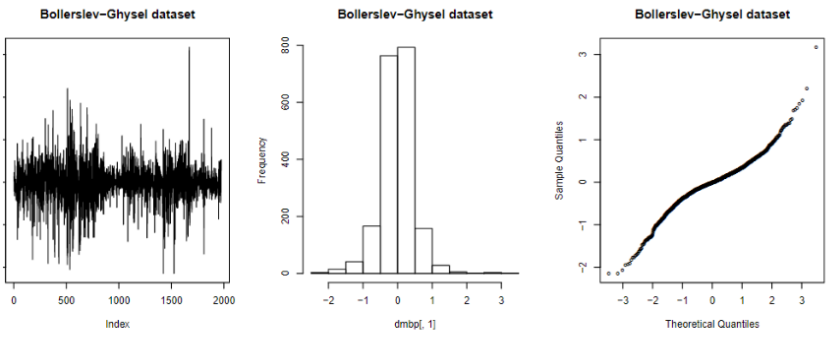


Figure: 7.1 Bollerslev-Ghysel benchmark dataset

ARCH

Definition.7.0.2 (ARCH model) An AutoRegressive Conditional Heteroscedasticity (ARCH(p)) model is defined hierarchically:
First define

$$X_t|\sigma_t = \sigma_t Z_t$$

where $Z_t \overset{i.i.d.}{\sim} N(0,1)$, but here σ_t is a random variable such that

$$\sigma_t^2 = \alpha_0 + \alpha_1 X_{t-1}^2 + \cdots + \alpha_p X_{t-p}^2$$

So the variance is 'time dependent' – a large value of X_t will result in period of high volatility.

Theorem.7.0.3 (Existence of stationary solution) For an ARCH(1) model we have, when $|\alpha_1| < 1$ there is a unique causal stationary solution of the equations of (7.0.2). It has the moment structure:

$$\begin{aligned} E(X_t) &= 0 \\ Var(X_t) &= \frac{\alpha_0}{1-\alpha_1} \\ Cov(X_{t+h}, X_t) &= 0 \end{aligned}$$

for $h > 0$.

- We will go through a proof of Theorem 7.0.3 in the next set of slides
- It will show that there is a stationary, but, non-linear representation in terms of the innovation process, $\{Z_t\}$.
- So its not in a $MA(\infty)$ representation but is stationary
- We note that if an ARCH(1) model then can generate a stationary process where the variance is 'time dependent' !
- This seems paradoxical at first

- We must be careful to differentiate between the unconditional variance

$$Var(X_t) = E(X_t^2) = \frac{\alpha_0}{1 - \alpha_1}$$

  which is a constant
- However the conditional variance

$$Var(X_t | Z_s, s \leq t) = \alpha_0 + \alpha_1 X_{t-1}^2$$

  clearly varies with the value of X_{t-1} so is time varying

Here is some simple code, (slightly altered from the version in notes):

```
 arch.sim <- function(n, alpha0, alpha1)
{
out <- rep(0, length=n)
for(i in 2:n)
{
out[i] <- sqrt(alpha0 + alpha1*out[i-1]^2)*rnorm(1)

}
  ts(out)
}
```

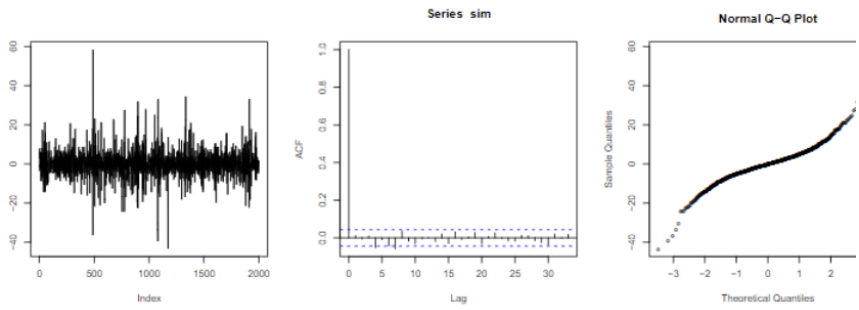


Figure: 7.2 Simulated ARCH(1) data

- Why are these models state space models?
- For ARCH(p) we have

$$\begin{aligned} X_t|\sigma_t &= \sigma_t Z_t \\ \sigma_t^2 &= \alpha_0 + \alpha_1 X_{t-1}^2 + \cdots + \alpha_p X_{t-p}^2 \end{aligned}$$

