Chapter 11

Kalman Filters

11.1 Introduction

We describe Bayesian Learning for sequential estimation of parameters (eg. means, AR coefficients). The update procedures are known as Kalman Filters. We show how Dynamic Linear Models, Recursive Least Squares and Steepest Descent algorithms are all special cases of the Kalman filter.

11.1.1 Sequential Estimation of Nonstationary Mean

In the lecure on Bayesian methods we described the sequential estimation of a stationary mean. We now extend that analysis to the nonstationary case.

A reasonable model of a time varying mean is that it can drift from sample to sample. If the drift is random (later on we will also consider deterministic drifts) then we have

$$\mu_t = \mu_{t-1} + w_t \tag{11.1}$$

where the random drift is Gaussian $p(w_t) = N(w_t; 0, \sigma_w^2)$ with drift variance σ_w^2. The data points are then Gaussian about mean μ_t. If they have a *fixed* variance σ_x^2 (later on we will also consider time-varing variance)

$$x_t = \mu_t + e_t \tag{11.2}$$

where $e_t = x_t - \mu_t$. Hence $p(e_t) = N(e_t; 0, \sigma_x^2)$.

At time $t-1$ our estimate of μ_{t-1} has a Gaussian distribution with mean $\hat{\mu}_{t-1}$ and variance $\hat{\sigma}_{t-1}^2$. We stress that this is the variance of our mean estimate and not the variance of the data. The standard error estimate for this variance (σ_t^2/t) is no longer valid as we have nonstationary data. We therefore have to estimate it as we go along.

This means we keep running estimates of the distribution of the mean. At time $t-1$ this distribution has a mean $\hat{\mu}_{t-1}$ and a variance $\hat{\sigma}_{t-1}^2$. The distribution at time t is then found from Bayes rule. Specifically, the prior distribution is given by

$$p(\mu_t) = N(\mu_t; \hat{\mu}_{t-1}, r_t) \tag{11.3}$$

where r_t is the prior variance (we add on the random drift variance to the variance from the previous time step)

$$r_t = \hat{\sigma}_{t-1}^2 + \sigma_w^2 \tag{11.4}$$

and the likelihood is

$$p(x_t|\mu_t) = N(x_t; \hat{\mu}_{t-1}, \sigma_x^2) \tag{11.5}$$

The posterior is then given by

$$p(\mu_t|x_t) = N(\mu_t; \hat{\mu}_t, \hat{\sigma}_t^2) \tag{11.6}$$

where the mean is

$$\hat{\mu}_t = \hat{\mu}_{t-1} + \frac{r_t}{\sigma_x^2 + r_t}(x_t - \hat{\mu}_{t-1}) \tag{11.7}$$

and the variance is

$$\hat{\sigma}_t^2 = \frac{r_t \sigma_x^2}{r_t + \sigma_x^2} \tag{11.8}$$

We now write the above equations in a slightly different form to allow for comparison with later estimation procedures

$$\begin{aligned} \hat{\mu}_t &= \hat{\mu}_{t-1} + K_t e_t \\ \hat{\sigma}_t^2 &= r_t(1 - K_t) \end{aligned} \tag{11.9}$$

where

$$K_t = \frac{r_t}{\sigma_x^2 + r_t} \tag{11.10}$$

and

$$e_t = x_t - \hat{\mu}_{t-1} \tag{11.11}$$

In the next section we will see that our update equations are a special case of a *Kalman filter* where e_t is the prediction error and K_t is the *Kalman gain*.

In figure 11.1 we give a numerical example where 200 data points were generated; the first 100 having a mean of 4 and the next 100 a mean of 10. The update equations have two paramaters which we must set (i) the data variance σ_x^2 and (ii) the drift variance σ_w^2. Together, these parameters determine (a) how responsive the tracking will be and (b) how stable it will be. The two plots are for two different values of σ_w^2 and $\sigma_x^2 = 1$. Later we will see how these two parameters can be learnt.

11.1.2 A single state variable

We now look at a general methodology for the sequential estimation of a nonstationary parameter (this can be anything - not necesarily the data mean).

