Acknowledgment

The authors wish to thank V. R. Sidorenko and A. Trushkin for helpful comments.

References

[1] L. R. Bahl, J. Cooke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.

[2] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, Mar. 1996.

[3] J. Hagenauer, "Source controlled channel decoding," *IEEE Trans. Commun.*, vol. 43, pp. 2449–2457, Sept. 1995.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding," in *Proc. IEEE Int. Conf. Communications* (Geneva, Switzerland, May 1993), pp. 1064–1070.

[5] R. Lucas, M. Bossert, and M. Breitbach, "On iterative soft-decision decoding of linear binary block codes and product codes," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 276–296, Feb. 1998.

[6] C. R. Hartmann and L. D. Rudolph, "An optimal symbol-by-symbol decoding rule for linear codes," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 514–517, Sept. 1976.

[7] G. Battail, M. C. Decouvelaere, and P. Godlewski, "Replication decoding," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 332–345, Sept. 1979.

[8] J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 76–80, Jan. 1978.

[9] V. V. Zyablov, and V. R. Sidorenko, "Bounds on complexity of trellis decoding of linear block codes," *Probl. Pered. Inform.*, vol. 29, no. 3, pp. 1–6, July–Sept. 1993.

[10] R. J. McEliece, "On the BCJR trellis for linear block codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1072–1092, July 1996.

[11] D. J. Muder, "Minimal trellises for block codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1049–1053, Sept. 1988.

[12] M. Lentmaier, "Soft iterative decoding of generalized low-density parity-check codes based on MAP decoding of component Hamming codes," Diploma Thesis, Dept. Inform. Technol., University of Ulm, Ulm, Germany, 1997.

A Soft Output Hybrid Algorithm for ML/MAP Sequence Estimation

Gary D. Brushe, *Member, IEEE*, Robert E. Mahony, *Member, IEEE*, and John B. Moore, *Fellow, IEEE*

***Abstract*— The classical Viterbi algorithm (ML sequence estimation) can be computed using a forward-backward structure, similar to that of the classical hidden Markov model forward-backward algorithm (MAP state estimation). This similarity is exploited to develop a hybrid algorithm which provides a mathematical connection between ML sequence estimation and MAP state estimation.**

***Index Terms*—Forward-backward algorithms, hidden Markov models, HMM's, sequence, soft outputs, state estimation, Viterbi algorithm.**

I. Introduction

The basic filter theory for hidden Markov models was first presented by Baum and his colleagues in a series of papers in the late 1960's and early 1970's [1]–[5]. These papers developed statistical estimation algorithms for a discrete-time Markovian process observed in noise. The model structure became known as a hidden Markov model (HMM) and since the mid-1980's has become increasingly popular in many engineering applications (e.g., signal and speech processing). In the field of communications, the typical application is one where the signal statistics are known *a priori* and the goal is to estimate the transmitted state sequence from observed data. There are two natural measures of the most probable state sequence transmitted. First, at each separate time, one may consider the states which are individually most likely. This is equivalent to estimating the maximum *a posteriori* probability (MAP) (sometimes known as minimum variance or conditional mean) state estimates. These estimates can be determined using the HMM forward-backward algorithm (HFBA) [7]. Conversely, one may consider the most likely state sequence over all the data. This is equivalent to computing the maximum-likelihood (ML) state sequence for the full data stream. This estimate can be determined using the Viterbi algorithm (VA) [7].

In 1974, Bahl *et al.* [8] explored the use of MAP estimation in decoding linear block and convolutional codes in order to minimize the symbol- (or bit-) error rate. They concluded that the increased computational complexity of MAP estimation was not warranted as the performance of the VA and the MAP estimators were effectively identical in the applications they considered. More recently,

Manuscript received September 11, 1995; revised January 13, 1998. The work of the authors at the Cooperative Research Centre for Robust and Adaptive Systems was supported by the Australian Commonwealth Government under the Cooperative Research Centre Program. The material in this correspondence was presented in part at the ISSPA'96, Gold Coast, Australia, August 25–30, 1996.

