Description and Implementation of Parameter Estimation for a Switching State-Space Model of Connected Worlds

Nick Hoernle

November 21, 2017

1 Introduction to Hidden Markov Models

A time-series is data that is obtained sequentially. Often the data have equal intervals between samples, and I am assuming this is true for this introduction. It is a fair assumption to make as the data can be re-sampled to present a uniformaly sampled dataset. When the time series is represented by an output vector, we let $\bar{\mathbf{y}}_t$ denote the vector of sampled values at time t.

A Hidden Markov Model (HMM) presents a framework for representing the joint probability distribution over a sequential collection of hidden and observed discrete random variables ($\bar{\mathbf{X}}$ and $\bar{\mathbf{y}}$ respectively) [1]. The above equal sampling assumption allows us to denote t an integer-valued time index that specifies one observation output. Note that a **Hidden Markov Model (HMM)** refers to the case where the states and output variables are assumed to be discrete, whereas a **State-Space Model (SSM)** refers to the case where the states and associated outputs are continuous valued. The update steps and math intuition behind both of these cases is analogous and I will refer to the models exchangeably throughout the equation derviation. I however, will refer strictly to the SSM when referring to continuous valued states and observations and the HMM when referring to the discrete valued switching variable that is preset in the switching state-space model. Is is worth introducing at this stage, the power of the switching SSM is such that it combines the continuous nature of the SSM with the discrete nature of the HMM, allowing effecting non-linear modelling from linear models.

The response vector is from an auto-regressive system that contains variables that are dependent upon one another and are dependent on their histories. An initial and general implementation can model this system as a Markov chain where the water level at time t is independent from all water levels $1 \dots t-2$ given the value of the water at time $t-1$. In the case of Connected Worlds, this is a reasonable assumption as the water flows can only depend on the previous levels of water in the system and on the user actions that dictate how the water should move from one Biome to another. Note that I have specifically chosen to model the system as a Hidden Markov Model (HMM) as this will all for a more general modelling approach when we choose to include more of the inter-dependent response variables (plants and animals).

For this discussion I have followed the notation and derivation of the fitering equatons presented by Shumway and Stoffer [2]. Following the notation for a first order HMM, we have the following state representation for the system:

$$\begin{aligned} \mathbf{X_t} &= \Phi \mathbf{X_{t-1}} + w_t \\ \mathbf{y_t} &= A\mathbf{X_t} + v_t \end{aligned} \tag{1}$$

where:

- $\mathbf{X_t}$ denotes the state vector at time t.
- $\mathbf{y_t}$ denotes the observed output vector at time t.
- A is the observation matrix of the state space model and denotes the linear transform of the state vector $\mathbf{X_t}$ to the observed vector $\mathbf{y_t}$.
- w_t is independently sampled random noise between state transitions, for the linear Gaussian state-space model, $w_t \sim \mathcal{N}(0, \mathcal{Q})$.
- ν_t is independently sampled random noise between state transitions, for the linear Gaussian state-space model, $\nu_t \sim \mathcal{N}(0, \mathcal{R})$.
- Φ denotes the transition matrix that governs the dynamics of the system from the state at time t to the state at time $t+1$.

It is useful to denote

$$x_t^s = E(x_t | y_{1:s}) \tag{2}$$

$$P_{t_1,t_2}^s = E[(x_{t_1} - x_{t_1}^s)(x_{t_2} - x_{t_2}^s)^T) \tag{3}$$

Equation 2 gives the expected value for the state at time t that depends on the observations $y_1, \ldots, y_s$. Equation 3 gives the covariance matrix of two observations at different times. P_t^s is used to denote the covariance of the data for $t_1 = t_2 = t$.

The structure of the state-space model presents an efficient conditional factorization of the joint probability distribution. The structure of the state-space model can be represented graphically, as shown in Figure 1

The problem of inference or state estimation for a state-space model with known parameters consists of estimating the posterior probabilities of the hidden variables given a sequence of observed values. The state-space inference problem can be broken into *filtering*, *smoothing* and *prediction* [3]. The goal of filtering is using all the data up to time t to calculate the probability of the hidden state X_t. Smoothing, aims to use all of the data available from time $1 \ldots T$ (with $T > t$) to calculate the probability of X_t. Lastly, prediction is calculating the probability of the future states X_{t+1} given all the data $1 \ldots t$ [4]. We are not concerned with prediction for this implementation.

Modelling the data in this framework also puts the data in a generative setting where the data can be generated by some Markov chain with known properties. Suppose with $Phi, \mu_0, \mathcal{Q}, \mathcal{R}$ given, we follow equation 1 to generate data for $t = 1 \ldots T$. We obtain generated data that is depected in figure 2.

