Contents

Part I

Literature, theory and background material

Chapter 4

Hidden Markov models

In this chapter we consider probabilistic graphical models of the form shown in Figure 4.1. We assume that $(X_0, X_1, ...)$ are each n state discrete random variables and that $(Y_0, Y_1, ...)$ are each m state discrete random variables. Models of this form are classically called hidden Markov models.

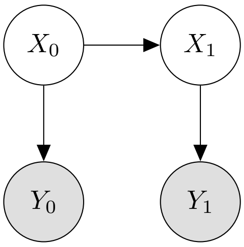

Figure 4.1: Graphical model used in this chapter.

Intuitively, this model represents the situation where we are not sure about the state of the world but we can observe some facet of it. At each time step our state changes stochastically according to the transition function. A new observation is (stochastically) generated from our new state. Generally, we attempt to infer the state of the world given the observations. Clearly two new random variables X_t, Y_t are created at each time step.

In this chapter we briefly describe Markov models because they link back to previous work done by the chemical engineering department at the University of Pretoria. We focus on hidden Markov models for the remainder of the chapter because the techniques we develop here generalise to linear latent dynamical systems which we discuss in chapter 6.

4.1 Markov models

A first order Markov model (sometimes called a Markov chain) is shown in Figure 4.2. Using the chain rule for Bayesian networks (Definition 3.19) we can immediately write down the joint probability distribution

$$P(X_{0:T}) = \Pi_{t=0}^{T} P(X_t|X_{t-1}) \text{ with } P(X_0|X_{-1}) = P(X_0). \tag{4.1}$$

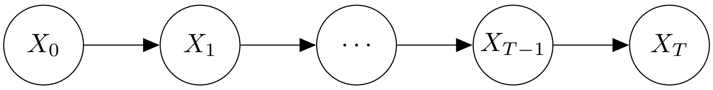

Figure 4.2: First order markov chain.

This model describes the forward propagation of a discrete random variable through time. It is interesting to study the marginal distribution of $P(X_T)$ as it evolves through time. By d-separation we know that $X_t \perp\!\!\!\perp X_{0:t-2}|X_{t-1}$. Thus, we only have to marginalise out the previous time step to compute the required distribution

$$P(X_T) = \sum_{x_{T-1}} P(X_T, x_{T-1}) = \sum_{x_{T-1}} P(x_{T-1})P(X_T|x_{T-1}). \tag{4.2}$$

Since we know that the transition function is a row stochastic $n \times n$ matrix (the random variable has n discrete states) we can write (4.2) in vector notation

$$P(X_t) = \mathbf{p}_t = \mathbf{A}\mathbf{p}_{t-1} = \mathbf{M}^{t-1}\mathbf{p}_1. \tag{4.3}$$

Note that $P(X_T)$ is a discrete random variable and can thus be expressed as a stochastic column vector i.e. $\sum_i P(x_t = i) = 1$. We have implicitly rewritten (4.3) in recursive format. Thus, we have a recursive expression for the marginal distribution of X. If, as $T \to \infty$, we have that $\mathbf{p}_{t\to\infty} = \mathbf{p}_\infty$ exists and is independent of $\mathbf{p}_0$ we call $\mathbf{p}_\infty$ the equilibrium distribution of the chain.

We define the stationary distribution, in matrix notation, by

$$\mathbf{p}_\infty = \mathbf{A}\mathbf{p}_\infty. \tag{4.4}$$

Recalling the definition of the eigenvalue problem we see that the stationary distribution is just the eigenvector corresponding to the unit eigenvalue of $\mathbf{A}$. While this model may seem simplistic it is the foundation of Google's PageRank algorithm [58]. Intuitively $\mathbf{p}_\infty$ represents the steady state probability distribution of the random variable X as it is propagated through time by the transition function A. See the work in [53] for an application specifically geared towards chemical engineering.

4.2 Hidden Markov models

Hidden Markov models extend Markov models by incorporating the observed random variables $(Y_0, Y_1, ...)$ as shown in Figure 4.1. At each time step it is now possible to observe the random variable Y_t which gives more information about the state of X_t. We are still in the setting of discrete random variables. It is not necessary to restrict $(Y_0, Y_1, ...)$ to be discrete but we do so for the sake of simplicity here. In later chapters we will model both hybrid and purely continuous systems.