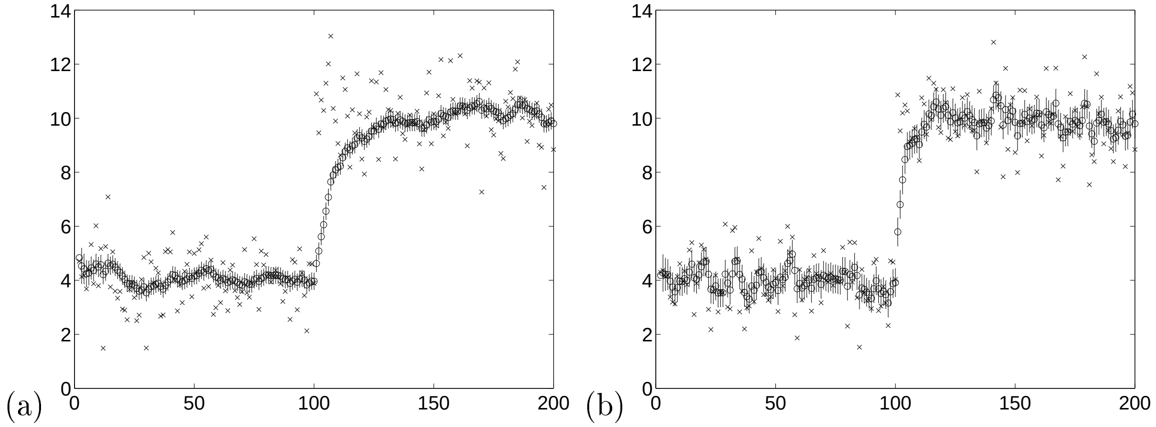

Figure 11.1: **Sequential estimation of nonstationary mean.** The graphs plot data values x_t (crosses) and estimated mean values $\hat{\mu}_t$ (circles) along with error bars $\hat{\sigma}_t$ (vertical lines) versus iteration number t for two different drift noise values (a) $\sigma_w^2 = 0.01$ and (b) $\sigma_w^2 = 0.1$.

The parameter's evolution is modelled as a *linear dynamical system*. The *state-space* equations are

$$\begin{array}{rclcrcl} \theta_t & = & g_t \theta_{t-1} + w_t, & & w_t & \sim & N(w_t; 0, \sigma_w^2) \\ x_t & = & f_t \theta_t + e_t, & & e_t & \sim & N(e_t; 0, \sigma_x^2) \end{array} \tag{11.12}$$

The value of the parameter at time t is referred to as the *state* of the system θ_t. This state can change deterministically, by being multiplied by g_t, and stochastically by added a random drift w_t. This drift is referred to as *state noise*. The observed data (eg. time series values) are referred to as *observations* x_t which are generated from the state according to the second equation. This allows for a linear transformation plus the addition of *observation noise*.

At time $t-1$ our estimate of θ_{t-1} has a Gaussian distribution with mean $\hat{\theta}_{t-1}$ and variance $\hat{\sigma}_{t-1}^2$. The prior distribution is therefore given by

$$p(\theta_t) = N(\theta_t; g_t \hat{\theta}_{t-1}, r_t) \tag{11.13}$$

where r_t is the prior variance

$$r_t = g_t^2 \hat{\sigma}_{t-1}^2 + \sigma_w^2 \tag{11.14}$$

and the likelihood is

$$p(x_t|\theta_t) = N(x_t; f_t \hat{\theta}_{t-1}, \sigma_x^2) \tag{11.15}$$

The posterior is then given by

$$p(\theta_t|x_t) = N(\theta_t; \hat{\theta}_t, \hat{\sigma}_t^2) \tag{11.16}$$

where

$$\begin{array}{rcl} \hat{\theta}_t & = & g_t \hat{\theta}_{t-1} + K_t e_t \\ \hat{\sigma}_t^2 & = & r_t (1 - K_t f_t) \end{array} \tag{11.17}$$

and

$$K_t = \frac{r_t}{\sigma_x^2 + f_t^2 r_t} f_t \tag{11.18}$$

The above equations constitute a 1-dimensional *Kalman Filter* (the state is 1-dimensional because there is only 1 state variable). Next we consider many state variables.

11.1.3 Multiple state variables

We now consider linear dynamical systems where data is generated according to the model

$$\begin{aligned} \boldsymbol{\theta}_t &= \boldsymbol{G}_t\boldsymbol{\theta}_{t-1} + \boldsymbol{w}_t, & \boldsymbol{w}_t &\sim N(\boldsymbol{w}_t; 0, \boldsymbol{W}_t) \\ \boldsymbol{y}_t &= \boldsymbol{F}_t\theta_t + \boldsymbol{v}_t, & \boldsymbol{v}_t &\sim N(\boldsymbol{v}_t; 0, \boldsymbol{V}_t) \end{aligned} \tag{11.19}$$

where $\boldsymbol{\theta}_t$ are 'state' or 'latent' variables, $\boldsymbol{G}_t$ is a 'flow' matrix, $\boldsymbol{w}_t$ is 'state noise' distributed according to a normal distribution with zero mean and covariance matrix $\boldsymbol{W}_t$, $\boldsymbol{y}_t$ are the multivariate observations, $\boldsymbol{F}_t$ is a transformation matrix and $\boldsymbol{v}_t$ is 'observation noise' distributed according to a normal distribution with zero mean and covariance matrix $\boldsymbol{V}_t$. The model is parameterised by the matrices $\boldsymbol{G}_t$, $\boldsymbol{W}_t$, $\boldsymbol{F}_t$ and $\boldsymbol{V}_t$. These parameters may depend on t (as indicated by the subscript).

