Signal Processing 108 (2015) 341–350

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Analysis of reduced-search BCJR algorithms for input estimation in a jump linear system

Fabio Fagnani [a], Sophie M. Fosson [b,*]

[a] Department of Mathematical Sciences, Politecnico di Torino, Italy
[b] Department of Electronics and Telecommunications, Politecnico di Torino, Italy

ARTICLE INFO

Article history:
Received 30 May 2013
Received in revised form
26 May 2014
Accepted 18 September 2014
Available online 28 September 2014

Keywords:
Input estimation
Jump linear systems
Reduced-search BCJR algorithms
Markov processes

ABSTRACT

Linear systems with unknown finite-valued inputs are of interest in all those hybrid frameworks where switches or jumps may change the continuous dynamics of a linear system. Many models have been proposed in this sense; in most cases, a probabilistic distribution on the input is assumed to be known and used as prior information for estimation. In this paper, we propose a simple model of jump linear system and develop low complexity algorithms, based on BCJR, to retrieve the input. We consider systems over a possibly infinite time horizon, which motivates the study of on-line, causal algorithms. Our main purpose is to provide a rigorous theoretical analysis of the performance of the proposed techniques: an error function is defined and its distribution is proved to converge, exploiting mathematical tools from Markov Processes theory and ergodic theorems.

1. Introduction

In this paper, we consider the linear system

$$\begin{cases} x_{k+1} = x_k + u_k \\ y_k = x_k + n_k \end{cases} \tag{1}$$

where $k = 0, \ldots, K-1$, $K \in \{1, 2, \ldots, \infty\}$; $\mathbf{u} = (u_0, \ldots u_{K-1}) \in \mathcal{U}^K$ is the unknown input sequence, whose components are symbols from a finite alphabet $\mathcal{U}$; $\mathbf{x} = (x_1, \ldots, x_K) \in \mathcal{X}^K$ is the unknown state sequence (as $\mathcal{U}$ is finite, $\mathcal{X}$ is countable); $\mathbf{y} = (y_1, \ldots, y_K) \in \mathbb{R}^K$ are the observables, corrupted by noises $n_1, \ldots, n_K$. The initial state is fixed to $x_0 = 0$. We complete the model by assuming the following probabilistic setting: the u_k's are realizations of i.i.d. uniform, Bernoulli random variables U_k's, while the n_k's are realizations of i.i.d. zero-mean Gaussian random variables N_k's with variance σ^2 (independent from U_k's). Our goal is to recover the input given the observables, the knowledge of $\mathcal{U}$, and the knowledge of the underlying probability distributions.

Even though very simple, this model has not been specifically addressed yet. It is indeed clear that by minimal variations it can be completely cast in different frameworks, like hidden Markov models (HMM) and input estimation (or deconvolution) of discrete linear systems. Just to mention a few examples, it can be interpreted as (a) an input/output HMM [1, Section 5]; (b) a deconvolution of discrete or quantized input linear systems [2–5]; (c) an intersymbol interference problem (ISI, [6]). In these contexts, however the nature of the problem and the approach are slightly different, e.g., because (a) the estimation of the states (and not of the input) generally is the main purpose; (b) deterministic and finite time horizon models are considered; (c) convolution filters of finite length are studied, and feedback inputs are envisaged for equalization purposes.

A further, more fitting setting for our model is the mode estimation problem in Markov jump linear systems, which are particular HMM with non-mutually independent

* Corresponding author.
E-mail address: sophie.fosson@polito.it (S.M. Fosson).

http://dx.doi.org/10.1016/j.sigpro.2014.09.027
0165-1684/

observables [7, Section 1.2]. Members of the family of the switching systems (namely, hybrid systems whose dynamics may change due to a finite-valued switch input), Markov jump systems are characterized by switch inputs ruled by finite-state Markov chains, which can well represents a number of applications [8].

In the literature on Markov jump systems, much effort has been devoted to state estimation, control and stabilization, using, among others, Kalman filtering and Viterbi algorithm [9–13]. Less attention has been instead devoted to mode estimation, that is, to the estimation of the input that originates the jumps [11,14,15]. The goal of this work is to focus on such input estimation. A practical example of mode estimation can be drawn from the failure detection framework: a switch may represent a failure that causes irreversible damages, then one just aims to detect it as soon as possible to stop the system before it breaks down, with no regard to tasks as stabilization or fault-tolerance. Promptness is indeed fundamental. Following such rationale, in this work we address to on-line, causal estimation, that is, at time step $k=1,2,\dots$ we provide an estimation of the current input symbol u_{k-1} based on the observation of $y_1,\dots,y_k$. We can then assume that $K\to\infty$, that is, in principle the system evolves over an infinite time horizon.