- The term X_t is observed directly
- Z_t is not observed (as usual)
- σ_t^2 is also not observed since α_i are not known
- Finally if σ_t^2 was known we would have a very simple model for X_t

ARCH & GARCH

- There are many variations on this theme:
- **Definition.**7.0.4 (GARCH model) The GARCH(p, q) model is defined by $X_t = \sigma_t Z_t$ where $Z_t \sim N(0, 1)$ i.i.d. and

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i X_{t-i}^2 + \sum_{i=1}^{q} \beta_i \sigma_{t-i}^2.$$

- As with Regression, Box-Jenkins we have the usual steps of statistical modelling
  1. Model Identification
  2. Estimation
  3. Diagnostic checking
  4. Consider alternative models if necessary
- The usual general arguments about bias-variance trade-off applies

Lecture 6

Intro

- This set of slides goes through a sketch of the proof of Theorem 7.0.3
- We will not be very formal about the mathematics of convergence but will be about the conditioning steps
- The emphasis is on understanding the non-linear representation
- Recall the definition of the ARCH(1) model as

$$\begin{aligned} X_t|\sigma_t &= \sigma_t Z_t \\ \sigma_t^2 &= \alpha_0 + \alpha_1 X_{t-1}^2 \end{aligned}$$

Proof of Theorem

Theorem. 7.0.3 (Existence of stationary solution)

- For an ARCH(1) model we have, when $|\alpha_1| < 1$ there is a unique causal stationary solution of the equations of (7.0.2).
- It has the moment structure:

$$\begin{aligned} E(X_t) &= 0, \\ Var(X_t) &= \frac{\alpha_0}{1-\alpha_1}, \\ Cov(X_{t+h}, X_t) &= 0, \end{aligned}$$

for $h > 0$.

- For an ARCH(1) model we have

$$\begin{aligned} X_t|\{\sigma_t\} &= \sigma_t Z_t, \\ \sigma_t^2 &= \alpha_0 + \alpha_1 X_{t-1}^2 \end{aligned}$$

where $Z_t \overset{i.i.d.}{\sim} N(0,1)$

- So

$$\begin{aligned} X_t^2 = \sigma_t^2 Z_t^2 &= \left\{\alpha_0 + \alpha_1 X_{t-1}^2\right\} Z_t^2 \\ &= \left\{\alpha_0 + \alpha_1 \left\{\alpha_0 + \alpha_1 X_{t-2}^2\right\} Z_{t-1}^2\right\} Z_t^2 \\ &= \left\{\alpha_0 + \alpha_1 \left\{\alpha_0 + \alpha_1 \left\{\alpha_0 + \alpha_1 X_{t-3}^2\right\} Z_{t-2}^2\right\} Z_{t-1}^2\right\} Z_t^2 \\ &= \alpha_0 \left[Z_t^2\right] + \alpha_0\alpha_1 \left[Z_t^2 Z_{t-1}^2\right] + \alpha_0\alpha_1^2 \left[Z_t^2 Z_{t-1}^2 Z_{t-2}^2\right] + \cdots \\ &\quad \cdots + \alpha_1^{n+1} X_{t-n-1}^2 \left[Z_t^2 Z_{t-1}^2 Z_{t-2}^2 \cdots Z_{t-n}^2\right] \end{aligned}$$

- If $|\alpha_1| < 1$ the last term tends to zero as $n \to \infty$ since the random terms do not get too large in probability
- So have

$$X_t^2 = \alpha_0 \sum_{j=0}^{\infty} \alpha_1^j Z_t^2 Z_{t-1}^2 Z_{t-2}^2 \cdots Z_{t-j}^2$$

- Hence immediately have

$$E(X_t^2) = \alpha_0 \sum_{j=0}^{\infty} \alpha_1^j = \frac{\alpha_0}{1-\alpha_1}$$

- So would have $Var(X_t) = \frac{\alpha_0}{1-\alpha_1}$ if we could show $E(X_t) = 0$

- For the rest of the argument we need to be clear about the causal structure
- Have representation that

$$X_t^2 = \alpha_0 \sum_{j=0}^{\infty} \alpha_1^j Z_t^2 Z_{t-1}^2 Z_{t-2}^2 \cdots Z_{t-j}^2$$