The Kalman filter is a recursive procedure for estimating the latent variables, $\boldsymbol{\theta}_t$ [29]. Meinhold and Singpurwalla [40] show how this estimation procedure is derived (also see lecture on Bayesian methods). The latent variables are normally distributed with a mean and covariance that can be estimated with the following recursive formulae

$$\begin{aligned} \hat{\boldsymbol{\theta}}_t &= \boldsymbol{G}_t\hat{\boldsymbol{\theta}}_{t-1} + \boldsymbol{K}_t\boldsymbol{e}_t \\ \Sigma_t &= \boldsymbol{R}_t - \boldsymbol{K}_t\boldsymbol{F}_t\boldsymbol{R}_t \end{aligned} \tag{11.20}$$

where $\boldsymbol{K}_t$ is the 'Kalman gain' matrix, $\boldsymbol{e}_t$ is the prediction error and $\boldsymbol{R}_t$ is the 'prior covariance' of the latent variables (that is, prior to $\boldsymbol{y}_t$ being observed). These quantities are calculated as follows

$$\begin{aligned} \boldsymbol{K}_t &= \boldsymbol{R}_t\boldsymbol{F}_t^T\left(\boldsymbol{V}_t + \boldsymbol{F}_t\boldsymbol{R}_t\boldsymbol{F}_t^T\right)^{-1} \\ \boldsymbol{e}_t &= \boldsymbol{y}_t - \boldsymbol{F}_t\boldsymbol{G}_t\hat{\boldsymbol{\theta}}_{t-1} \\ \boldsymbol{R}_t &= \boldsymbol{G}_t\boldsymbol{\Sigma}_{t-1}\boldsymbol{G}_t^T + \boldsymbol{W}_t \end{aligned} \tag{11.21}$$

To apply these equations you need to know the parameters $\boldsymbol{G}_t$, $\boldsymbol{W}_t$,$\boldsymbol{F}_t$ and $\boldsymbol{V}_t$ and make initial guesses for the state mean and covariance; $\hat{\boldsymbol{\theta}}_0$ and $\boldsymbol{\Sigma}_0$. Equations (3) and (2) can then be applied to estimate the state mean and covariance at the next time step. The equations are then applied recursively.

A useful quantity is the likelihood of an observation given the model parameters before they are updated

$$p(\boldsymbol{y}_t) = N\left(\boldsymbol{y}_t; \boldsymbol{F}_t\hat{\boldsymbol{\theta}}_{t-1}, \boldsymbol{V}_t + \boldsymbol{F}_t\left(\boldsymbol{G}_t^T\boldsymbol{\Sigma}_{t-1}\boldsymbol{G}_t\right)\boldsymbol{F}_t^T\right) \tag{11.22}$$

In Bayesian terminology this likelihood is known as the evidence for the data point [14]. Data points with low evidence correspond to periods when the statistics of the underlying system are changing (non-stationarity) or, less consistently, to data points having large observation noise components.

The state-space equations may be viewed as a dynamic version of factor analysis where the factor, $\boldsymbol{\theta}_t$, evolves over time according to linear dynamics. Shumway and Stoffer [56] derive an Expectation-Maximisation (EM) algorithm (see next lecture) in which the parameters of the model $\boldsymbol{G}$, $\boldsymbol{W}$ and $\boldsymbol{V}$ can all be learnt. Only $\boldsymbol{F}$ is assumed known. Note that these parameters are no longer dependent on t. This does not, however, mean that the model is no longer dynamic; the state, $\boldsymbol{\theta}_t$, is still time dependent. Ghahramani and Hinton [22] have recently extended the algorithm to allow $\boldsymbol{F}$ to be learnt as well. These learning algorithms are batch learning algorithms rather than recursive update procedures. They are therefore not suitable for 'on-line' learning (where the learning algorithm has only one 'look' at each observation).

In the engineering and statistical forecasting literature [44] [11] the transformation matrix, $\boldsymbol{F}_t$, is known. It is related to the observed time series (or other observed time series) according to a known deterministic function set by the statistician or 'model builder'. Assumptions are then made about the flow matrix, $\boldsymbol{G}_t$. Assumptions are also made about the state noise covariance, $\boldsymbol{W}_t$, and the observation noise covariance, $\boldsymbol{V}_t$, or they are estimated on-line. We now look at a set of assumptions which reduces the Kalman filter to a 'Dynamic Linear Model'.

11.1.4 Dynamic Linear Models

In this section we consider Dynamic Linear Models (DLMs) [11] which for a univariate time series are

$$\begin{array}{rcllcl} \boldsymbol{\theta}_t & = & \boldsymbol{\theta}_{t-1} + \boldsymbol{w}_t, & \boldsymbol{w}_t & \sim & N(\boldsymbol{w}_t; 0, \boldsymbol{W}_t) \\ y_t & = & \boldsymbol{F}_t\theta_t + \boldsymbol{v}_t, & \boldsymbol{v}_t & \sim & N(\boldsymbol{v}_t; 0, \sigma_t^2) \end{array} \tag{11.23}$$

This is a linear regression model with time-varying coefficients. It is identical to the generic Kalman filter model with $\boldsymbol{G}_t = \boldsymbol{I}$. Substituting this into the update equations gives

$$\begin{array}{rcl} \hat{\boldsymbol{\theta}}_t & = & \hat{\boldsymbol{\theta}}_{t-1} + \boldsymbol{K}_t e_t \\ \boldsymbol{\Sigma}_t & = & \boldsymbol{R}_t - \boldsymbol{K}_t\boldsymbol{F}_t\boldsymbol{R}_t \end{array} \tag{11.24}$$