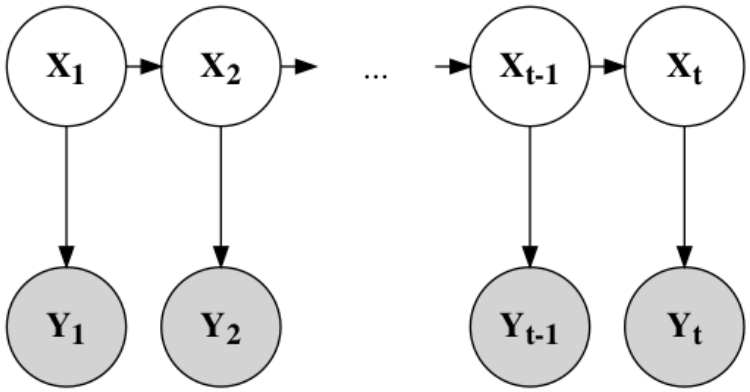

Figure 1: A state-space model is a directed acyclic graph (DAG) where the observations and states are structured such that there exists a conditional independence between any observation at time t and the rest of the graph given the state of the system at time t. There further exists a conditional independence between any state and all previous states, given the parent state of the system.

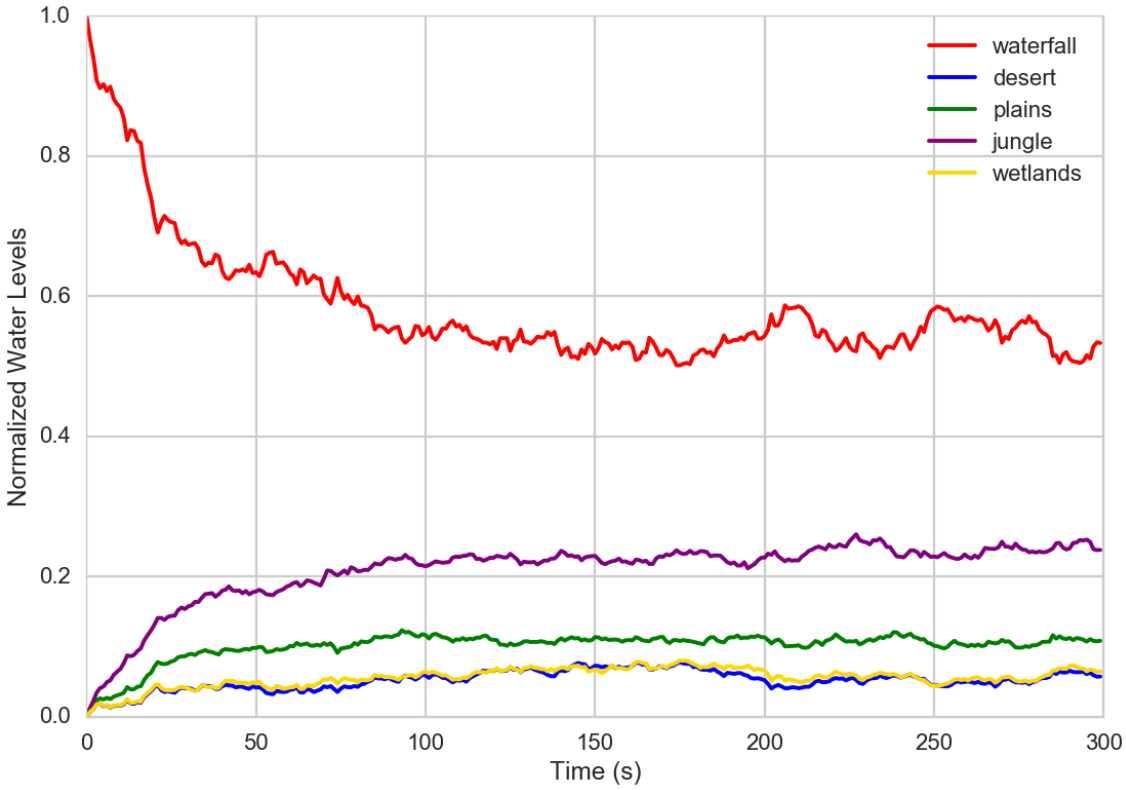

Figure 2: Data can be seen to be a simple random walk with the 5 co-dependent water levels.

2 Description of the Baum-Welsh (Expectation Maximization) algorithm

The problem of learning the parameters of a state-space model can be done via a maximum likelihood approach, in which a single value for the parameters is estimated. Note that that the fully Bayesian approach treats the parameters as random variables themselves and either computes or approximates the posterior distribution of the parameters given the data [3]. An implementation of the maximum likelihood method for extimating the parameters of the state-space model can be found in this notebook.

Expectation Maximization (EM) presents an alternative method for off-line learning of the parameters involved with the state transition and the noise governing the transitions and the observations in the system. We rather follow an Expectation Maximisation approach to solve for the state transition matrix given the observation data $\mathbf{y}$ and the hidden state data $\mathbf{X}$. The implementation of this code can be found in this notebook.