- We have that X_t is purely a function of $\{Z_s | s \leq t\}$ which we denote as

$$\mathcal{H}_t = \{Z_s, s \leq t\}$$

- Further, σ_t^2 is purely a function of $\mathcal{H}_{t-1}$

- Using this and the Law of total expectation (conditional expectation formula)

$$\begin{aligned} E(X_t) &= E_{\mathcal{H}_{t-1}}\left[E(X_t|\mathcal{H}_{t-1})\right] \\ &= E_{\mathcal{H}_{t-1}}\left(E(\sigma_t Z_t|\mathcal{H}_{t-1})\right) \\ &= E_{\mathcal{H}_{t-1}}\left[E(\sigma_t|\mathcal{H}_{t-1})E(Z_t|\mathcal{H}_{t-1})\right] \\ &= E_{\mathcal{H}_{t-1}}\left[E(\sigma_t|\mathcal{H}_{t-1})E(Z_t)\right] \\ &= E_{\mathcal{H}_{t-1}}\left[E(\sigma_t|\mathcal{H}_{t-1})0\right] \\ &= 0 \end{aligned}$$

- So have now

$$E(X_t) = 0 \Rightarrow Var(X_t) = \frac{\alpha_0}{1-\alpha_1}$$

- Similarly, again using conditional expectation formula, when $h > 0$

$$E(X_{t+h}X_t) = E_{\mathcal{H}_{t+h-1}}\left[E(X_{t+h}X_t|\mathcal{H}_{t+h-1})\right]$$

- Now since X_t is a function of $\mathcal{H}_t \subseteq \mathcal{H}_{t+h-1}$ we get

$$\begin{aligned} E(X_{t+h}X_t|\mathcal{H}_{t+h-1}) &= E_{\mathcal{H}_{t+h-1}}\left[E(X_t\sigma_{t+h}Z_{t+h}|\mathcal{H}_{t+h-1})\right] \\ &= E_{\mathcal{H}_{t+h-1}}\left[E(X_t\sigma_{t+h}Z_{t+h}|\mathcal{H}_{t+h-1})\right] \\ &= 0 \end{aligned}$$

- So, since the mean is zero we get

$$Cov(X_{t+h}, X_t) = 0$$

which completes the proof

Lecture 7

Intro

- The set of slides looks at other state space models
- These are the hidden Markov models discussed when we first introduced the state space idea
- These are also similar to the model used in the case study
- Here though the hidden state will be discrete

- **Definition.**6.1.1 (State space model) If $\{Y_t\}$ is an (observed) time series with (unobserved) state process $\{\theta_t\}$. In a state space model the dependence between these is defined by the graphical model shown in Fig. 1

Figure: State space structure

- Suppose that θ_i is a two state Markov chain and that Y_i is normal with a mean and variance which only depend on the state θ_i

- The states can represent the state of the market
- State 1: the market is growing so positive mean and small volatility
- State 2: the market is in recession so negative mean and high volatility
- We transition between states with a transition matrix of the form

$$\begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix}$$

with p, q quite small so that market stays in the same state for a long time.

- Consider the following simple simulation
- There are two unobserved states, red and green
- When in the red state the daily change in price is modelled as $N(0.1, 0.5^2)$
- When in the green state the daily change in price is modelled as $N(-0.2, 2^2)$
- The probability of staying in the red state is 0.99
- The probability of staying in the green state is 0.98
- We plot both the changes in price and the price itself

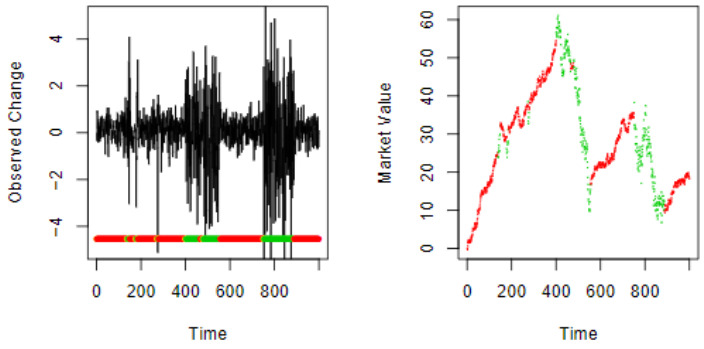


Figure: State space experiment

- If we were fitting this model to data we have to deal with the fact that the state is not observed in practice
- This is called an hidden markov model
- It can be fitted using the EM algorithm or with Bayesian methods