In general a hidden Markov model is just a specific case of the general dynamic Bayesian network class of graphical models. As such we already know that to fully specify the model we only require a prior state distribution $P(X_0)$, the transition probability function $P(X_t|X_{t-1})$, the observation (or sometimes called the emission) probability function $P(Y_t|X_t)$ and the Bayesian network graphs of the initial time step and the next two time steps. We assume that the model's structure repeats at each time step and thus we only require the graph as shown in Figure 4.1.

We assume that the transition and observation probability functions are stationary. Consequently they may be represented by the row stochastic square matrices $P(x_t = i|x_{t-1} = j) = A$ and $P(y_t = i|x_t = j) = B$. Intuitively this means that the probability of state $x_{t-1} = j$ going to state $x_t = i$ is A_{ij}. Similarly, B_{ij} is the probability of observing $y_t = i$ if the underlying state is $x_t = j$.

For the purposes of this dissertation we will always assume that the model parameters are known. In section 3.3.2 the four primary inference techniques were briefly mentioned. We now derive recursive expressions for each inference technique for discrete models of the form shown in Figure 4.1. The tools and techniques we develop here will be useful in the following chapters.

4.2.1 Filtering

The goal of filtering is to find $P(X_t|y_{0:t})$: the distribution of the current state given all the past observations. The corresponding graphical model is shown in Figure 4.3.

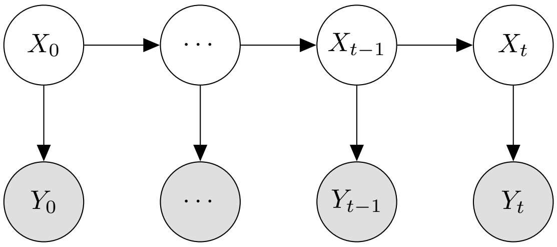

Figure 4.3: Filtering graphical model.

We start the derivation by noting that X_{t-1} d-separates X_t from $X_{0:t-2}$. Thus X_{t-1} contains all the hidden state information of the system up to and including $t-1$. This is not surprising since we have assumed a first order Markov system. This is why we only marginalise over the reduced state joint $P(X_t, X_{t-1}|y_{0:t})$ in

$$\begin{aligned} P(X_t|y_{0:t-1}) &= \sum_{x_{t-1}} P(X_t, x_{t-1}|y_{0:t-1}) \\ &= \sum_{x_{t-1}} P(x_{t-1}|y_{0:t-1}) P(X_t|y_{0:t-1}, x_{t-1}) \\ &= \sum_{x_{t-1}} P(x_{t-1}|y_{0:t-1}) P(X_t|x_{t-1}). \end{aligned} \tag{4.5}$$

Where the expansion followed from the chain rule of Bayesian networks (Definition 3.19) and the cancellation followed from the conditional independence assumption of the transition function. Now we define $\alpha(X_t) \equiv P(X_t|y_{0:t})$. Then

$$\begin{aligned} \alpha(X_t) &= \frac{P(y_t|X_t, y_{0:t-1}) P(X_t|y_{0:t-1})}{P(y_t|y_{0:t-1})} \\ &= \frac{P(y_t|X_t) P(X_t|y_{0:t-1})}{P(y_t|y_{0:t-1})} \\ &= \frac{P(y_t|X_t) \sum_{x_{t-1}} P(x_{t-1}|y_{0:t-1}) P(X_t|x_{t-1})}{P(y_t|y_{0:t-1})} \\ &= \frac{P(y_t|X_t) \sum_{x_{t-1}} \alpha(x_{t-1}) P(X_t|x_{t-1})}{P(y_t|y_{0:t-1})} \end{aligned} \tag{4.6}$$

follows from (4.5) and by application of Bayes' theorem (Theorem 3.4). It is not actually necessary to calculate $p(y_t|y_{0:t-1})$ as it is only a normalisation constant. We thus have a recursion relation for the filtered posterior distribution X_t with initial condition $\alpha(X_1) = P(X_1, y_1) = P(X_1)P(y_1|X_1)$ by

$$\alpha(X_t) \propto P(y_t|X_t) \sum_{x_{t-1}} \alpha(x_{t-1}) P(X_t|x_{t-1}). \tag{4.7}$$

One often uses logarithms to perform the filter calculations as machine precision errors become a problem for large t due to the multiplication of small fractions. The recursive filtering algorithm we derived is often called the forwards algorithm in literature [3].