where

$$
\begin{aligned}
\boldsymbol{K}_t &= \frac{\boldsymbol{R}_t \boldsymbol{F}_t^T}{\sigma_{\hat{y}_t}^2} && (11.25) \\
\boldsymbol{R}_t &= \boldsymbol{\Sigma}_{t-1} + \boldsymbol{W}_t \\
\sigma_{\hat{y}_t}^2 &= \sigma_t^2 + \sigma_\theta^2 \\
\sigma_\theta^2 &= \boldsymbol{F}_t \boldsymbol{R}_t \boldsymbol{F}_t^T \\
e_t &= y_t - \hat{y}_t \\
\hat{y}_t &= \boldsymbol{F}_t \hat{\boldsymbol{\theta}}_{t-1}
\end{aligned}
$$

$$(11.26)$$

where $\hat{y}_t$ is the prediction and $\sigma_{\hat{y}_t}^2$ is the estimated prediction variance. This is composed of two terms; the observation noise, σ_t^2, and the component of prediction variance due to state uncertainty, σ_θ^2. The likelihood of a data point under the old model (or evidence) is

$$p(y_t) = N\left(y_t; \hat{y}_t, \sigma_{\hat{y}_t}^2\right) \tag{11.27}$$

If we make the further assumption that the transformation vector (its no longer a matrix because we have univariate predictions) is equal to $\boldsymbol{F}_t = -[y_{t-1}, y_{t-2}, ..., y_{t-p}]$ then we have a Dynamic Autoregressive (DAR) model.

To apply the model we make initial guesses for the state (AR parameters) mean and covariance ($\hat{\boldsymbol{\theta}}_0$ and $\boldsymbol{\Sigma}_0$) and use the above equations. We must also plug in guesses for the state noise covariance, $\boldsymbol{W}_t$, and the observation noise variance, σ_t^2. In a later section we show how these can be estimated on-line. It is also often assumed that the state noise covariance matrix is the isotropic matrix, $\boldsymbol{W}_t = q\boldsymbol{I}$. Next, we look at a set of assumptions that reduce the Kalman filter to Recursive Least Squares.

11.1.5 Recursive least squares

If there is no state noise ($\boldsymbol{w}_t = 0$,$\boldsymbol{W}_t = 0$) and no state flow ($\boldsymbol{G}_t = \boldsymbol{I}$) then the linear dynamical system in equation (1) reduces to a static linear system ($\boldsymbol{\theta}_t = \boldsymbol{\theta}$). If we further assume that our observations are univariate we can re-write the state-space equations as

$$y_t = \boldsymbol{F}_t \theta + \boldsymbol{v}_t, \qquad \boldsymbol{v}_t \sim N(\boldsymbol{v}_t; 0, \sigma_t^2) \tag{11.28}$$

This is a regression model with constant coefficients. We can, however, estimate these coefficients in a recursive manner by substituting our assumptions about $\boldsymbol{W}_t$, $\boldsymbol{G}_t$ and $\boldsymbol{V}_t$ into the Kalman filter update equations. This gives

$$
\begin{aligned}
\hat{\boldsymbol{\theta}}_t &= \hat{\boldsymbol{\theta}}_{t-1} + \boldsymbol{K}_t e_t \qquad (11.29)\\
\boldsymbol{\Sigma}_t &= \boldsymbol{\Sigma}_{t-1} - \boldsymbol{K}_t \boldsymbol{F}_t \boldsymbol{\Sigma}_{t-1}
\end{aligned}
$$

$$(11.30)$$

where

$$
\begin{aligned}
\boldsymbol{K}_t &= \frac{\boldsymbol{\Sigma}_{t-1}\boldsymbol{F}_t^T}{\sigma_{\hat{y}_t}^2} \qquad (11.31)\\
\sigma_{\hat{y}_t}^2 &= \sigma_t^2 + \sigma_\theta^2\\
\sigma_\theta^2 &= \boldsymbol{F}_t \boldsymbol{\Sigma}_{t-1} \boldsymbol{F}_t^T\\
e_t &= y_t - \hat{y}_t\\
\hat{y}_t &= \boldsymbol{F}_t \hat{\boldsymbol{\theta}}_{t-1}
\end{aligned}
$$

$$(11.32)$$

where $\hat{y}_t$ is the prediction and $\sigma_{\hat{y}_t}^2$ is the estimated prediction variance. This is composed of two terms; the observation noise, σ_t^2, and the component of prediction variance due to state uncertainty, σ_θ^2.

The above equations are identical to the update equations for recursive least squares (RLS) as defined by Abraham and Ledolter (equation (8.60) in [1]).

The likelihood of a data point under the old model (or evidence) is

$$p(y_t) = N\left(y_t; \hat{y}_t, \sigma_{\hat{y}_t}^2\right) \qquad (11.33)$$

If we make the further assumption that the transformation vector (its no longer a matrix because we have univariate predictions) is equal to $\boldsymbol{F}_t = -[y_{t-1}, y_{t-2}, ..., y_{t-p}]$ then we have a recursive least squares estimation procedure for an autoregressive (AR) model.

To apply the model we make initial guesses for the state (AR parameters) mean and covariance ($\hat{\boldsymbol{\theta}}_0$ and $\boldsymbol{\Sigma}_0$) and use the above equations. We must also plug in our guess for the observation noise variance, σ_t^2. In a later section we show how this can be estimated on-line.