- We can also look at the marginal distribution of the changes in price
- We see the 'heavy tail' issue
- This is because we are modelling the changes in price with a mixture of normals
- If we knew the state we can do a better job of explaining the 'heavy tails'

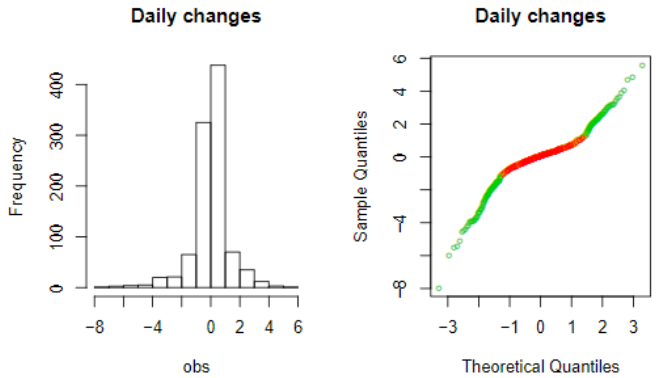


Figure: State space experiment

Lecture 8

Intro

- In this set of slides we look at the Kalman filter
- This is a recursive algorithm which can run in 'real time' – that is at the same rate that the data is collected
- It is a state space model which is forecasting the current value of an unobserved state (filtering)
- It was famous for being a important part of the Apollo moon landing
- It has many applications currently and we will look at a ball tracking problem in baseball and other sports

- In a general state space model we have an (unobserved) state variable at time s and a set of observations $y_1, \ldots, y_t$
- We subdivide problems by:
  (i) *filtering* is the case where (s=t),
  (ii) *state prediction* is the case when $s > t$ and
  (iii) *smoothing* is the case $s < t$.
- If we are filtering we might was to do the state prediction in a time which is faster than the rate that new data arrives
- This is critical for 'real time' control applications
- With real time control problems you typically need to be able to compute predictions faster than you are gathering data.

- There are many different forms of the notation that are used for setting up the Kalman filter
- Here I will use that found in *Time Series Analysis and its Applications* by Shumway and Stoffer (Not the same as lecture notes)
- A version of the book can be downloaded from Stoffer's site
  `https://www.stat.pitt.edu/stoffer/tsa4/`
  which you can download when on campus using Waterloo's *SpringerLink* argement
- The site also has lots of good R code and libraries

Illustrative Example

- It is common to see, when watching sports like baseball, cricket and tennis ball-tracking algorithms
- These can be used to make decisions: in tennis if a ball was on the line or out; in cricket decide on a LBW decision.
- Also they are used to report statistics such as the distance a home run was hit in baseball.
- The tracking comes from camera information, and the underlying Physics, and have to be done in real time, or very close to it.
- We will explore a simplified version of one of these algorithms

- Lets first consider the Physics of these models
- This is through well-known and validated equations which explain the path of a ball
- They are differentiatial equation models that take into account gravity, air resistance, spin *etc*
- For our example here we will simply use the dynamics associated with gravity for illustration
- We will also work in one vertical dimension for simplicity and clarity

- We have the ball's height at time t – which to start with is continuous – is $z(t)$.
- We have, from physics theory, that

$$\frac{d^2z}{dt^2}(t) = -g,$$

- We can solve this linear differential equation to give

$$z(t) = z(t_0) + \frac{dz}{dt}(t_0)(t - t_0) - \frac{g}{2}(t - t_0)^2.$$

  where t_0 is some initial time.
- We do **not** need to have an analytic solution in fact, but it will make this presentation easier to follow.