4.2.2 Smoothing

The goal of smoothing is to find $P(X_t|y_{0:T})$ for $t \leq T$: the distribution of the state at t given all the past and future observations to T. The smoothing algorithm we study here is called the parallel smoothing algorithm. The recursion expression we derive is often called the backwards algorithm in literature [43]. The graphical model corresponding to this situation is shown in Figure 4.4.

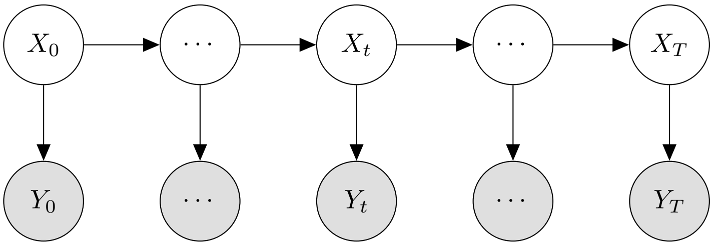

Figure 4.4: Smoothing graphical model.

We start by splitting the joint $P(X_t, y_{0:T}) = P(X_t, y_{0:t})P(y_{t+1:T}|X_t, y_{0:t})$ by the chain rule and using d-separation to reduce it further $P(X_t, y_{0:T}) = P(X_t, y_{0:t})P(y_{t+1:T}|X_t)$. Effectively the last step implies that future observations are independent of past observations given the current state. We defined $\beta(X_t) \equiv P(y_{t+1:T}|X_t)$ and continue

$$\begin{aligned} P(y_{t:T}|X_{t-1}) &= \sum_{x_t} P(y_t, y_{t+1:T}, x_t|X_{t-1}) \\ &= \sum_{x_t} P(y_{t+1:T}, x_t|X_{t-1})P(y_t|y_{t+1:T}, x_t, X_{t-1}) \\ &= \sum_{x_t} P(y_{t+1:T}, x_t|X_{t-1})P(y_t|x_t) \\ &= \sum_{x_t} P(x_t|X_{t-1})P(y_{t+1:T}|x_t, X_{t-1})P(y_t|x_t) \\ &= \sum_{x_t} P(x_t|X_{t-1})P(y_{t+1:T}|x_t)P(y_t|x_t). \end{aligned} \tag{4.8}$$

We have made judicious use of the implied independence assertions (via d-separation) of Figure 4.1. Making use of the definition of β we have

$$\begin{aligned} &\beta(X_{t-1}) = \sum_{x_t} P(x_t|X_{t-1})\beta(x_t)P(y_t|x_t) \text{ for } 1 \le t \le T \\ &\text{with } \beta(X_T) = 1. \end{aligned} \tag{4.9}$$

The recursion initial condition $\beta(X_T) = 1$ stems from Bayes' theorem (Theorem 3.4) and the definition of α as illustrated by

$$\begin{aligned} P(X_T|y_{0:T}) &= \frac{P(X_T, y_{0:T})}{P(y_{0:T})} \\ &= \alpha(X_T)\beta(X_T) \\ &= P(X_T|y_{0:T})\beta(X_T) \\ &\implies \beta(X_T) = 1. \end{aligned} \tag{4.10}$$

Note that β is not a probability function. The smoothed posterior is given by applying Bayes' theorem

$$P(X_t|y_{0:T}) = \frac{\alpha(X_t)\beta(X_t)}{\sum_{x_t} \alpha(x_t)\beta(x_t)}. \tag{4.11}$$

Together the $\alpha - \beta$ recursions are called the forwards-backwards algorithm and find extensive use in general purpose exact inference of dynamic Bayesian networks [43].

Numerical issues may also become problematic due to the multiplication of small positive numbers. In practice it is often necessary to work in the log space to attenuate these problems [3].