Given the parameters in Φ we are able to calculate the Complete Data Likelihood from $\{\mathbf{X_{0:n}}, \mathbf{y_{1:n}}\}$:

$$p_\Theta(\mathbf{X_{0:n}}, \mathbf{y_{1:n}}) = p_{\mu_0,\Sigma_0}(\mathbf{X_0}) \prod_{t=1}^{n} p_{\Phi,\mathbf{Q}}(\mathbf{X_t}|\mathbf{X_{t-1}}) p_R(\mathbf{y_t}|\mathbf{X_t}) \tag{4}$$

Using the Gaussian assumptions that are given above, and ignoring constants, we have the complete data negative log-likelihood:

$$\begin{aligned} -2L_{X,Y}(\Theta) = & \, ln(|\Sigma_0|) + (X_0 - \mu_0)^T \Sigma_0^{-1} (X_0 - \mu_0) \\ & + nln(|Q|) + \sum_{t=1}^{n} (X_t - \Phi X_{t-1})^T Q^{-1} (X_t - \Phi X_{t-1}) \\ & + nln(|R|) + \sum_{t=1}^{n} (y_t - AX_t)^T R^{-1} (y_t - AX_t) \end{aligned} \tag{5}$$

2.1 Baum-Welsh Implementation

Algorithm 1 Baum-Welsh Expectation Maximization

Initialize: $\Theta = \{\mu_0, \Sigma_0, \Phi, \mathcal{Q}, \mathcal{R}\}$
while $-\ln_y(\Theta^{(j)})$ not converged **do**
  Perform E step using Θ^{j-1} to calculate outputs from the mean and variance-covariance state estimates which are the output from the Kalman smoother X_t^n, P_t^n and $P_{t,t-1}^n$.
  Perform the M step to update $\Theta = \{\mu_0, \Sigma_0, \Phi, \mathcal{Q}, \mathcal{R}\}$ using the MLE from the complete-data likelihood. Compute the incomplete-data likelihood.
end while

Algorithm 1 can be implemented on the generated data and the parameters for Φ, the transition matrix and $\mathcal{Q}$ the state transition noise can be approximated from the MLE approach. Note that the expectation step, to calculate the mean and variance state estimates are provided in Appendix A. The result is shown in figure 3. It is worth noting that while the actual values in the $\hat{\Phi}$ matrix differ quite substantially from the input values

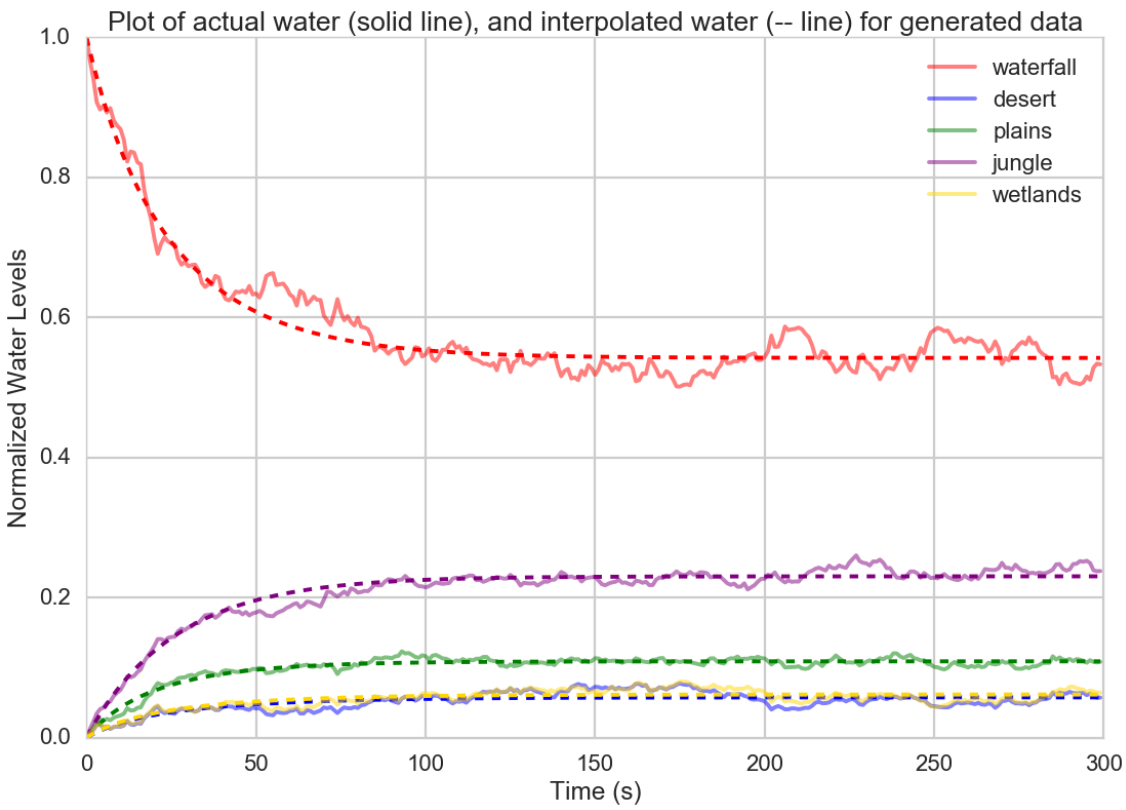

Figure 3: The transition matrix for the Markov chain can be learnt from the data and can be seen to approximate the system dynamics well.

Φ, the eigen vector that is associated with the largest eigen value of the estimated state transition matrix is equal to the eigen vector of the original vector. This does confirm that we are recovering the transition dynamics satisfactorily.

3 Considering the Switching State-Space Model - Description of New Optimization Task

We can now turn to the harder problem where we assume there are distinct switch points in the time-series response data where the system characteristics undergo distinctly different transition dynamics. An intuitive explination for the need for this level of modeling is that students may be executing actions under a specific plan or strategy. If they update their plan to reflect a new solution to solving a given problem based on new insight into the state dynamics, this will result in a markedly different response from the output vector.