G. D. Brushe is with the Signal Analysis Discipline, Communications Division, ESRL of the Defence Science and Technology Organization, Salisbury, SA 5108, Australia, and the Cooperative Research Centre for Robust and Adaptive Systems, c/- The Australian National University, Canberra, ACT, Australia.

J. B. Moore is with the Department of Systems Engineering, RSISE, The Australian National University, Canberra, ACT, Australia, and the Cooperative Research Centre for Robust and Adaptive Systems, c/- The Australian National University, Canberra, ACT, Australia.

R. E. Mahony is with the Cooperative Research Centre for Robust and Adaptive Systems, c/- The Australian National University, Canberra, ACT, Australia.

Publisher Item Identifier S 0018-9448(98)06901-6.

a renewed interest in MAP estimates has followed from the interest in soft-output Viterbi algorithms (SOVA's) [10]. These algorithms are necessary in demodulation, decoding, and equalization problems [9] in telecommunications, where several decoding algorithms need to be concatenated together, and the hard-decision output of the traditional VA applied early in the procedure results in degradation of performance in later decoding steps. Recent work in investigating computationally efficient approximations of the HFBA has lead to the development of an algorithm referred to as the Max-Log-MAP algorithm [20]. This algorithm is of interest since Robertson *et al.* [20] showed that maximizing the metric provided by the Max-Log-MAP algorithm results in the same state sequence estimate that is obtained by the VA and, consequently, must be related in some manner to ML probabilities. There has been previous work in computing ML probability distributions at each time instance in a data stream (cf., for example, Hayes *et al.* [19]), however, the Max-Log-MAP algorithm provides an efficient algorithm for computing these probability distributions. Nevertheless, algorithms such as the Max-Log-MAP algorithm are computationally more expensive than the VA. They do, however, generate soft outputs as well as estimating the same state sequence as the VA. Recent work in efficient VLSI implementations [21] along with improvements in the speed of low-cost DSP chips reduce the dependence on low computational cost solutions.

In this correspondence we provide a theoretical study of the similarities and differences between the MAP and the ML state sequence estimates and the HFBA and the VA used for the computation of these estimates. Our goal is to provide a simple extension of the concepts of the HFBA and the VA couched in the terminology adopted by Rabiner *et al.* [6], [7]. We show that the interpretation of the VA as the maximization of an ML probability measure can be extended to provide an ML probability distribution for states at each separate time instance as well as over the full data set. Interestingly, the computation of this probability distribution is achieved in an analogous manner to the HFBA and we refer to this algorithm as the Viterbi forward-backward algorithm (VFBA). The logarithmic version of the VFBA generates the Max-Log-MAP algorithm presented in [20].

The similarity between the HFBA and the VFBA leads to the concept of a hybrid algorithm. The hybrid algorithm (based on the Varadhan–Laplace lemma [11], [12]) provides a continuously varying connection between ML sequence estimation (via the VFBA and hence the VA) and MAP state estimation (via the HFBA). This connection is achieved using a tuning parameter $0 < \mu < \infty$. We show that in the limits, $\mu \to \infty$ and $\mu \to 0$, the hybrid algorithm yields the VFBA and the HFBA, respectively. The construction of the hybrid algorithm emphasises the similarities in processing data for ML or MAP state estimates. Indeed, there may be applications where a delicate performance dependence exists between ML and MAP state estimates. In such cases, the use of a hybrid scheme (with on-line adaptation of the tuning parameter) may result in performance gains.

The scope of this correspondence, however, is the thorough theoretical development of the hybrid algorithm. Practical applications for the hybrid algorithm is the subject of continuing research.

