Finite Alphabet Estimation

Graham C. Goodwin

Day 5: Lecture 3

17th September 2004

International Summer School
Grenoble, France
1. Introduction

We have seen in earlier lectures that constrained estimation problems can be formulated in a similar fashion to constrained control problems using the idea of receding horizon optimization.

This is also true of constrained estimation problems where the decision variables must satisfy finite alphabet constraints.
Finite alphabet estimation problems arise in many application, for example,:

- estimation of transmitted signals in digital communication systems where the signals are known to belong to a finite alphabet (say ±1);

- state estimation problems where a disturbance is known to take only a finite set of values (for example, either “on” or “off”).
To fix ideas, we refer to the specific problem of estimating a signal drawn from a given finite alphabet that has been transmitted over a noisy dispersive communication channel. This problem, which is commonly referred to as one of channel equalisation, can be formulated as a fixed-delay maximum likelihood detection problem. The resultant detector estimates each symbol based upon the entire sequence received to a point in time and hence constitutes, in principle, a growing memory structure.
In order to address this problem, various simplified detectors of fixed memory and complexity have been proposed. The simplest such scheme is the decision feedback equaliser [DFE], which is a symbol-by-symbol detector. Recall the development in Day 1: Lecture 1.
Consider a linear channel (which may include a whitening matched filter and any other pre-filter) with scalar input $u_k$ drawn from a finite alphabet $\mathcal{U}$. The channel output $y_k$ is scalar and is assumed to be perturbed by zero-mean additive white Gaussian noise $n_k$ of variance $r$, denoted by $n_k \sim N(0, r)$. 
This is described by the state space model

\[ x_{k+1} = Ax_k + Bu_k, \]
\[ y_k = Cx_k + Du_k + n_k, \]  
(1)

where \( x_k \in \mathbb{R}^n \). The above model may equivalently be expressed in transfer function form as

\[ y_k = H(\rho)u_k + n_k, \quad H(\rho) = D + C(\rho I - A)^{-1}B = h_0 + \sum_{i=1}^{\infty} h_i \rho^{-i}, \]

where

\[ h_0 = D, \quad h_i = CA^{i-1}B, \quad i = 1, 2, \ldots \]  
(2)

\( \rho \) denotes the forward shift operator, \( \rho v_k = v_{k+1} \), where \( \{v_k\} \) is any sequence.
We incorporate an a priori state estimate into the problem formulation. We fix integers $L_1 \geq 0$, $L_2 \geq 1$ and suppose, for the moment, that

$$x_{k-L_1} \sim N(z_{k-L_1}, P),$$

that is, $z_{k-L_1}$ is a given a priori estimate for $x_{k-L_1}$ which has a Gaussian distribution. The matrix $P^{-1}$ reflects the degree of belief in this a priori state estimate. Absence of prior knowledge of $x_{k-L_1}$ can be accommodated by using $P^{-1} = 0$, and decision feedback is achieved by taking $P = 0$, which effectively locks $x_{k-L_1}$ at $z_{k-L_1}$. 
We define the vectors

\[
\mathbf{u}_k \triangleq [u_{k-L_1} \quad u_{k-L_1+1} \quad \cdots \quad u_{k+L_2-1}]^T,
\]

\[
\mathbf{y}_k \triangleq [y_{k-L_1} \quad y_{k-L_1+1} \quad \cdots \quad y_{k+L_2-1}]^T.
\]

The vector \(\mathbf{y}_k\) gathers time samples of the channel output and \(\mathbf{u}_k\) contains channel inputs, which are the decision variables of the estimation problem considered here.
The *maximum a posteriori* [MAP] sequence detector, which at time $t = k$ provides an estimate of $u_k$ and $x_{k-L_1}$ based upon the received data contained in $y_k$, maximises the probability density function $^2$

$$
 p\left(\left[ \begin{array}{c} u_k \\ x_{k-L_1} \end{array} \right] \bigg| y_k \right) = \frac{p\left(y_k \bigg| \left[ \begin{array}{c} u_k \\ x_{k-L_1} \end{array} \right] \right) p\left(\left[ \begin{array}{c} u_k \\ x_{k-L_1} \end{array} \right] \right)}{p\left(y_k \right)}, \quad (4)
$$

$^2$For ease of notation, in what follows we will denote all (conditional) probability density functions by $p$. The specific function referred to will be clear from the context.
Note that only the numerator of the above expression influences the maximisation. Assuming that $u_k$ and $x_{k-L_1}$ are independent, if $u_k$ is white), it follows that

$$p\left(\begin{bmatrix} u_k \\ x_{k-L_1} \end{bmatrix}\right) = p(x_{k-L_1}) p(u_k).$$