The new state update equation describes $m = 1 \dots \mathcal{M}$ independent state-space models that are each governed by their own transition dynamics. There is a switching variable that selects which chain (and hence plan) is active at any given time. This is summarised in equation 6, where there are $m = 1 \dots \mathcal{M}$ models and the S_t variable is a categorical that at each time step in the response that chooses which state-space equation governs the output response (note that the S_t variable itself follows the discrete dynamics of a hidden markov

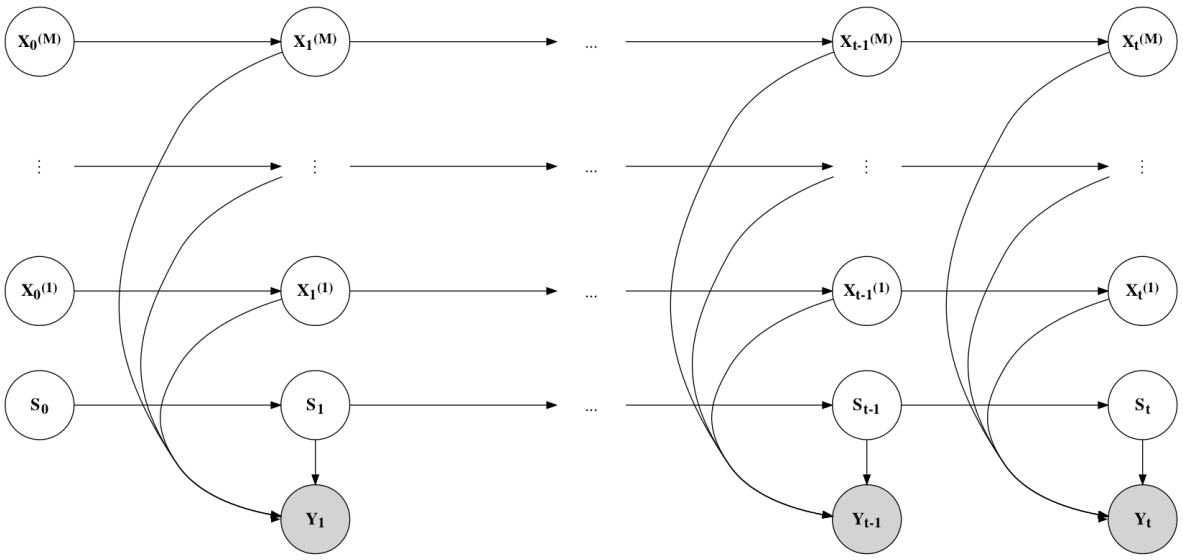

Figure 4: Graphical model for the switching-state space model described by equation 6

chain itself).

$$\begin{aligned} \mathbf{X_t^{(m)}} &= \Phi^{(m)}\mathbf{X_{t-1}^{(m)}} + w_t^{(m)} \\ \mathbf{y_t} &= S_t A^{(m)}\mathbf{X_t^{(m)}} + v_t^{(m)} \end{aligned} \tag{6}$$

The state dynamics can be summarised by the graphical model that is presented in figure 4.

Again, under a generative model, the data can be generated to follow the dynamics of three independent markov chains, and the switchpoints can be drawn from a uniform random variable over the total time duration that is present (for the Bayesian modeling to come, note that this is the same representation as a Poisson random variable with a rate parameter that is determined by the lenght of the duration and the expected number of splitpoints in the interval).

Under the assumption that the switchpoints are known, K Markov chains can be trained independently on the $K+1$ intervals of the response data. The resulting transition parameters are inferred from the data shown in figure 6.

4 Deriving Switch-Points and Markov Chain Parameters from Switching State-Space Model

Studying the 'innovations' from the forward Kalman Filter. Where:

$$\text{innovation}_t = y_t - \mathbf{A}\mathbf{X_{t-1}^t} \tag{7}$$

and $\mathbf{X_{t-1}^t}$ is derived from the forward filtering equations.