In Section II of this correspondence, we present a hidden Markov model. In Section III, we briefly describe the VA, the HFBA, and the VFBA. These three algorithms are presented using a consistent notation (cf. [7]), so that their similarities can be readily identified. Section IV presents the hybrid algorithm while Section V is the conclusion.

II. Model

Consider the following system known as a hidden Markov model (cf. Rabiner [7]). Assume a finite number N of states, $S = \{S_1, S_2, \cdots, S_N\}$, and denote the state at time t by q_t.

The process is assumed to be Markovian, that is,

$$P[q_{t+1} = S_j | q_t = S_i, q_{t-1} = S_k, \cdots] = P[q_{t+1} = S_j | q_t = S_i] := a_{ij} \qquad (1)$$

where $a_{ij}, 1 \le i, j \le N$ is known as the state transition probability from state S_i to state S_j.

The states are observed via a process

$$O_t = C(q_t) + w_t \qquad (2)$$

where C is a deterministic function (which, in a communications system, would be determined by the modulation type of the signal) and w_t is the noise process. Denote the probability density function of w_t by $\phi(\ )$. The probability of observing O_t given state S_j at time $t, b_j(O_t)$, is given by

$$b_j(O_t) = P[O_t | q_{\hat{t}} = S_j] = \phi(O_t - C(S_j))], \qquad 1 \le j \le N. \qquad (3)$$

Typically, we assume white Gaussian noise with variance σ^2, thus

$$b_j(O_t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-1}{2\sigma^2} |O_t - C(S_j)|^2\right), \qquad 1 \le j \le N. \qquad (4)$$

To complete the model it is usual to include the initial state distribution π, where

$$\pi(i) = P[q_1 = S_i], \qquad 1 \le i \le N. \qquad (5)$$

Though in signal processing applications, where no *a priori* state information is known, then π is usually chosen uniform, $\pi = ((1/N), \cdots, (1/N))$.

III. ML Sequence and MAP State Estimation Algorithms

In this section we briefly review the Viterbi algorithm (VA), the HMM forward-backward algorithm (HFBA) and the Viterbi forward-backward algorithm (VFBA). These algorithms are all presented using a consistent notation (cf. [7]) which is also carried through in Section IV. It should be noted that practical implementations for these algorithms exist [20], [21] and they should be used in preference to directly implementing the algorithms as they are presented in this section. The VA and VFBA determine ML state sequence estimates, while the HFBA determines MAP state estimates. The VFBA and the HFBA also provide probability measures (reliability information) for the state estimates.

A. The Viterbi Algorithm

The Viterbi algorithm (VA) [13] determines the ML state sequence over the time interval $\{1, 2, \cdots, T\}$ given a sequence of observations $O = \{O_1, O_2, \cdots, O_T\}$. This is accomplished using forward dynamic programming [14]. Let $\delta_t(i)$ denote the ML information probability measure, that is, the probability of being in state S_i at time t maximized over all possible paths which end in state S_i, with the observed data up to time t

$$\delta_t(i) = \max_{q_1, q_2, \cdots, q_{t-1}} P[q_1, \cdots, q_{t-1}, q_t = S_i, O_1, O_2, \cdots, O_t]. \qquad (6)$$

This can be computed via the recursion [7]

$$\delta_t(i) = \max_j [\delta_{t-1}(j) a_{ji}] b_i(O_{\hat{t}}), \qquad 1 \le i \le N, 2 \le t \le T \qquad (7)$$

and associated with an estimate of the most likely state at time T

$$\hat{q}_T = \arg\max_{1 \le j \le N} [\delta_T(j)] \qquad (8)$$

there is an estimated state sequence

$$\hat{q}_1, \cdots, \hat{q}_{T-1} = \arg \max_{q_1, \cdots, q_{T-1}} P[q_1, \cdots, q_{T-1}, q_T = \hat{q}_T, O].$$