- If we now discretise time, so that we have $t - t_0 = \epsilon$, we have by setting $z(t) = z_t$ and $\frac{dz}{dt}(t) = \dot{z}_t$,

$$z_{t+1} = z_t + \epsilon \dot{z}_t - \epsilon^2 \frac{g}{2},$$

- Furthermore we have from the original equation

$$\frac{d^2 z}{d\tau^2}(\tau) = -g$$

- By the finite difference method,

$$-g = \frac{d^2 z}{d\tau^2}(\tau) \approx \frac{\frac{dz}{d\tau}(\tau + \epsilon) - \frac{dz}{d\tau}(\tau)}{\epsilon}$$

In terms of discrete time

$$\begin{aligned} z_{t+1} &= z_t + \epsilon \dot{z}_t - \epsilon^2 \frac{g}{2} \\ \dot{z}_{t+1} &\approx \dot{z}_t - \epsilon g \end{aligned}$$

Let us therefore define the state vector $\mathbf{x}_t := (z_t, \dot{z}_t)^T$ and so, the physics gives us the state equation

$$\mathbf{x}_{t+1} = \begin{pmatrix} 1 & \epsilon \\ 0 & 1 \end{pmatrix} \mathbf{x}_t - \begin{pmatrix} 0.5\epsilon^2 \\ \epsilon \end{pmatrix} g$$

- The discretisation error in the above calculation can be thought of as an error term giving the state equation

$$\begin{aligned} \mathbf{x}_{t+1} &= \begin{pmatrix} 1 & \epsilon \\ 0 & 1 \end{pmatrix} \mathbf{x}_t - \begin{pmatrix} 0.5\epsilon^2 \\ \epsilon \end{pmatrix} g + \mathbf{w_t} \\ &\overset{\text{denoted by}}{=} \Phi \mathbf{x}_{t-1} + \Upsilon + \mathbf{w}_t \end{aligned}$$

- This determines the theoretical behaviour of the body, but might not exactly what is observed due to noise
- In our example have a camera which records a (noisy) measurement of $\mathbf{x}_t$.

$$\begin{aligned} \mathbf{y}_t &= z_t + v_t = \begin{pmatrix} 1 & 0 \end{pmatrix} \mathbf{x}_t + \mathbf{v}_t. \\ &\overset{\text{denoted by}}{=} A_t \mathbf{x}_t + \mathbf{v}_t. \end{aligned}$$

- The terms Φ, A_t come from the Physics and understanding the measurement system
- The distributions of $\mathbf{w_t}$ and $\mathbf{v}_t$ are found by calibration experiments with, in the example shown here

$$\mathbf{w_t} \overset{i.i.d.}{\sim} N\left((0,0)^T, \begin{pmatrix} 2\times 10^{-5} & 0 \\ 0 & 10^{-2} \end{pmatrix}\right)$$
$$\mathbf{v_T} \overset{i.i.d.}{\sim} N(0, 10^{-2})$$

  and $\epsilon = 2 \times 10^{-3}$.
- In this example the laws of physics and calibration experiments give us information which is clearly highly informative and can be used for forecasting and filtering
- In this case the model is coming from known Physics and not an empirical model selection process like Box-Jenkins

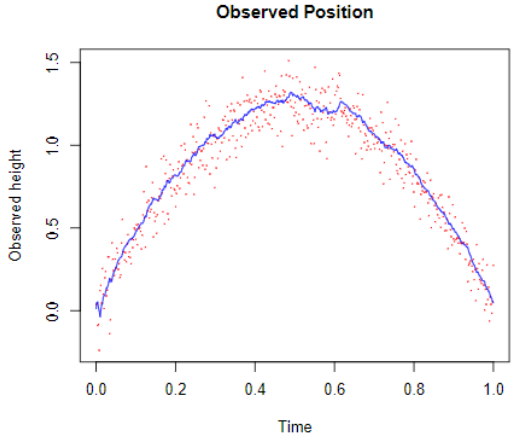


Figure: Kalman Filter output

- The KF makes a one step ahead prediction
- As the next new noisy information (red dots) comes in the KF almost immediately updates this estimate of the (now) current value
- In some ways it behaves like exponential smoothing or Holt-Winters
- It also updates its estimate of uncertainty in real time
- It also estimates the full state variable even thought it is of larger dimension than the observation