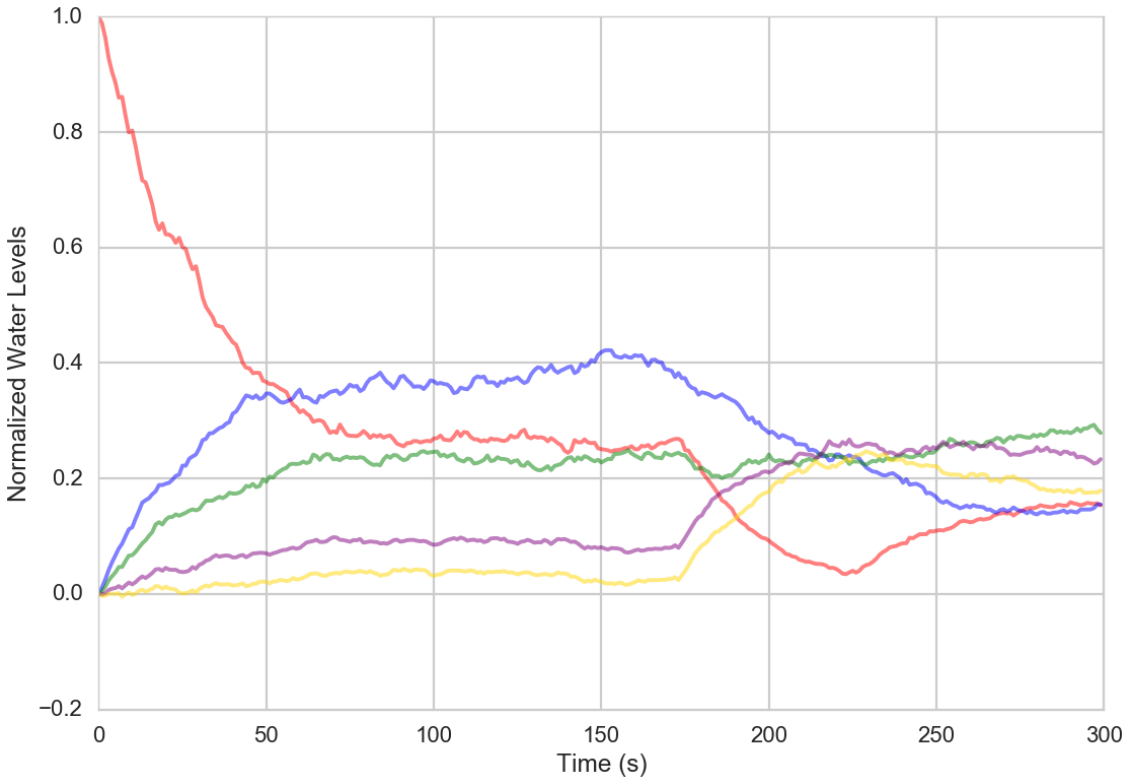

Figure 5: Data is generated from the switching state-space model that is described by figure 4. Note that the data is still a Gaussian random walk but now there are distinct switch-points where the system dynamics undergo explicit changes to the dynamics. Note that the switch times are $t = 174s$ and $t = 227s$ respectively.

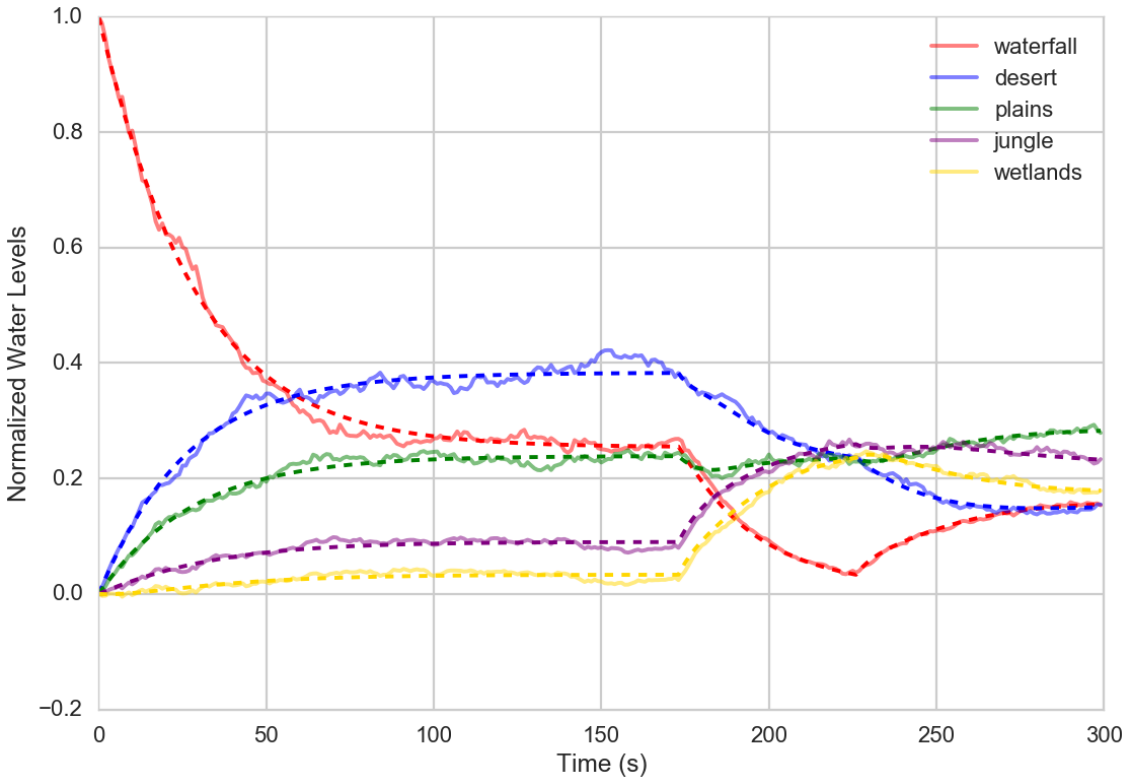

Figure 6: Inferred parameters from switching state-space model with known switch-points.

I have used the information encoded in the innovations of each chain to make an iterative update algorithm to find the best position of the switch-points for a given number of switch-points. This procedure is summarized in algorithm 2.