In practice the maximizing state sequence is extracted using a backtracking process. This is done by keeping track of the argument which maximized $\delta_t(i)$ at each time

$$\psi_t(i) = \arg \max_{1 \le j \le N} [\delta_{t-1}(j) a_{ji}], \qquad 1 \le i \le N, \, 2 \le t \le T \tag{9}$$

and then backtracking to obtain the most likely state sequence estimate, i.e.,

$$\hat{q}_t = \psi_{t+1}(\hat{q}_{t+1}), \qquad t = T-1, T-2, \cdots, 1. \tag{10}$$

B. The HMM Forward-Backward Algorithm

The HMM forward-backward algorithm (HFBA) computes the MAP state probabilities of the transmitted sequence given a block of observations. Let $\gamma_t(i)$ denote the probability of being in state S_i at time t given the observed data sequence:

$$\gamma_t(i) := P[q_t = S_i | O] \tag{11}$$

where $O = \{O_1, O_2, \cdots, O_T\}$ is the observed data sequence. The MAP state estimates are obtained by maximizing γ_t over i

$$\hat{q}_t = \arg \max_i [\gamma_t(i)], \qquad 1 \le t \le T. \tag{12}$$

To compute γ_t, using Bayes rule, observe that

$$P[q_t = S_i | O] = \frac{P[q_t = S_i, O]}{P[O]} \tag{13}$$

and $P[q_t = S_i, O]$ can be factored into

$$\begin{aligned} P[q_t, = S_i, O] &= P[O_1, O_2, \cdots, O_t, q_t = S_i] \\ &\quad \cdot P[O_{t+1}, O_{t+2}, \cdots, O_T | q_t = S_i] \\ &= \alpha_t(i) \beta_t(i) \end{aligned} \tag{14}$$

where

$$\alpha_t(i) = P[O_1, O_2, \cdots, O_t, q_t = S_i]$$

depends only on information obtained up to time t and

$$\beta_t(i) = P[O_{t+1}, O_{t+2}, \cdots, O_T | q_t = S_i]$$

which depends only on information from $t+1$ to T. Commonly, $\alpha_t(i)$ and $\beta_t(i)$ are referred to as the forward and backward information probability measures, respectively.

They can be computed recursively as follows (cf. [7]):

$$\begin{aligned} \alpha_t(i) &= \left[\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji} \right] b_i(O_t), \\ \alpha_0(i) &= \pi(i), \qquad 1 \le i \le N, \;\; 2 \le t \le T. \end{aligned} \tag{15}$$

$$\begin{aligned} \beta_t(i) &= \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \\ \beta_T(i) &= 1, \qquad 1 \le i \le N, \;\; T-1 \ge t \ge 1. \end{aligned} \tag{16}$$

C. The Viterbi Forward-Backward Algorithm

The Viterbi forward-backward algorithm (VFBA) computes an *a posteriori* probability measure for each state at each time but which is maximized over all valid paths which pass through that state. This probability measure gives a degree of confidence for the state estimate obtained at each time. The probability measures could also be directly used as soft outputs to a next stage VA, as required in communications systems which use concatenated VA's.

Consider the ML information probability measure $\delta_{\hat{t}}(i)$, (6), given in the VA. Assuming that one is given a block of observed data $O = \{O_1, O_2, \cdots, O_T\}$ then it is reasonable to consider a new information probability measure at time $t \in \{1, \cdots, T\}$ determined by

$$\begin{aligned} \tilde{\gamma}_t(i) = \max_{q_1, \cdots, q_{t-1}} \max_{q_{t+1}, \cdots, q_T} & P[q_1, \cdots, q_{t-1}, q_t = S_i, \\ & q_{t+1}, \cdots, q_T, O], \qquad 1 \le i \le N, \quad 1 \le t \le T. \end{aligned} \tag{17}$$

This information probability measure determines the probability of being in state S_i at time t maximized over all possible paths which pass through state S_i, given the observed data sequence $O = \{O_1, O_2, \cdots, O_T\}$. We will refer to this information probability measure as the "*a posteriori* maximum path probability (AMPP)" measure.