4.2.3 Viterbi decoding

The goal of Viterbi decoding is to find $x^*_{0:T} = \underset{x_{0:T}}{\arg\max}\, P(x_{0:T}|y_{0:T})$: finding the most likely sequence of states which best describe the observations by attempting to find the sequence

$x_{0:T}$ such that the joint probability function $P(x_{0:T}, y_{0:T})$ is maximised. This is equivalent to finding $\arg\max_{x_{0:T}} P(x_{0:T}|y_{0:T})$ because, if one uses the chain rule on the joint, the observations will just be a constant factor. The graphical model used here is similar to that of Figure 4.4.

Intuitively we first attempt to find the maximum of the joint and then determine which sequence of states led to this maximal joint. By using the chain rule for Bayesian networks we can rewrite the joint maximisation problem as

$$\begin{aligned}\max_{x_{0:T}} P(x_{0:T}, y_{0:T}) &= \max_{x_{0:T}} \Pi_{t=1}^{T} P(y_t|x_t)P(x_t|x_{t-1}) \\ &= \left(\max_{x_{0:T-1}} \Pi_{t=1}^{T-1} P(y_t|x_t)P(x_t|x_{t-1})\right) \max_{x_T} P(y_T|x_T)P(x_T|x_{T-1}).\end{aligned} \tag{4.12}$$

Defining $\mu(X_{t-1}) \equiv \max_{x_t} P(y_t|x_t)P(x_t|x_{t-1})$ we can rewrite (4.12) as

$$\max_{x_{0:T}} P(x_{0:T}, y_{0:T}) = \max_{x_{0:T-1}} \Pi_{t=1}^{T-1} P(y_t|x_t)P(x_t|x_{t-1})\mu(x_{T-1}). \tag{4.13}$$

Thus we have a recursive expression,

$$\begin{aligned}&\mu(x_{t-1}) = \max_{x_t} P(y_t|x_t)P(x_t|x_{t-1})\mu(x_t) \text{ for } 2 \le t \le T \\ &\text{with } \mu(x_T) = 1,\end{aligned} \tag{4.14}$$

to find the value of the joint under the most likely sequence of states given the observations.

The recursive expression in (4.14) implies that the effect of maximising over over the previous time step can be compressed into a message (a function) of the current time step. Effectively we pass theses messages backward in time to find the maximum joint in terms of x_0. We then find the state which maximises this joint and pass this message forward. Continuing in this way we have

$$\begin{aligned}x_1^* &= \arg\max_{x_1} P(y_1|x_1)P(x_1)\mu(x_1) \\ x_2^* &= \arg\max_{x_2} P(y_2|x_2)P(x_2|x_1^*)\mu(x_2) \\ &\vdots \\ x_t^* &= \arg\max_{x_t} P(y_t|x_t)P(x_t|x_{t-1}^*)\mu(x_t).\end{aligned} \tag{4.15}$$

This algorithm is called the Viterbi algorithm. It is computationally efficient since the optimisations occur only on a single variable. Readers familiar with dynamic programming will recognise that we have effectively performed a variant of dynamic programming in the preceding derivation.

4.2.4 Prediction

The goal of prediction is to find $P(X_{t+1}|y_{0:t})$ and $P(Y_{t+1}|y_{0:t})$: the predicted hidden and observed state given all the previous observations. The one step ahead prediction expression for both the states and observations is derived here. The graphical model corresponding to the state prediction is shown in Figure 4.5.

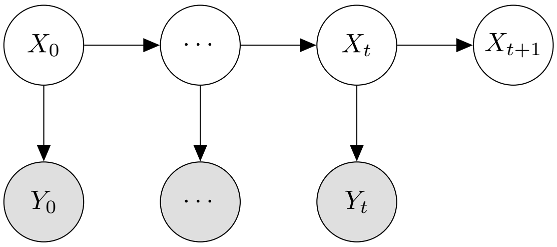

Figure 4.5: State prediction graphical model.

We again start by noticing that given all the observations up to time t the current state d-separates all previous states. Thus, to infer information about the next state we only need to marginalise out the current state X_t. Furthermore, the next state d-separates the next observation from all the previous states. Thus, to infer information about the next observation we additionally only need to marginalise out X_{t+1}.