Algorithm 2 Search for Switch-Points

Initialize: K switch-points at equally spaced intervals on the time-series.
while Found switch-points have not converged **do**
  Use Baum-Welsh (1) to find parameters of the chains given the known position of the switch-points.
  Use the forward Kalman Filter to evaluate the innovation at time t $\forall t \in [0 \dots T]$
  Evaluate the error metric from the innovations to find more appropriate switch-points for the entire chain.
  if the found switch-points collapse to the same point **then**
    $K = K - 1$
  end if
end while

For the first iteration of this algorithm, the innovations can be evaluated to gain intuition into how the forward Kalman Filter gives an appropriate error metric for the data. The plotted innovations for three difference chains over time can be seen in figure 7. The innovations are independent Gaussian random vectors with zero means and variance-covariance matrices $\Sigma_t = A_t P_t^{t-1} A_t^T + R$. It is worth noting that the innovations of a Kalman Filter can be viewed in a linear regression sense, in that we are evaluating the expected value of $\mathbf{y_t}|\mathbf{X_t}, \mathbf{\Theta}$ and thus minimising the sum of squared errors is exactly the same as miximizing the likelihood of the given model parameters over the observed data. This is what the algorithm 2 achieves.

The final found switch-points can be seen in figure 8.

5 Demonstration on Connected Worlds Data

For this demonstration, I have used the Connected Worlds logs from 16/11/2017[1]. I first assume there are no switch-points that are present and I run the Baum-Welsh 1 algorithm on the data. I have transformed the raw connected worlds data so that:

1. the total water in the system sums to 1.
2. the water in the four biomes are shown distinctly, and the rest of the water (the wetlands, the desert, the floor and the waterfall) are shown grouped together.

[1] the session name is '12-00-37-ESSIL_October_Test'

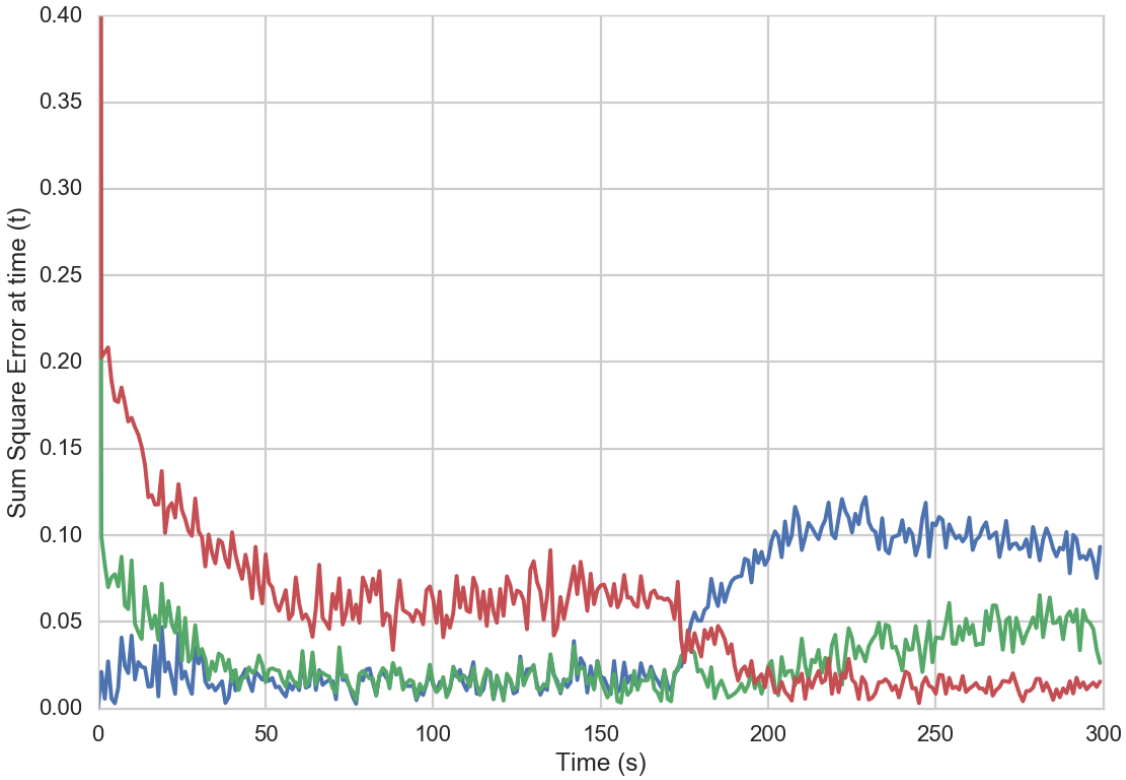

Figure 7: Plot of innovations for three chains given starting switch-points that are equally spaced on the time series.