To compute (17), we split it into two parts (analogous to (14)).

$$\begin{aligned} \tilde{\gamma}_t(i) &= \max_{q_1, \cdots, q_{t-1}} P[q_1, \cdots, q_{t-1}, q_t = S_i, O_1, \cdots, O_t] \\ &\quad \cdot \max_{q_{t+1}, \cdots, q_T} P[q_{t+1}, \cdots, q_T, O_{t+1} \cdots, O_T | q_t = S_i] \\ &= \delta_t(i) \tilde{\beta}_t(i). \end{aligned} \tag{18}$$

Observe that the forward probability measure $\delta_t(i)$ is exactly the standard VA's ML information probability measure, while the backward probability measure,[1] $\tilde{\beta}_t(i)$ can be computed inductively (analogous to Rabiner [7]) as follows:

1) Initialization:

$$\tilde{\beta}_T(i) = 1, \qquad 1 \le i \le N. \tag{19}$$

2) Recursion:

$$\begin{aligned} \tilde{\beta}_t(i) &= \max_{1 \le j \le N} [\tilde{\beta}_{t+1}(j) a_{ij} b_j(O_{t+1})], \\ &\qquad 1 \le i \le N, \, T-1 \ge t \ge 1. \end{aligned} \tag{20}$$

The AMPP $\tilde{\gamma}_t(i)$ is a probability measure for every state at each time, based on path constraints. To obtain a state sequence estimate from the VFBA choose the state with the maximum AMPP $\tilde{\gamma}_t(i)$ at each time t, i.e.,

$$\hat{q}_t = \arg \max_i [\tilde{\gamma}_t(i)], \qquad 1 \ge t \ge T. \tag{21}$$

Note: At each time t, a probability distribution is determined (i.e., the AMPP distribution) and $\hat{q}_t$ is the state with the maximum AMPP (cf. the Max-Log-MAP algorithm [20]). Also, the state sequence $\{\hat{q}_t\}$ is the same as would be obtained via the VA (see [20] for details).

It may also be possible to reduce the computational complexity of the VFBA by using implementations such as those described in [20] and [21] or by applying known HMM based techniques (i.e., on-line implementation using fixed-lag or sawtooth-lag smoothing [15]). Reduced state Viterbi techniques [16]–[18] may also be applicable.

[1] Recall that the Markov property ensures that once $q_t = S_i$ is taken as known, all other prior data becomes redundant.

IV. A Hybrid Viterbi/HMM Forward-Backward Algorithm

Comparing the VFBA and the HFBA shows that the structures of these two algorithms is similar. In this section, the structural similarity is exploited to develop a hybrid Viterbi/HMM forward-backward algorithm[2] which provides a mathematical connection (via a tuning parameter μ) between computing the VA sequence estimates and the HFBA state estimates. This single hybrid algorithm can be used to compute either ML sequence estimates (using values of $\mu \to \infty$) or MAP state estimates (using values of $\mu \to 0$), for a given a block of observations. The hybrid algorithm is also able to provide estimates which interpolate between ML sequence and MAP state estimates (using values of μ between 0 and ∞), that is, if there is an application where, in some sense, this provides meaningful estimates.

Let μ be a real positive parameter (i.e., $0 < \mu < \infty$) and consider the μ-dependent forward and backward information probability measures ($\kappa_t^\mu(i)$ and $\tau_t^\mu(i)$, respectively) defined recursively as

$$\begin{aligned}\kappa_t^\mu(i) &= b_i(O_t)\,\frac{(1+(N-1)e^{-\mu})}{\mu}\\ &\quad \cdot \ln\left(\frac{1}{N}\sum_{j=1}^{N}\left[\exp(\mu\kappa_{t-1}^\mu(j)a_{ji}\right]\right)\\ \kappa_1^\mu(i) &= \pi(i), \qquad 2 \le t \le T \end{aligned} \tag{22}$$