We start with predicting the next state distribution. We have applied the chain rule, the independence assertions and used the definition of α to write

$$\begin{aligned} P(X_{t+1}|y_{0:t}) &= \sum_{x_t} P(X_{t+1}, x_t|y_{0:t}) \\ &= \sum_{x_t} P(x_t|y_{0:t}) P(X_{t+1}|x_t, y_{0:t}) \\ &= \sum_{x_t} P(x_t|y_{0:t}) P(X_{t+1}|x_t) \\ &= \sum_{x_t} \alpha(x_t) P(X_{t+1}|x_t). \end{aligned} \tag{4.16}$$

Clearly the state prediction uses the filtered state estimate and projects that forward using the transition function. Next we derive the observation prediction. Again we apply the chain rule, use the independence assertions and use the definition of α to write

$$\begin{aligned} P(Y_{t+1}|y_{0:t}) &= \sum_{x_t, x_{t+1}} P(x_{t+1}, x_t, Y_{t+1}|y_{0:t}) \\ &= \sum_{x_t, x_{t+1}} P(x_t|y_{0:t}) P(x_{t+1}, Y_{t+1}|y_{0:t}, x_t) \\ &= \sum_{x_t, x_{t+1}} P(x_t|y_{0:t}) P(x_{t+1}|y_{0:t}, x_t) P(Y_{t+1}|y_{0:t}, x_t, x_{t+1}) \\ &= \sum_{x_t, x_{t+1}} P(x_t|y_{0:t}) P(x_{t+1}|x_t) P(Y_{t+1}|x_{t+1}) \\ &= \sum_{x_t, x_{t+1}} \alpha(x_t) P(x_{t+1}|x_t) P(Y_{t+1}|x_{t+1}). \end{aligned} \tag{4.17}$$

Clearly the observation prediction is just an extension of the state prediction. We effectively just predict the next state and use the observation function to predict the observation distribution.

It is pleasing that the prediction expressions are closely related to each other and effectively only depend on the filtered state estimate and the transition or observation functions. This realisation hold for more general problems and will become important when we consider controlling the system. More on this later.

4.3 Burglar localisation problem

While the focus of this dissertation is not on hidden Markov model type problems it is nevertheless instructive to consider a simple example to build some intuition about inference of random variables. Thus, it is desirable to conduct a numerical experiment using the previously derived inference techniques. The type of problem we consider here is a localisation problem. This type of problem (and its extensions) has many applications in robotics and object tracking.

The problem is taken from Chapter 23 in Barber's book *Bayesian Reasoning and Machine Learning* [3]. Briefly, it is necessary to infer the location of a burglar in your house given observations (noises) you perceive from an adjoining room. You discretise the room the burglar is in into n^2 blocks. The room is then the discrete random variable X. You observe two distinct types of noises: creaks and bumps. From the knowledge of your house you know which blocks are likely to creak and which are likely to bump if the burglar is on that block i.e. if the random variable X is in a specific state. This is shown in Figure 4.6: a dark block indicates it is likely to emit a noise with probility 0.9 and a light block will emit a noise with probability 0.01 if the burglar is on that block. The noises are independent of each other.

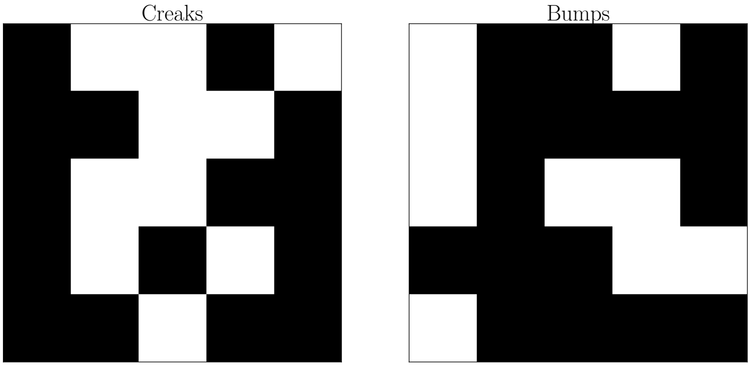

Figure 4.6: Burglar problem observations.