Appendices

A Kalman Filter Update Equations

$$x_t^{t-1} = \Phi x_{t-1}^{t-1} \tag{8}$$

$$P_t^{t-1} = \Phi P_{t-1}^{t-1} \Phi^T + Q \tag{9}$$

$$x_t^t = x_t^{t-1} + K_t \epsilon_t \tag{10}$$

$$\epsilon_t = y_t - E(y_t | t_{1:t-1}) = y_t - x_t^{t-1} \tag{11}$$

$$P_t^t = [I - K_t] P_t^{t-1} \tag{12}$$

Where:

$$K_t = P_t^{t-1} [P_t^{t-1} + R]^{-1} \tag{13}$$

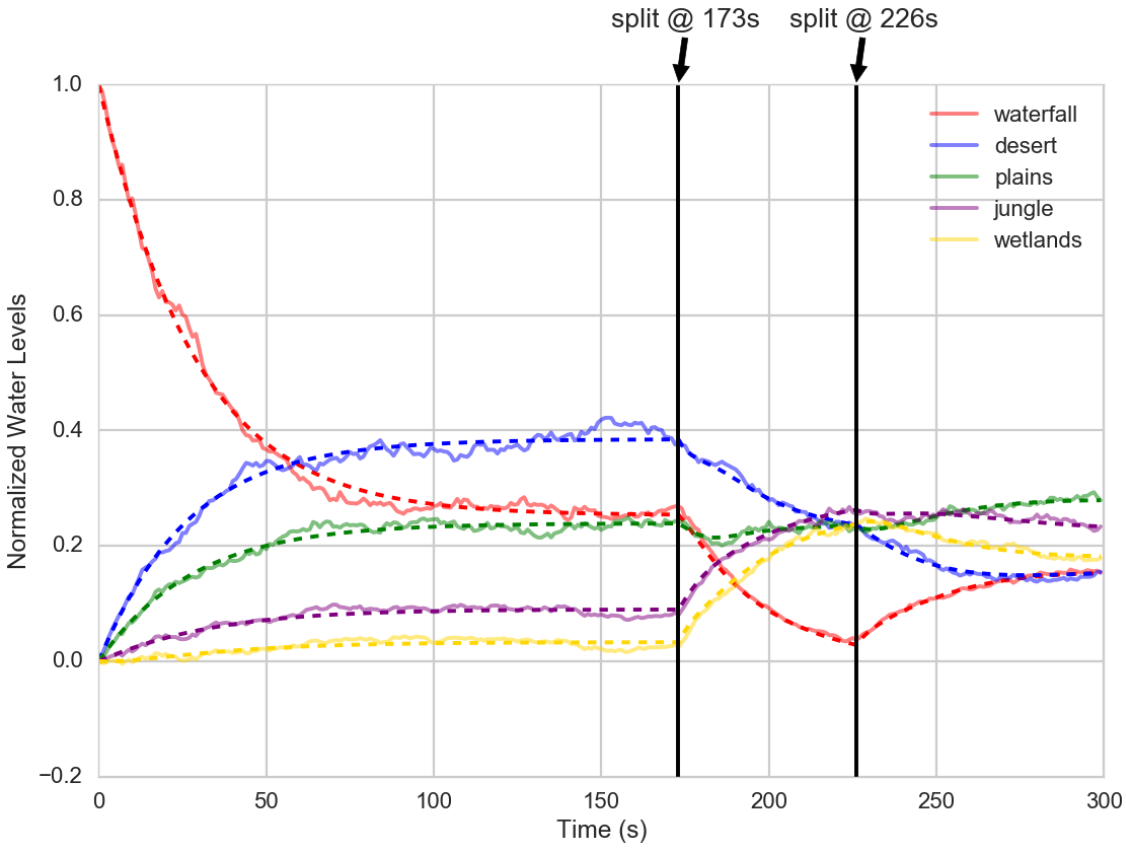

Figure 8: Learnt switchpoints and inferred SSM parameters on generated data (note that the true splitpoints were at $t = 174s$ and $t = 227s$ respectively.)