Our focus is thus to study algorithms able to envisage all the mentioned points and above all provide a rigorous theoretical analysis of their performance, namely an analytical computation of a suitable error function. Our attention to the theoretical results justifies the simplicity of the model, which is far from real systems in particular for (a) the one-dimensionality; (b) the basic dynamics; (c) the input distribution (the inputs are independent, not ruled by a Markov chain). However, these oversimplifying assumptions allow us to keep the analysis readable, while extensions in the direction of multi-dimensional, generic linear systems, and Markov sources are not difficult to design and implement [16,17]. An other assumption we will do throughout the paper is that $\mathcal{U}=\{0,1\}$, that is, the input is binary; this does not affect the nature of the problem, has been considered in many models [7, Example 10.3.2], and may have non-trivial applications, e.g., in image processing [18] and failure detection [19]. Extensions will be however discussed after our exposition.

The literature about unknown input estimation in linear systems is extensive, but more oriented to continuous inputs (see, e.g., [20–22] and the more recent [23,24]). In most of these works, Kalman filtering based techniques are developed to deal with unknown inputs. In principle, such techniques could be adapted also to the discrete input case (for example, by quantizing the computed estimation), but studying methodologies that take into account the discrete nature from the beginning is preferable.

Given the binary, probabilistic nature of the input, the techniques we will use for the estimation are naturally based on bit-MAP (Maximum a posteriori) estimation. Since we aim at working on-line, we need iterative, causal methods, and the optimal choice in this sense is a causal version of the well-known BCJR algorithm (that we will name cBCJR), which implements only the forward recursion of the BCJR. Since the number of possible states may tend to infinity, BCJR and cBCJR are affected by complexity problems. For this motivation, we will reduce the number of survivor states at each step. More precisely, we will analyze the cases of 1 and 2 survivor states (respectively named 1-cBCJR and 2-cBCJR) that provide good performance despite their very low complexity. In particular, 2-cBCJR will be shown to have performance close to cBCJR.

In the last few decades, BCJR has been mainly used in digital transmissions for the decoding of convolutional codes. This observation yields to notice that our system itself could be described in terms of a digital transmission problem: we have a binary input message, that is somehow *encoded* by an operation of convolution, and then *transmitted* over an AWGN channel (as the observation noise we consider is white Gaussian). Hence, the input estimation algorithms that we will study actually are *decoding* algorithms. The main difference with respect to convolutional codes setting lies in the non finite-state trellis. Moreover, as we will discuss later, performance is not uniform with respect to different inputs.

The paper is organized as follows. In Section 2, we introduce our model, describe its properties, and discuss a suitable metric to evaluate the estimation error. In Section 3, we derive the algorithms that minimize such error and modify them to obtain causality and low complexity. Numerical simulations are the proposed to check their good performance. Section 4 is devoted to the theoretical analysis of 1-cBCJR; technical proofs and mathematical details are postponed to Section 5. In Section 7, we finally discuss the theoretical analysis of 2-cBCJR.

We conclude the introduction with some notation to be used later in the paper.

1.1. Notation

In the following, we will denote by $\mathbb{N}_0$ the set of non-negative integer numbers. We will use capital letters to name random variables. Given a set A, we define the indicator function $\mathbb{1}_A(x)=1$ if $x\in A$, $\mathbb{1}_A(x)=0$ otherwise; $\mathrm{erfc}(x){:=}2/\sqrt{\pi}\int_x^{+\infty}e^{-s^2}\,ds$, $x\in\mathbb{R}$, is the complementary error function. We will denote by $\mathbf{P}$ the transition probability matrix of a Markov chain and by $P(\cdot,\cdot)$ the transition probability kernel of a Markov process; $\mathbb{E}$ will be the mean operator. Moreover, we will use the abbreviations i.i.d. for independent, identically distributed; p.v. for probability vector; p.m. for probability measure.

Finally, we will use the following acronyms: BER for Bit Error Rate; CBER for conditional BER; BCJR for the decoding algorithm by Bahl, Cocke, Jelinek and Raviv; MCRE for Markov chain in random environment; EMP for extended Markov process.

2. Problem statement and error functions

Given system (1) with the described assumptions, we consider the problem

$$\text{(P1): given } \mathbf{y}=(y_1,\dots,y_K),\ \text{estimate } \mathbf{u}=(u_0,\dots,u_{K-1})$$

and its causal version

$$\text{(P2): for any } k=1,\dots,K,\ \text{given } \mathbf{y}_1^k=(y_1,\dots,y_k),\ \text{estimate } u_{k-1}.$$

As said in the introduction, much attention will be devoted to (P2), to handle systems evolving in time and potentially