The burglar moves up, down, left and right with equal probability where appropriate. See Barber for more details on the example. It is necessary to perform inference to determine the path of the burglar both in real time and with the benefit of hindsight. Applying the inference techniques we developed earlier results in Figure 4.7.

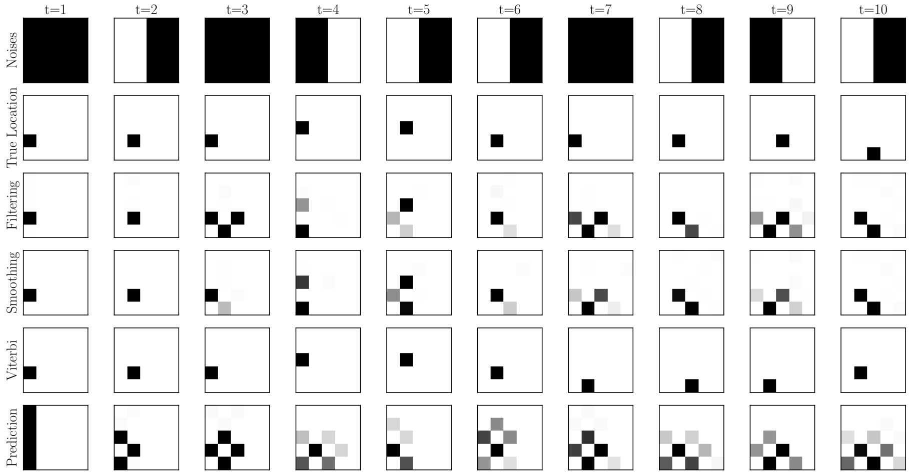

Figure 4.7: Burglar problem: filter, smoothing, Viterbi decoding and prediction.

In this context filtering means we estimate the location of the burglar given all the past observations at the current time step. This inference can be done on-line. Smoothing means we attempt to estimate his position with hindsight given all the observations starting from the first time step and moving forwards. In Viterbi decoding we attempt to estimate the most likely path path of the burglar. Finally the prediction algorithm is self-explanatory.

It is interesting to note that smoothed posterior converges to the filtered posterior near the end of the time window. Reflecting on the expression for smoothing this is not surprising since at $t = T$ the smoothing component of the forwards-backwards algorithm is unity. Therefore we see that the smoothed state estimate converges to the filtered state estimate as t approaches T.

It is also interesting to note that the prediction algorithm is very much dependent on the quality of the transition function. The four block pattern readily apparent in the prediction distribution originates from the transition function (the burglar is equally likely to move in any direction). This strongly implies that the closer the transition function is to reality the better our predictions will be.

Finally, it is important to understand the benefit of using this approach as opposed to the exhaustive “if this then that” approach. Firstly, the latter approach scales exponentially with the number of variables because one would need to fully consider all the possibilities to infer any sort of belief. Secondly, the former approach has an associated probability distribution: the certainty of our inferred belief is automatically quantified e.g. the darker the blocks the more sure we are about our inference. Thus, the techniques we developed make room for uncertainty about the correctness of the answer.

Hidden Markov models are very powerful and have found many uses e.g. speech recognition, object tracking and bio-informatics [3]. Many extensions of the basic model (see Figure 4.1) exist which are much more expressive. However, we are interested in modelling and reasoning about continuous random variables. For such applications the hidden Markov model, due to the discrete assumption, is inappropriate. Fortunately, the techniques investigated in this chapter carry over to the continuous case as we will see in Chapter 6.

Part II

Single model systems

Part III

Multiple model systems

Bibliography

[1] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation.* John Wiley and Sons, 2001.

[2] D. Barber. Expectation correction for smoothed inference in switching linear dynamical systems. *Journal of Machine Learning*, 7:2515–2540, 2006.

[3] D. Barber. *Bayesian Reasoning and Machine Learning.* Cambridge University Press, 2012.

[4] I. Batina, A.A. Stoorvogel, and S. Weiland. Optimal control of linear, stochastic systems with state and input constraints. In *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002.