$$\begin{aligned}\tau_t^\mu(i) &= \frac{(1+(N-1)e^{-\mu})}{\mu}\\ &\quad \cdot \ln\left(\frac{1}{N}\sum_{j=1}^{N}\left[\exp(\mu\tau_{t+1}^\mu(j)a_{ij}b_j(O_{t+1}))\right]\right)\\ \tau_T^\mu(i) &= 1, \qquad T-1 \ge t \ge 1\end{aligned} \tag{23}$$

for $1 \le i \le N$. Let $\theta_t^\mu(i)$ be defined as the hybrid *a posteriori* information probability measure, and determined by the product of the forward and backward information probability measures, i.e.,

$$\theta_t^\mu(i) = \kappa_t^\mu(i)\tau_t^\mu(i), \qquad 1 \le t \le T. \tag{24}$$

A maximum hybrid *a posteriori* state estimate can be obtained by determining which state has the maximum value of $\theta_t^\mu(i)$ at each time, i.e.,

$$\hat{q}_t^\mu = \arg\max_{1\le j\le N}\,[\theta_t^\mu(j)], \qquad 1 \le t \le T \tag{25}$$

although the $\theta_t^\mu(i)$'s could also be used as soft outputs.

Remark: In practice, the $\ln((1/N)\,\Sigma_{j=1}^N\,[\exp(\cdot)])$ would be computed using the Jacobian logarithm in a manner similar to [20, eq. (16)].

The motivation for considering the information probability measures ((22) and (23)) is given by the following proposition.

Proposition 1: Let $d_j\colon \Re^+ \to \Re^+, j = 1, \cdots, N$, be continuous functions with well-defined limits at zero and infinity (that is, $d_j^0 = \lim_{\mu\to 0}\,\{d_j(\mu)\}$ and $d_j^\infty = \lim_{\mu\to\infty}\,\{d_j(\mu)\}$). Then

a) $$\lim_{\mu\to 0}\left\{\frac{1}{\mu}\,\ln\left(\frac{1}{N}\sum_{j=1}^{N}\exp(\mu d_j(\mu))\right)\right\} = \frac{1}{N}\sum_{j=1}^{N} d_j^0 \tag{26}$$

b) $$\lim_{\mu\to\infty}\left\{\frac{1}{\mu}\,\ln\left(\frac{1}{N}\sum_{j=1}^{N}\exp(\mu d_j(\mu))\right)\right\} = \max_j\,[d_j^\infty]. \tag{27}$$

[2] This will be referred to as the "Hybrid Algorithm" in the rest of this correspondence.

Proof:
Part a):
Using the asymptotic expansion

$$e^x = 1 + x + O(x^2), \qquad \text{for } x \to 0$$

gives

$$\begin{aligned}&\lim_{\mu\to 0}\left\{\frac{1}{\mu}\,\ln\left(\frac{1}{N}\sum_{j=1}^{N}\exp(\mu d_j(\mu))\right)\right\}\\ &\quad = \lim_{\mu\to 0}\left\{\frac{1}{\mu}\,\ln\left(\frac{1}{N}\sum_{j=1}^{N}(1+\mu d_j(\mu)+O(\mu^2))\right)\right\}\\ &\quad = \lim_{\mu\to 0}\left\{\frac{1}{\mu}\,\ln\left(1+\frac{\mu}{N}\sum_{j=1}^{N}d_j(\mu)+O(\mu^2)\right)\right\}.\end{aligned} \tag{28}$$