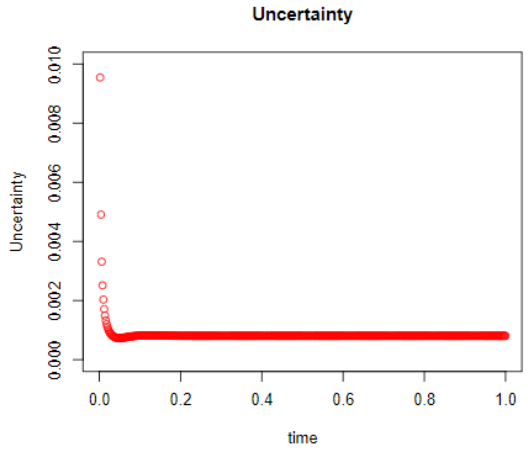


Figure: Kalman Filter output

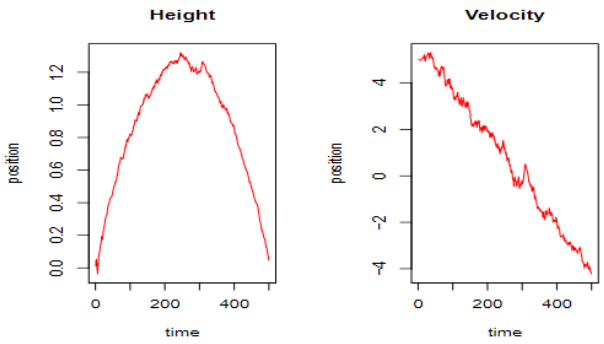


Figure: Kalman Filter output

Lecture 9

Intro

- In this set of slides we look at the Kalman filter
- This is a recursive algorithm which can run in 'real time' – that is at the same rate that the data is collected
- It is a state space model which is forecasting the current value of an unobserved state
- We looked at a simple ball tracking problem
- Here we look at the algorithm's general structure

Notation

Definition.(State Structure for KF) Consider a p-dimensional state variable $\mathbf{x}_t$ which satisfies the *state equation*

$$\mathbf{x}_t = \Phi \mathbf{x}_{t-1} + \Upsilon \mathbf{u}_t + \mathbf{w}_t \tag{1}$$

where $\mathbf{u}_t$ are a set of control or input variables and $\mathbf{w}_t \overset{i.i.d}{\sim} MVN_p(0, Q)$
The observed part of the system is defined via the *observation* equation

$$\mathbf{y}_t = A_t \mathbf{x}_t + \Gamma \mathbf{u}_t + \mathbf{v}_t \tag{2}$$

where $\mathbf{y}$ is q-dimensional and $\mathbf{v}_t \overset{i.i.d}{\sim} MVN_q(0, R)$ and the two noise terms, $\mathbf{w}, \mathbf{v}$ are independent of one another

- In our case the matrices $\Phi, \Upsilon, A_t, \Gamma$ are known from the physics and variances Q, R in (1) and (2) are considered known from calibration
- The state variable $\mathbf{x}_t$ is the object of interest and is unobserved
- We think of the observation variable, $\mathbf{y}_t$, as a transformed noisy version of the state variable which we do observe
- We also may want to use $\mathbf{u}_t$ to represent controls or outside influences from physics

- Since Equation (1) is recursive we require an initial condition $\mathbf{x}_0$ which is $MVN_p(0, \Sigma_0)$
- This is assumed to be independent of the noise terms
- Again in the simplest example we will consider Σ_0 known from calibrations

Algorithm

- The algorithm at each time period calculates a *prediction* which is then *corrected* as soon as new information become available.
- Each step is a low dimensional calculation
- Let us define
$$\mathbf{x}_t^s := E(\mathbf{x}_t|\mathbf{y}_1, \dots \mathbf{y}_s)$$
to be the best mean square estimate of $\mathbf{x}_t$ based on $Y_s := (\mathbf{y}_1, \dots, \mathbf{y}_s)$
- and
$$P_{t_1,t_2}^s := E\left((\mathbf{x}_{t_1} - \mathbf{x}_{t_1}^s)(\mathbf{x}_{t_2} - \mathbf{x}_{t_2}^s)^T\right)$$
with this being P_t^s when $t_1 = t_2 = t$

Definition.(Kalman filter) For the state space model defined by (1) and (2) we have from the conditional mean

$$\mathbf{x}_t^{t-1} = \Phi \mathbf{x}_{t-1}^{t-1} + \Upsilon \mathbf{u}_t \tag{3}$$