Hence, if all finite alphabet-constrained symbol sequences $u_k$ are equally likely (an assumption that we make in what follows), then the MAP detector that maximises is equivalent to the following maximum likelihood sequence detector

$$\begin{bmatrix} \hat{u}_k \\ \hat{x}_{k-L_1} \end{bmatrix} \triangleq \arg \max_{u_k, x_{k-L_1}} \left\{ p(y_k \mid \begin{bmatrix} u_k \\ x_{k-L_1} \end{bmatrix}) p(x_{k-L_1}) \right\}. \quad (5)$$
\[ \hat{u}_k \triangleq \left[ \hat{u}_{k-L_1} \ \hat{u}_{k-L_1+1} \ \cdots \ \hat{u}_k \ \cdots \ \hat{u}_{k+L_2-1} \right]^T, \quad (6) \]

and \( u_k \) needs to satisfy the constraint

\[ u_k \in U^N, \quad U^N \triangleq U \times \cdots \times U, \quad N \triangleq L_1 + L_2, \quad (7) \]

in accordance with the restriction \( u_k \in U \).
Our working assumption is that the initial channel state $x_{k-L_1}$ has a Gaussian probability density function

$$p(x_{k-L_1}) = \frac{1}{(2\pi)^{n/2}(\det P)^{1/2}} \exp \left\{ -\frac{\|x_{k-L_1} - z_{k-L_1}\|^2_{P^{-1}}}{2} \right\}.$$  (8)
We rewrite the channel model at time instants
\( t = k - L_1, k - L_1 + 1, \ldots, k + L_2 - 1 \) in block form as

\[
y_k = \Psi u_k + \Gamma x_{k-L_1} + n_k.
\]

Here,

\[
\begin{align*}
n_k & \triangleq \begin{bmatrix} n_{k-L_1} \\ n_{k-L_1+1} \\ \vdots \\ n_{k+L_2-1} \end{bmatrix}, \\
\Gamma & \triangleq \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix}, \\
\Psi & \triangleq \begin{bmatrix} h_0 & 0 & \ldots & 0 \\ h_1 & h_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ h_{N-1} & \ldots & h_1 & h_0 \end{bmatrix}.
\end{align*}
\]

The columns of \( \Psi \) contain truncated impulse responses of the model.
Since the noise $n_k$ is assumed Gaussian with variance $r$, it follows that

$$p\left(y_k \bigg| \begin{bmatrix} u_k \\ x_{k-L_1} \end{bmatrix} \right) = \frac{1}{(2\pi)^{N/2}(\det R)^{1/2}} \exp \left\{ \frac{-\|y_k - \Psi u_k - \Gamma x_{k-L_1}\|_R^2}{2} \right\}$$

(9)

where the matrix $R \triangleq \text{diag}\{r, \ldots, r\} \in \mathbb{R}^{N \times N}$. 
Applying the natural logarithm, one obtains the sequence detector

\[
\begin{bmatrix}
\hat{u}_k \\
\hat{x}_{k-L_1}
\end{bmatrix} = \arg \min_{u_k, x_{k-L_1}} V(u_k, x_{k-L_1})
\]  

(10)

The objective function \( V \) is defined as

\[
V(u_k, x_{k-L_1}) \triangleq \|x_{k-L_1} - z_{k-L_1}\|_{P^{-1}}^2 + \|y_k - \Psi u_k - \Gamma x_{k-L_1}\|_{R^{-1}}^2
\]

\begin{align*}
&= \|x_{k-L_1} - z_{k-L_1}\|_{P^{-1}}^2 + r^{-1} \sum_{j=k-L_1}^{k+L_2-1} (y_j - C\hat{x}_j - Du_j)^2,
\end{align*}

(11)
The vectors $\tilde{x}_j$ denote predictions of the channel states $x_j$. They satisfy,

$$\tilde{x}_{j+1} = A\tilde{x}_j + Bu_j \quad \text{for} \quad j = k - L_1, \ldots, k + L_2 - 1,$$

$$\tilde{x}_{k-L_1} = x_{k-L_1}. \quad (12)$$
Remark (Notation)

Since $\hat{u}_k$ and $\hat{x}_{k-L_1}$ are calculated using data up to time $t = k + L_2 - 1$, they could perhaps be more insightfully denoted as $\hat{u}_{k|k+L_2-1}$ and $\hat{x}_{k-L_1|k+L_2-1}$, respectively. However, in order to keep the notation simple, we will here avoid double indexing, in anticipation that the context will always allow for correct interpretation.
As a consequence of considering the joint probability density function, the objective function includes a term which allows one to obtain an a posteriori state estimate $\hat{x}_{k-L_1}$ which differs from the a priori estimate $z_{k-L_1}$ as permitted by the confidence matrix $P^{-1}$. 
3. Information Propagation

Having set up the fixed horizon estimator as the finite alphabet optimiser, we next show how this information can be utilised as part of a moving horizon scheme.
Minimisation of the objective function $V$ yields the entire optimising sequence $\hat{u}_k$. However, following our usual procedure, we will utilise a moving horizon approach in which only the present value$^3$

$$\hat{u}_{k}^{\text{OPT}} \triangleq \begin{bmatrix} 0_{L_1} & 1 & 0_{L_2-1} \end{bmatrix} \hat{u}_k,$$  

(13)

will be delivered at the output of the detector.

---

$^3$The row vector $0_m \in \mathbb{R}^{1 \times m}$ contains only zeros.
At the next time instant the optimisation is repeated, providing $\hat{u}_{k+1}^{\text{OPT}}$ and so on. Thus, the data window “slides” (or moves) forward in time. The scheme previews $L_2 - 1$ samples, hence has a decision-delay of $L_2 - 1$ time units.