11.1.6 Estimation of noise parameters

To use the DLM update equations it is necessary to make guesses for the state noise covariance, $\boldsymbol{W}_t$, and the observation noise variance, σ_t^2. In this section we show how these can be estimated on-line. Note, we either estimate the state noise or the observation noise - not both.

Jazwinski's method for estimating state noise

This method, reviewed in [14] is ultimately due to Jazwinski [28] who derives the following equations using the MLII approach (see Bayes lecture). We assume that the state noise covariance matrix is the isotropic matrix, $\boldsymbol{W} = q\boldsymbol{I}$. The parameter q can be updated according to

$$q = h\left(\frac{e^2 - \sigma_{q0}^2}{\boldsymbol{F}_t \boldsymbol{F}_t^T}\right) \tag{11.34}$$

where $h(x)$ is the 'ramp' function

$$h(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{11.35}$$

and σ_{q0}^2 is the estimated prediction variance assuming that $q = 0$

$$\sigma_{q0}^2 = \sigma_t^2 + \boldsymbol{F}_t \boldsymbol{\Sigma}_{t-1} \boldsymbol{F}_t^T \tag{11.36}$$

Thus, if our estimate of prediction error assuming no state noise is smaller than our observed error (e^2) we should infer that the state noise is non-zero. This will happen when we transit from one stationary regime to another; our estimate of q will increase. This, in turn, will increase the learning rate (see later section). A smoothed estimate is

$$q_t = \alpha q_{t-1} + (1-\alpha) h\left(\frac{e^2 - \sigma_{q0}^2}{\boldsymbol{F}_t \boldsymbol{F}_t^T}\right) \tag{11.37}$$

where α is a smoothing parameter. Alternatively, equation 11.34 can be applied to a window of samples [14].

Jazwinski's method for estimating observation noise

This method, reviewed in [14] is ultimately due to Jazwinski [28] who derives the following equations by applying the MLII framework (see Bayes lecture). Equation 11.26 shows that the estimated prediction variance is composed of two components; the observation noise and the component due to state uncertainty. Thus, to estimate the observation noise one needs to subtract the second component from the measured squared error

$$\sigma_t^2 = h\left(e_t^2 - \boldsymbol{F}_t \boldsymbol{R}_{t-1} \boldsymbol{F}_t^T\right) \tag{11.38}$$

This estimate can be derived by setting σ_t^2 so as to maximise the evidence (likelihood) of a new data point (equation 11.27). A smoothed estimate is

$$\sigma_t^2 = \alpha\sigma_{t-1}^2 + (1-\alpha)h\left(e_t^2 - \boldsymbol{F}_t\boldsymbol{R}_{t-1}\boldsymbol{F}_t^T\right) \tag{11.39}$$

where α is a smoothing parameter. Alternatively, equation 11.38 can be applied to a window of samples [14].

For RLS these update equations can be used by substituting $\boldsymbol{R}_t = \Sigma_{t-1}$. We stress, however, that this estimate is especially unsuitable for RLS applied to non-stationarity data (but then you should only use RLS for stationary data, anyway). This is because the learning rate becomes dramatically decreased.

We also stress that Jazwinski's methods cannot both be applied at the same time; the 'extra' prediction error is explained *either* as greater observation noise *or* as greater state noise.

Skagens' method

Skagen [57] lets $W = \rho\sigma_t^2\boldsymbol{I}$ ie. assumes the state noise covariance is isotropic with a variance that is proportional to the observation noise σ_t^2.

He observes that if ρ is kept fixed then varying σ_t^2 over six orders of magnitude has little or no effect on the Kalman filter updates. He therefore sets σ_t^2 to an arbitrary value eg. 1.

He then defines a measure R as the relative reduction in prediction error due to adaption and chooses ρ to give a value of $R = 0.5$.

11.1.7 Comparison with steepest descent

For a linear predictor, the learning rule for 'on-line' steepest descent is [3]

$$\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_{t-1} + \alpha\boldsymbol{F}_t^T e_t \tag{11.40}$$

where α is the learning rate, which is fixed and chosen arbitrarily beforehand. This method is otherwise known as Least Mean Squares (LMS). Haykin [27] (page 362) discusses the conditions on α which lead to a convergent learning process. Comparison of the above rule with the DLM learning rule in equation 11.25 shows that DLM has a learning rate *matrix* equal to

$$\boldsymbol{\alpha} = \frac{\boldsymbol{\Sigma}_{t-1} + q_t\boldsymbol{I}}{\sigma_t^2 + \sigma_\theta^2} \tag{11.41}$$

The average learning rate, averaged over all state variables, is given by

$$\alpha_{DLM} = \frac{1}{p}\frac{Tr\left(\mathbf{\Sigma}_{t-1} + q_t\boldsymbol{I}\right)}{(\sigma_t^2 + \sigma_\theta^2)} \tag{11.42}$$

where $Tr()$ denotes the trace of the covariance matrix and p is the number of state variables.

DLM thus uses a learning rate which is directly proportional to the variance of the state variables and is inversely proportional to the estimated prediction variance.