This is the *prediction step*. It has a mse which satisfies:

$$P_t^{t-1} = \Phi P_{t-1}^{t-1} \Phi^T + Q \tag{4}$$

We can then also predict the next observed value by

$$\mathbf{y}_t^{t-1} = A_t \mathbf{x}_t^{t-1} + \Gamma \mathbf{u}_t$$

Definition.(Continued) The *correction step* or *update* is defined by

$$\mathbf{x}_t^t = \mathbf{x}_t^{t-1} + K_t\left(\mathbf{y}_t - A_t \mathbf{x}_t^{t-1} - \Gamma \mathbf{u}_t\right) \tag{5}$$

It has a MSE/Posterior variance which satisfies:

$$P_t^t = [I - K_t A_t] P_t^{t-1} \tag{6}$$

where the term K_t is called the *Kalman gain* and is defined by

$$K_t := P_t^{t-1} A^T \left[A_t P_t^{t-1} A_t^T + R\right]^{-1} \tag{7}$$

- The method works recursively: first using (3) to compute $\mathbf{x}_t^{t-1}$, then predicting the up-coming value of the observed process

$$\mathbf{y}_t^{t-1} := A_t \mathbf{x}_t^{t-1} + \Gamma \mathbf{u}_t$$

- As soon the new information comes in (i.e. $\mathbf{y}_t$) it updates the prediction of the state using (5) with the error correction form

$$\mathbf{x}_t^t = \mathbf{x}_t^{t-1} + K_t \left(\mathbf{y}_t - \mathbf{y}_t^{t-1} \right)$$

- Then (6) and (7) updates the Kalman gain and the uncertainty
- It is important to note that the matrix inversion needed for the gain K_t has a low dimension which does not get bigger as the number of observations gets bigger
- In the physics example it is always a 2×2 matrix inverse

Lecture 10

Intro

- In this set of slides we look at the details of the Kalman filter theory
- We show the link to error correcting methods like exponential smoothing and Holt-Winters
- We also show the link to the Bayesian ideas of up-dating a prior distribution to a posterior as new data arrives
- The algebra and notation can hide the essentially simple structure of the method

- Here is a simplified version of a Bayesian example for the case $y \sim N(\theta, 1)$
- If the prior distribution for θ was in the Normal family i.e.

$$\theta \sim N(\theta; \mu_{\text{prior}}, \sigma^2_{\text{prior}}) \propto \exp\left[-\frac{(\theta - \mu_{\text{prior}})^2}{2\sigma^2_{\text{prior}}} \right]$$

- Then the posterior, after seeing y, is

$$\begin{aligned} Pr(\theta|y) &\propto Lik(\theta) \times \text{prior}(\theta) \\ &= N\left(\frac{(\mu_{\text{prior}} + y\sigma^2_{\text{prior}})}{(1 + \sigma^2_{\text{prior}})}, \frac{\sigma^2_{\text{prior}}}{(1 + \sigma^2_{\text{prior}})} \right) \end{aligned}$$

- Have iterative update rules for the posterior mean

$$\mu_{\text{prior}} \to \mu_{\text{posterior}} := \frac{(\mu_{\text{prior}} + y\sigma^2_{\text{prior}})}{(1 + \sigma^2_{\text{prior}})} \overset{\text{more data}}{\to} \cdots \overset{\text{more data}}{\to} E(Y)$$

Kalman Filter

- Recall we have the Kalman Filter state space structure

$$\mathbf{x}_t = \Phi \mathbf{x}_{t-1} + \Upsilon \mathbf{u}_t + \mathbf{w}_t \quad (1)$$
$$\mathbf{y}_t = A_t \mathbf{x}_t + \Gamma \mathbf{u}_t + \mathbf{v}_t \quad (2)$$

- $\mathbf{u}_t$ are a set of control or input variables and $\mathbf{w}_t \overset{i.i.d}{\sim} \text{MVN}_p(0, Q)$,
- $\mathbf{y}$ is q-dimensional and $\mathbf{v}_t \overset{i.i.d}{\sim} \text{MVN}_q(0, R)$
- the two noise terms, $\mathbf{w}, \mathbf{v}$ are independent of one another

- We want to get insight of how the structure of Equations (1, 2) lead to the prediction-correction iterative cycle of the Kalman filter and it's error correction structure
- We want to explore the links with Bayesian updating from prior to posterior and the Kalman filter
- We will highly simplify notation to get to the essentials
- Drop the control terms $\Upsilon \mathbf{u}_t, \Gamma \mathbf{u}_t$,
- Work in one-dimension case