The window length $N = L_1 + L_2$ fixes the complexity of the computations needed. It is intuitively clear that good performance of the detector can be ensured if $N$ is sufficiently large. However, in practice, there is a strong incentive to use small values for $L_1$ and $L_2$, since large values give rise to high complexity in the associated computations to be performed at each time step.
4. Decision-Directed Feedback

The provision of an a priori estimate, $z_{k-L_1}$, together with an associated degree of belief via the term $\|x_{k-L_1} - z_{k-L_1}\|^2_{P^{-1}}$ in (11) provides a means of propagating the information contained in the data received before $t = k - L_1$. Consequently, an information horizon of growing length is effectively obtained in which the computational effort is fixed by means of the window length $N$. 
One possible approach to choose the a priori state estimate is as follows: Each optimisation step provides estimates for the channel state and input sequence. These decisions can be re-utilised in order to formulate a priori estimates for the channel state $x_k$. We propose that the estimates be propagated in blocks according to

$$z_k = A^{N} \hat{x}_{k-N} + M \hat{u}_{k-L_2},$$

where $M \triangleq \begin{bmatrix} A^{N-1}B & A^{N-2}B & \ldots & AB & B \end{bmatrix}$. In this way, the estimate obtained in the previous block is rolled forward. Indeed, in order to operate in a moving horizon manner, it is necessary to store $N$ a priori estimates.

\footnote{Since $z_k$ is based upon channel outputs up to time $k-1$, it could alternatively be denoted as $\hat{x}_{k|k-1}$; see also Remark ??}. 
Figure: Information propagation with parameters $L_1 = 1$ and $L_2 = 2$. 
Since channel states depend on the finite alphabet input, one may well question the assumption made above that $x_{k-L_1}$ is Gaussian. However, we could always use this structure by interpreting the matrix $P$ as a design parameter.
As a guide for tuning $P$, we recall that in the unconstrained case, where the channel input and initial state are Gaussian, that is, $u_k \sim N(0, Q)$ and $x_0 \sim N(\mu_0, P_0)$, the Kalman filter provides the minimum variance estimate for $x_{k-L_1}$. Its covariance matrix $P_{k-L_1}$ obeys the Riccati difference equation.

$$P_{k+1} = AP_k A^T - K_k (CP_k C^T + r + DQD^T)K_k^T + QBQ^T, \quad k \geq 0, \quad (14)$$

where $K_k \triangleq (AP_k C^T + BQD^T)(CP_k C^T + r + DQD^T)^{-1}$. 
A further simplification occurs if we replace the above recursion by its steady state equivalent. In particular, it is well-known that, under reasonable assumptions, $P_k$ converges to a steady state value $P$ as $k \to \infty$. The matrix $P$ satisfies the following algebraic Riccati equation:

$$P = APA^T - K(CPC^T + r + DQD^T)K^T + QBQ^T,$$  \hspace{1cm} (15)

where $K = (APC^T + BQD^T)(CPC^T + r + DQD^T)^{-1}$. 
Of course, the Gaussian assumption on $u_k$ is not valid in the constrained case. However, the choice may still provide good performance. Alternatively, one may simply use $P$ as a design parameter and test different choices via simulation studies.
Here we follow similar arguments to those used with respect to finite alphabet control to obtain a closed form expression for the solution to the finite alphabet estimation problem. This closed form expression utilises a vector quantiser as defined earlier.
Lemma: Closed Form Solution

The optimisers given the constraint $u_k \in \mathbb{U}^N$ are given by

$$\hat{u}_k = \Omega^{-1/2} q_{\mathbb{U}^N} (\Omega^{-1/2} (\Lambda_1 y_k - \Lambda_2 z_{k-L_1})), \quad (16)$$

$$\hat{x}_{k-L_1} = \gamma \left( P^{-1} z_{k-L_1} + \Gamma^T R^{-1} y_k - \Gamma^T R^{-1} \Psi \hat{u}_k \right), \quad (17)$$

where

$$\Omega = \Psi^T \left( R^{-1} - R^{-1} \Gamma \gamma \Gamma^T R^{-1} \right) \Psi, \quad \Omega^{1/2} \Omega^{1/2} = \Omega,$$

$$\gamma = (P^{-1} + \Gamma^T R^{-1} \Gamma)^{-1},$$

$$\Lambda_1 = \Psi^T \left( R^{-1} - R^{-1} \Gamma \gamma \Gamma^T R^{-1} \right),$$

$$\Lambda_2 = \Psi^T R^{-1} \Gamma \gamma P^{-1}. \quad (18)$$
The nonlinear function $q_{\tilde{U}^N}(\cdot)$ is the nearest neighbour vector quantiser. The image of this mapping is the set

$$\tilde{U}^N = \Omega^{1/2} U^N \triangleq \{ \tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_r \} \subset \mathbb{R}^N,$$

with $\tilde{v}_i = \Omega^{1/2} v_i$, $v_i \in U^N$. 

(19)