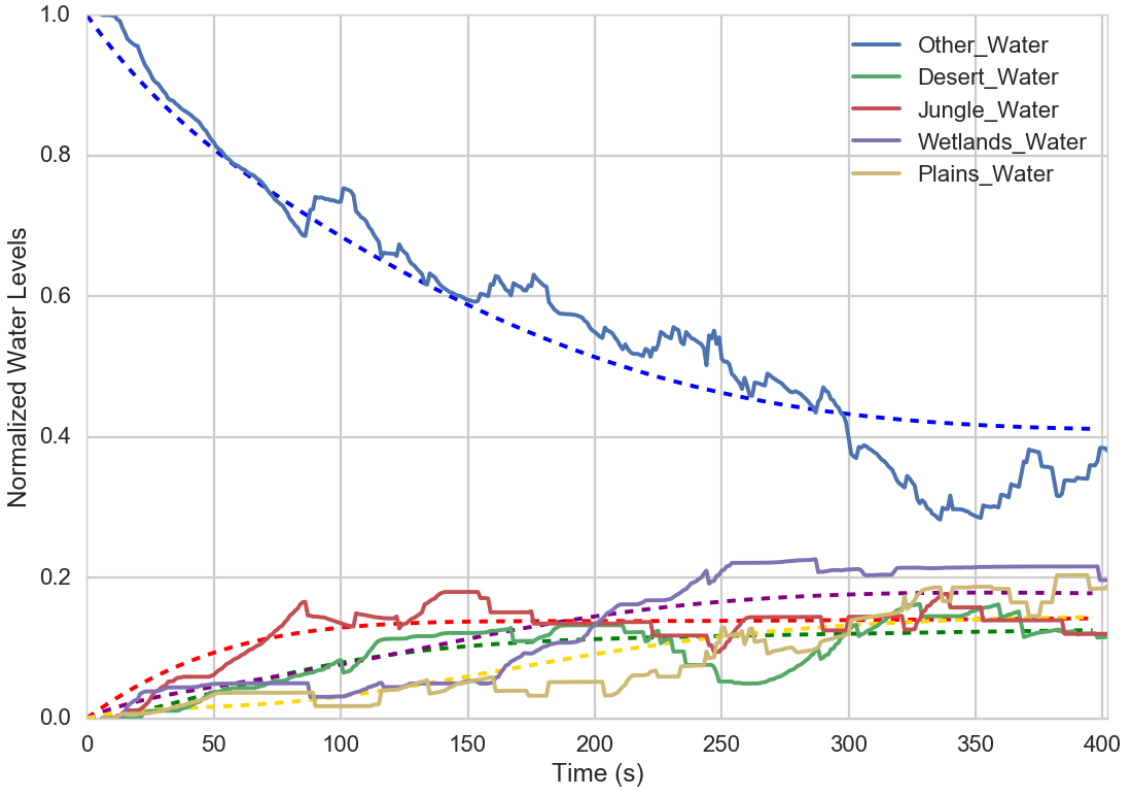

Figure 9: Inferred parameters for a Connected Worlds session with no switch-points.

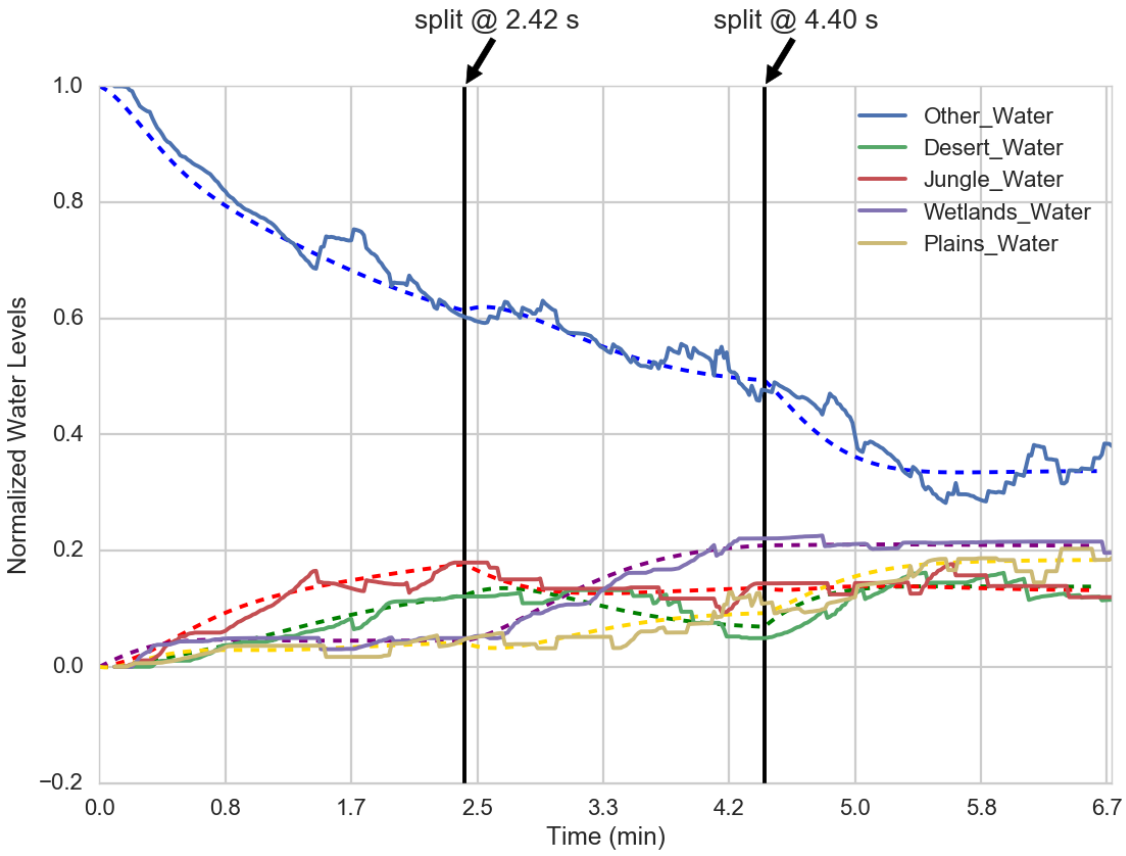

Figure 10: Inferred parameters for a Connected Worlds Session with the learnt switching-points highlighted.

References

[1] Z. Ghahramani, "An introduction to hidden markov models and bayesian networks," *International journal of pattern recognition and artificial intelligence*, vol. 15, no. 01, pp. 9–42, 2001.

[2] R. H. Shumway and D. S. Stoffer, *Time series analysis and its applications: with R examples*. Springer Science & Business Media, 2006.

[3] Z. Ghahramani and G. E. Hinton, "Variational learning for switching state-space models," *Neural computation*, vol. 12, no. 4, pp. 831–864, 2000.

[4] B. D. Anderson and J. B. Moore, "Optimal filtering," *Englewood Cliffs*, vol. 21, pp. 22–95, 1979.