- Our estimates will be conditional expectations – i.e posterior means
- We work with normality assumptions
- Define $\mathcal{H}_s = \{y_s, y_{s-1}, \cdots\}$.
- Our target is to compute $E(X_t|\mathcal{H}_t)$ though an update rule:
  1. Predict using $E(X_t|\mathcal{H}_{t-1})$
  2. Correct when y_t becomes available.
- Now suppose that conditionally on $\mathcal{H}_{t-1}$ we have 'priors and likelihood'

$$\begin{aligned} X_t|\mathcal{H}_{t-1} &\sim N(\Phi x_{t-1}, P_{t-1}) \\ &\overset{\text{denote by}}{=} N(\mu_{t-1}, P_{t-1}) \quad (3) \\ Y_t|x_t &\sim N(Ax_t, R) \quad (4) \end{aligned}$$

- Lets look at the Bayesian interpretation of Equations (3-4)
- Equation (3) describes how much we know about the state at time t given the observations up to $t-1$
- It a full distribution having conditional expectation as point estimation and conditional variance to measure uncertainty: its our prior at time $t-1$
- Equation (4) describes the uncertainty in the observation given we know the state at time t
- We are going to combine them to get the posterior which tells us what we know at time t.

- One key assumption is that of Normality here.
- We will see that if Equation (3) is Normal at stage $t-1$ the corresponding term will also be normal at stage t.
- This assumption can be analysed further but if t is large it is usually reasonable to assume normality
- We can think of Equation (3) as being a prior for X_t, and Equation (4) as the likelihood
- We are updating prior mean and prior variance to posterior mean and posterior variance

- In terms of the Bayesian setting and notation we have

$$\begin{aligned} Lik(y_t) &= P(y_t|x_t) = \frac{1}{\sqrt{2\pi R}} \exp\left[-\frac{(y_t - Ax_t)^2}{2R}\right] \\ \text{Prior}(x_t) &= \frac{1}{\sqrt{2\pi P_{t-1}}} \exp\left[-\frac{(x_t - \mu_{t-1})^2}{2P_{t-1}}\right] \end{aligned}$$

- So get posterior

$$\begin{aligned} P(x_t|y_t) &\propto \exp\left[-\frac{(y_t - Ax_t)^2}{2R}\right] \exp\left[-\frac{(x_t - \mu_{t-1})^2}{2P_{t-1}}\right] \\ &\propto \exp\left\{-\frac{1}{2}\left[\left(\frac{A^2}{R} + \frac{1}{P_{t-1}}\right)x_t^2 - 2\left(\frac{\mu_{t-1}}{P_{t-1}} + \frac{y_t A}{R}\right)x_t\right]\right\} \\ &:= N(\mu_t, P_t) \end{aligned}$$

- So have updated our knowledge

$$N(\mu_{t-1}, P_{t-1}) \xrightarrow{y} N(\mu_t, P_t)$$

- Direct calculation gives the posterior variance and mean as

$$\begin{aligned} P_t &= \left(\frac{A^2}{R} + \frac{1}{P_{t-1}}\right)^{-1} = \frac{RP_{t-1}}{A^2P_{t-1}+R} \\ \mu_t &= \frac{\left(\frac{\mu_{t-1}}{P_{t-1}} + \frac{y_t A}{R}\right)}{\left(\frac{A^2}{R} + \frac{1}{P_{t-1}}\right)} = \mu_{t-1} + K_t(y - A\mu_{t-1}) \end{aligned}$$

  by defining

$$K_t := \frac{AP_{t-1}}{A^2P_{t-1}+R}$$

  i.e. the gain.

- So have have the update rule which takes the form

$$\begin{aligned} \mu_{t-1} \to \mu_t &:= \mu_{t-1} + K_t(y - A\mu_{t-1}) \\ &= \mu_{t-1} + K_t \times (\text{error in prediction of } y) \\ P_{t-1} &\to P_t := (I - K_tA)P_{t-1} \end{aligned}$$

- This form is the one dimensional version of the general case.
- It shows directly the error correcting form.
- It defines K as the sample dependent - and therefore non-constant – version of the training parameter in exponential smoothing
- The generalisation to the general matrix form is simply (long) algebra