If the prediction variance due to state uncertainty is significantly smaller than the prediction variance due to state noise ($\sigma_\theta^2 \ll \sigma_t^2$), as it will be once the filter has reached a steady solution, then increasing the state noise parameter, q_t, will increase the learning rate. This is the mechanism by which DLM increases its learning rate when a new dynamic regime is encountered.

The average learning rate for the RLS filter is

$$\alpha_{RLS} = \frac{1}{p}\frac{Tr\left(\mathbf{\Sigma}_{t-1}\right)}{(\sigma_t^2 + \sigma_\theta^2)} \tag{11.43}$$

As there is no state noise ($q_t = 0$) there is no mechanism by which the learning rate can be increased when a new dynamic regime is encountered. This underlines the fact that RLS is a stationary model. In fact, RLS behaves particularly poorly when given non-stationary data. When a new dynamic regime is encountered, σ_θ^2 will increase (and so may σ_t^2 if we're updating it online). This leads not to the desired increase in learning rate, but to a decrease.

For stationary data, however, the RLS model behaves well. As the model encounters more data the parameter covariance matrix decreases which in turn leads to a decrease in learning rate. In on-line gradient descent learning it is desirable to start with a high learning rate (to achieve faster convergence) but end with a low learning rate (to prevent oscillation). RLS exhibits the desirable property of adapting its learning rate in exactly this manner. DLM also exhibits this property when given stationary data, but when given non-stationary data, has the added property of being able to increasing its learning rate when necessary.

We conclude this section by noting that DLM and RLS may be viewed as linear on-line gradient descent estimators with *variable* learning rates; RLS for stationary data and DLM for non-stationary data.

11.1.8 Other algorithms

The Least Mean Squares (LMS) algorithm [27] (Chapter 9) is identical to the steepest-descent method (as described in this paper) - both methods have constant learning rates.

Our comments on the RLS algorithm are relevant to RLS as defined by Abraham and Ledolter [1]. There are, however, a number of variants of RLS. Haykin [27] (page 564) defines an exponentially weighted RLS algorithm, where past samples are given exponentially less attention than more recent samples. This gives rise to a *limited* tracking ability (see chapter 16 in [27]). The tracking ability can be further improved by adding state noise (Extended RLS-1 [27], page 726) or a non-constant state transition matrix (Extended RLS-2 [27], page 727). The Extended RLS-1 algorithm is therefore similar to the DAR model described in this paper.

11.1.9 An example

This example demonstrates the basic functioning of the dynamic AR model and compares it to RLS.

A time series was generated consisting of a 10Hz sine wave in the first second, a 20Hz sinewave in the second second and a 30Hz sine wave in the third second. All signals contained additive Gaussian noise with standard deviation 0.1. One hundred samples were generated per second.

A DAR model with $p = 8$ AR coefficients was trained on the data. The algorithm was given a fixed value of observation noise ($\sigma_t^2 = 0.2$). The state noise was initially set to zero and was adapted using Jazwinski's algorithm described in equation 11.34, using a smoothing value of $\alpha = 0.1$. The model was initialised using linear regression; the first p data points were regressed onto the $p + 1$th data point using an SVD implementation of least squares, resulting in the linear regression weight vector $\boldsymbol{w}_{LR}$. The state at time step $t = p + 1$ was initialised to this weight vector; $\boldsymbol{\theta}_{p+1} = \boldsymbol{w}_{LR}$. The initial state covariance matrix was set to the linear regression covariance matrix, $\Sigma_{p+1} = \sigma_t^2 \boldsymbol{F}_{p+1} \boldsymbol{F}_{p+1}^T$. Model parameters before time $p + 1$ were set to zero.

An RLS model (with $p = 8$ AR coefficients) was also trained on the data. The algorithm was given a fixed value of observation noise ($\sigma_t^2 = 0.2$). The model was initilised by setting $\boldsymbol{\theta}_{p+1} = \boldsymbol{w}_{LR}$ and $\Sigma_{p+1} = \boldsymbol{I}$ (setting $\Sigma_{p+1} = \sigma_t^2 \boldsymbol{F}_{p+1} \boldsymbol{F}_{p+1}^T$ resulted in an initial learning rate that was'nt sufficiently large for the model to adapt to the data - see later).

Figure 11.2 shows the original time series and the evidence of each point in the time series under the DAR model. Data points occuring at the transitions between different dynamic regimes have low evidence.

Figure 11.3 shows that the state noise parameter, q, increases by an amount necessary for the estimated prediction error to equal the actual prediction error. The state noise is high at transitions between different dynamic regimes. Within each dynamic regime the state noise is zero.

Figure 11.4 shows that the variance of state variables reduces as the model is exposed to more data from the same stationary regime. When a new stationary regime is encountered the state variance increases (because q increases).