[5] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatics*, 35:407–427, 1999.

[6] C.M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[7] L. Blackmore, Hui Li, and B. Williams. A probabilistic approach to optimal robust path planning with obstacles. In *American Control Conference*, June 2006.

[8] L. Blackmore, O. Masahiro, A. Bektassov, and B.C. Williams. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *IEEE Transactions on Robotics*, 26, 2010.

[9] M. Cannon, B. Kouvaritakis, and X. Wu. Probabilistic constrained mpc for multiplicative and additive stochastic uncertainty. *IEEE Transactions on Automatic Control*, 54(7), 2009.

[10] A.L. Cervantes, O.E. Agamennoni, and J.L Figueroa. A nonlinear model predictive control system based on weiner piecewise linear models. *Journal of Process Control*, 13:655–666, 2003.

[11] R. Chen and J.S. Liu. Mixture kalman filters. *Journal of Royal Statistical Society*, 62(3):493–508, 2000.

[12] J.J. Dabrowski and J.P. de Villiers. A method for classification and context based behavioural modelling of dynamical systems applied to maritime piracy. *Expert Systems with Applications*, 2014.

[13] B.N. Datta. *Numerical Methods for Linear Control Systems - Design and Analysis.* Elsevier, 2004.

[14] F. Daum and J. Huang. Particle flow for nonlinear filters. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5920–5923, May 2011.

[15] M. Davidian. Applied longitudinal data analysis. North Carolina State University, 2005.

[16] J.P. de Villiers, S.J. Godsill, and S.S. Singh. Particle predictive control. *Journal of Statistical Planning and Inference*, 141:1753–1763, 2001.

[17] N. Deo. *Graph Theory with Applications to Engineering and Computer Science.* Prentice-Hall, 1974.

[18] M. Diehl, H.J. Ferreau, and N. Haverbeke. Efficient numerical methods for nonlinear mpc and moving horizon estimation. *Control and Information Sciences*, 384:391–417, 2009.

[19] A. Doucet and A.M. Johansen. A tutorial on particle filtering and smoothing: fifteen years later. Technical report, The Institute of Statistical Mathematics, 2008.

[20] A.D. Doucet, N.J. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624, March 2001.

[21] J. Du, C. Song, and P. Li. Modeling and control of a continuous stirred tank reactor based on a mixed logical dynamical model. *Chinese Journal of Chemical Engineering*, 15(4):533–538, 2007.

[22] The Economist. In praise of bayes. Article in Magazine, September 2000.

[23] C. Edwards, S.K. Spurgeon, and R.J. Patton. Sliding mode observers for fault detection and isolation. *Automatica*, 36:541–553, 200.

[24] H.C. Edwards and D.E. Penny. *Elementary Differential Equations.* Pearson, 6th edition edition, 2009.

[25] W. Forst and D. Hoffmann. *Optimisation - Theory and Practice.* Springer, 2010.

[26] O.R. Gonzalez and A.G. Kelkar. *Electrical Engineering Handbook.* Academic Press, 2005.

[27] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113, 1993.

[28] R. Isermann and P. Balle. Trends in the application of model based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5):709–719, 1997.

[29] K. Ito and K. Xiong. Gaussian filters for nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 45(5):910–928, 2000.

[30] R. J. Jang and C.T. Sun. *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence.* Prentice-Hall, 1996.

[31] D. Koller and N. Friedman. *Probabilistic Graphical Models.* MIT Press, 2009.

[32] K. B. Korb and A. E. Nicholson. *Bayesian Artificial Intelligence.* Series in Computer Science and Data Analysis. Chapman & Hall, first edition edition, 2004.

[33] M. Kvasnica, M. Herceg, L. Cirka, and M. Fikar. Model predictive control of a cstr: a hybrid modeling approach. *Chemical Papers*, 64(3):301–309, 2010.

[34] J.H. Lee, M. Morari, and C.E. Garcia. *Model Predictive Control.* Prentice Hall, 2004.

[35] U.N. Lerner. *Hybrid Bayesian Networks for Reasoning about Complex Systems.* PhD thesis, Stanford Univesity, 2002.