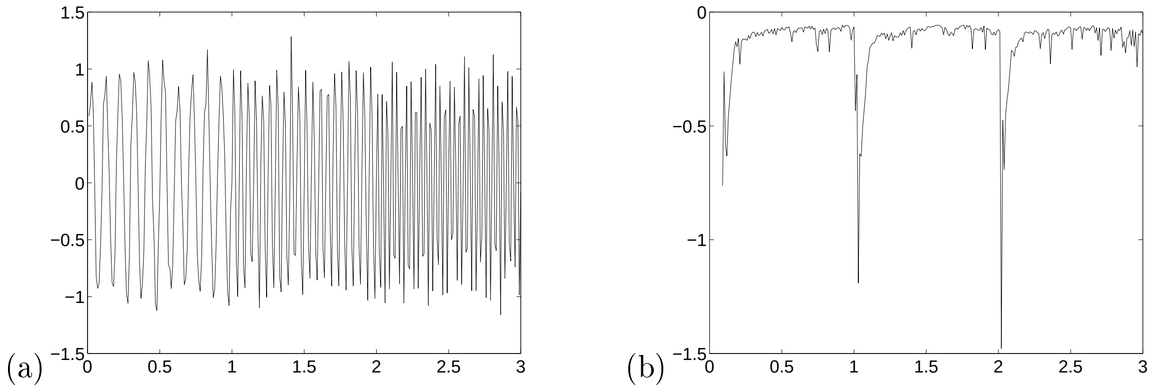

Figure 11.2: (a) Original time series (b) Log evidence of data points under DAR model, $\log p(y_t)$.

Figure 11.5 shows that the learning rate of the DAR model increases when the system enters a new stationary regime, whereas the learning rate of RLS actually decreases. The RLS learning rate is initially higher because the state covariance matrix was initialised differently (initialising it in the same way gave much poorer RLS spectral estimates).

Figure 11.6 shows the spectral estimates obtained from the DAR and RLS models. The learning rate plots and spectrogram plots show that DAR is suitable for non-stationary data whereas RLS is not.

11.1.10 Discussion

Dynamic Linear Models, Recursive Least Squares and Steepest-Descent Learning. are special cases of linear dynamical systems and their learning rules are special cases of the Kalman filter. Steepest-Descent Learning is suitable for modelling stationary data. It uses a learning rate parameter which needs to be high at the beginning of learning (to ensure fast learning) but low at the end of learning (to prevent oscillations). The learning rate parameter is usually hand-tuned to fulfill these criteria. Recursive Least Squares is also suitable for modelling stationary data. It has the advantage of having an adaptive learning rate that reduces gradually as learning proceeds. It reduces in response to a reduction in the uncertainty (covariance) of the model parameters. Dynamic Linear Models are suitable for stationary and non-stationary enviroments. The models possess state-noise and observation noise parameters which can be updated on-line so as to maximise the evidence of the observations.

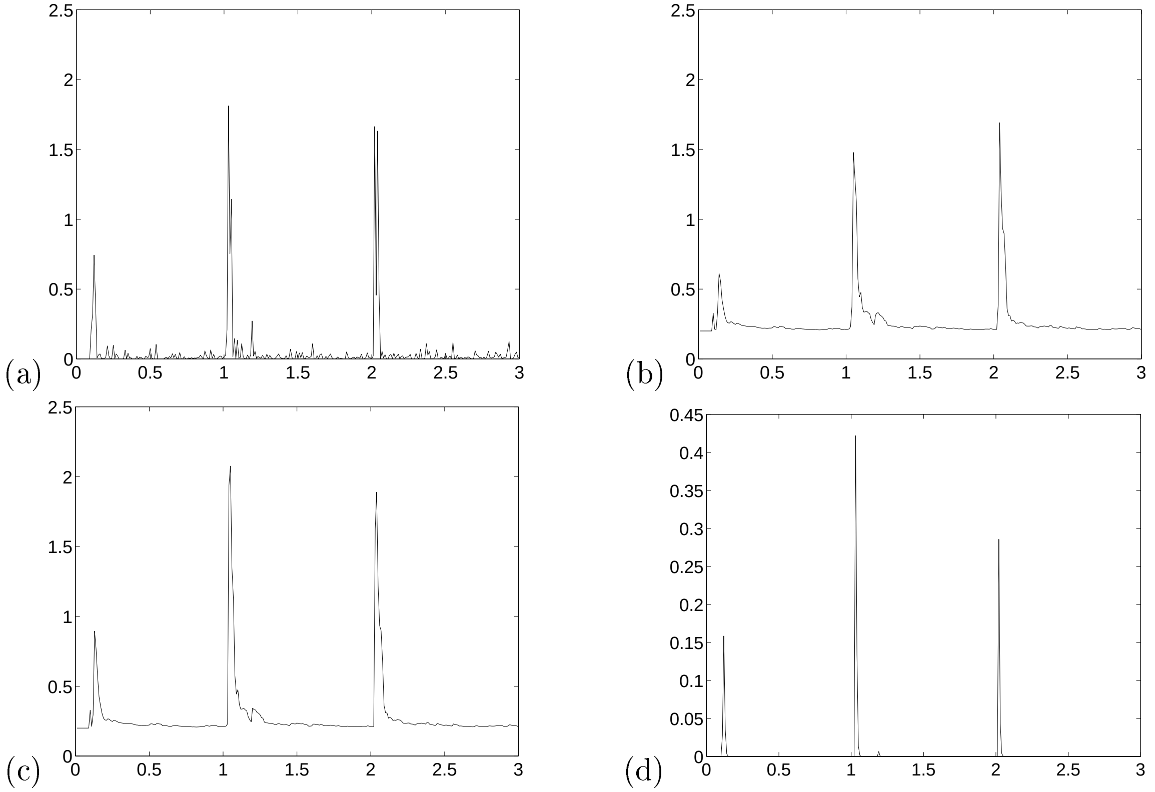

Figure 11.3: (a) Squared prediction error, e_t^2, (b) Estimated prediction error with $q_t = 0$, σ_{q0}^2, (c) Estimated prediction error, $\sigma_{\hat{y}_t}^2$ (the baseline level is due to the fixed observation noise component, $\sigma_t^2 = 0.2$) and (d) Estimate of state noise variance, q_t. The state noise, q_t, increases by an amount necessary for the estimated prediction error (plot c) to equal the actual predicition error (plot a) - see equation 11.34.