[36] P. Li, M. Wendt, H. Arellano-Garcia, and G. Wozny. Optimal operation of distrillation processes under uncertain inflows accumulated in a feed tank. *American Institute of Chemical Engineers*, 2002.

[37] P. Li, M. Wendt, and G. Wozny. A probabilistically constrained model predictive controller. *Automatica*, 38:1171–1176, 2002.

[38] W.L. Luyben. *Process Modeling, Simulation and Control for Chemical Engineers.* McGraw-Hill, 2nd edition edition, 1990.

[39] J.M. Maciejowski. *Predictive Control with constraints.* Prentice-Hall, 2002.

[40] O. Masahiro. Joint chance-constrained model predictive control with probabilistic resolvability. *American Control Conference*, 2012.

[41] P. Mhaskar, N.H. El-Farra, and P.D. Christofides. Stabilization of nonlinear systems with state and control constraints using lyapunov-based predictive control. *Systems and Control Letters*, 55:650–659, 2006.

[42] K.P. Murphy. Switching kalman filters. Technical report, Compaq Cambridge Research Lab, 1998.

[43] K.P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning.* PhD thesis, University of California, Berkeley, 2002.

[44] K.P. Murphy. *Machine Learning: A Probabilistic Perspective.* MIT Press, 2012.

[45] N. Nandola and S. Bhartiya. A multiple model approach for predictive control of nonlinear hybrid systems. *Journal of Process Control*, 18(2):131–148, 2008.

[46] L. Ozkan, M. V. Kothare, and C. Georgakis. Model predictive control of nonlinear systems using piecewise linear models. *Computers and Chemical Engineering*, 24:793–799, 2000.

[47] T. Pan, S. Li, and W.J. Cai. Lazy learning based online identification and adaptive pid control: a case study for cstr process. *Industrial Engineering Chemical Research*, 46:472–480, 2007.

[48] J.B. Rawlings and D.Q. Mayne. *Model Predictive Control*. Nob Hill Publishing, 2009.

[49] B. Reiser. Confidence intervals for the mahalanobis distance. *Communications in Statistics: Simulation and Computation*, 30(1):37–45, 2001.

[50] Y. Sakakura, M. Noda, H. Nishitani, Y. Yamashita, M. Yoshida, and S. Matsumoto. Application of a hybrid control approach to highly nonlinear chemical processes. *Computer Aided Chemical Engineering*, 21:1515–1520, 2006.

[51] A.T. Schwarm and Nikolaou. Chance constrained model predictive control. Technical report, University of Houston and Texas A&M University, 1999.

[52] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle filtering. *Mathematical Advances in Data Assimilation*, 2008.

[53] S.J. Streicher, S.E. Wilken, and C. Sandrock. Eigenvector analysis for the ranking of control loop importance. *Computer Aided Chemical Engineering*, 33:835–840, 2014.

[54] D.H. van Hessem and O.H. Bosgra. Closed-loop stochastic dynamic process optimisation under input and state constraints. In *Proceedings of the American Control Conference*, 2002.

[55] D.H. van Hessem, C.W. Scherer, and O.H. Bosgra. Lmi-based closed-loop economic optimisation of stochastic process operation under state and input constraints. In *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001.

[56] H. Veeraraghavan, P. Schrater, and N. Papanikolopoulos. Switching kalman filter based approach for tracking and event detection at traffic intersections. *Intelligent Control*, 2005.

[57] D. Wang, W. Wang, and P. Shi. Robust fault detection for switched linear systems with state delays. *Systems, Man and Cybernetics*, 39(3):800–805, 2009.

[58] R.S. Wills. Google's pagerank: the math behind the search engine. Technical report, North Carolina State University, 2006.

[59] J. Yan and R.R. Bitmead. Model predictive control and state estimation: a network example. In *15th Triennial World Conference of IFAC*, 2002.

[60] J. Yan and R.R. Bitmead. Incorporating state estimation into model predictive control and its application to network traffic control. *Automatica*, 41:595–604, 2005.

[61] M.B. Yazdi and M.R. Jahed-Motlagh. Stabilization of a cstr with two arbitrarily switching modes using model state feedback linearisation. *Chemical Engineering Journal*, 155(3):838–843, 2009.