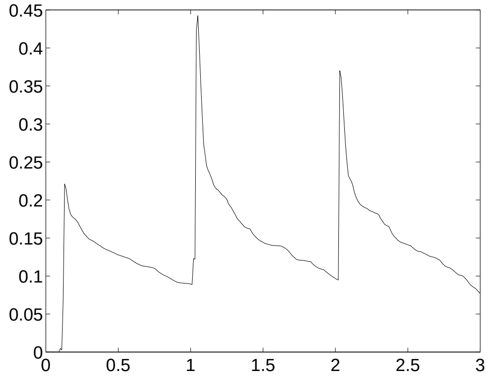

Figure 11.4: Average prior variance of state variables, $\frac{1}{p}Tr(R_t)$. As the model is exposed to more data from the same stationary regime the estimates of the state variables become more accurate (less variance). When a new stationary regime is encountered the state variance increases (because q increases).

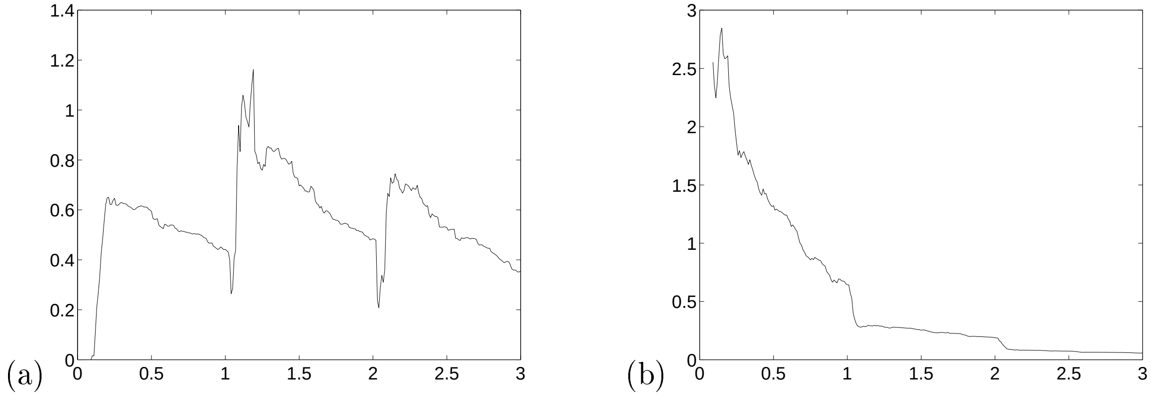

Figure 11.5: Average learning rates for (a) DAR model (b) RLS model. The learning rate for RLS is set to a higher initial value (indirectly by setting $\mathbf{\Sigma}$ to have larger entries) to give it a better chance of tracking the data. The DAR model responds to a new dynamic regime by increasing the learning rate. The RLS responds by decreasing the learning rate and is therefore unable to track the nonstationarity.

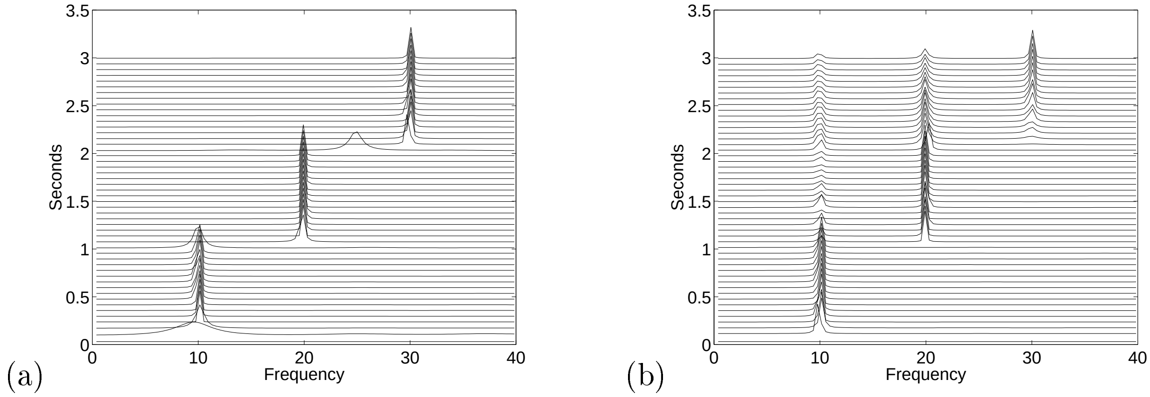

Figure 11.6: Spectrograms for (a) DAR model (b) RLS model.