Similarly, using the approximation

$$\ln(1+x) = x + O(x^2), \qquad \text{for } x \to 0$$

and observing that $\mu\,\Sigma_{j=1}^N\,d_j(\mu) = O(\mu)$ for $\mu \to 0$, gives,

$$\begin{aligned}&\lim_{\mu\to 0}\left\{\frac{1}{\mu}\,\ln\left(\frac{1}{N}\sum_{j=1}^{N}\exp(\mu d_j(\mu))\right)\right\}\\ &\quad = \lim_{\mu\to 0}\left\{\frac{1}{\mu}\left[\left(\frac{\mu}{N}\sum_{j=1}^{N}d_j(\mu)+O(\mu^2)\right)+O(\mu^2)\right]\right\}\\ &\quad = \lim_{\mu\to 0}\left\{\frac{1}{N}\sum_{j=1}^{N}d_j(\mu)+O(\mu)\right\}.\end{aligned} \tag{29}$$

Finally, taking the limit gives

$$\lim_{\mu\to 0}\left\{\frac{1}{\mu}\,\ln\left(\frac{1}{N}\sum_{j=1}^{N}\exp(\mu d_j(\mu))\right)\right\} = \frac{1}{N}\sum_{j=1}^{N}d_j^0 \tag{30}$$

□

Part b):
Using the identity $\ln(ab) = \ln(a) + \ln(b)$ gives

$$\lim_{\mu\to\infty}\left\{\frac{1}{\mu}\,\ln\left(\frac{1}{N}\right)+\frac{1}{\mu}\,\ln\left(\sum_{j=1}^{N}\exp(\mu d_j(\mu))\right)\right\} \tag{31}$$

but the left-hand term $\to 0$ as $\mu \to \infty$. Applying the Varadhan–Laplace lemma to the remaining expression yields (cf. [11], [12])

$$\lim_{\mu\to\infty}\left\{\frac{1}{\mu}\,\ln\left(\frac{1}{N}\sum_{j=1}^{N}\exp(\mu d_j(\mu))\right)\right\} = \max_j\,[d_j^\infty]. \tag{32}$$

□

It remains to show that this result ensures that the hybrid algorithm does in fact interpolate between ML state sequence estimation (obtained via either the above VFBA or the VA) and MAP state estimation (obtained via the HFBA). The proof of this has been split into two lemmas to aid presentation.

Lemma 1: The forward information probability measure $\kappa_t^\mu(i)$ limits to

a) $\lim_{\mu\to 0}\,\{\kappa_t^\mu\} \to \alpha_t(i)$—the forward information probability measure of the HFBA, and
b) $\lim_{\mu\to\infty}\,\{\kappa_t^\mu\} \to \delta_t(i)$—the forward probability measure of the VA (and the VFBA).

Proof:
Part a):
Initialize $\kappa_1^0(i) := \pi(i)$. Assume that

$$\kappa_{t-1}^0(i) = \lim_{\mu\to 0} \{\kappa_{t-1}^\mu(i)\}$$

exists and observe that

$$\kappa_t^0(i) := \lim_{\mu\to 0} \left\{ b_i(O_{\hat{t}}) \frac{(1+(N-1)e^{-\mu})}{\mu} \cdot \ln\left(\frac{1}{N}\sum_{j=1}^N [\exp(\mu\kappa_{t-1}^\mu(j)a_{ji})]\right)\right\}. \tag{33}$$

Taking the limit of the right-hand side, noting $\lim_{\mu\to 0}\{e^{-\mu}\} = 1$, and using Proposition 1, Part a) gives

$$\begin{aligned}\kappa_t^0(i) &= b_i(O_t)\sum_{j=1}^N \left[\lim_{\mu\to 0}\{\kappa_{t-1}^\mu(j)\}a_{ji}\right] \\ &= b_i(O_t)\sum_{j=1}^N [\kappa_{t-1}^0(j)a_{ji}].\end{aligned} \tag{34}$$

Thus by induction $\kappa_t^0(i)$ exists for all $t = 1,\cdots,T$ and, in fact, $\kappa_t^0(k) = \lim_{\mu\to 0}\ \kappa_t^\mu(i)$. Comparing (34) and (15) it follows that

$$\kappa_t^0(i) \equiv \alpha_t(i). \tag{35}$$

□

Part b):
Initialize $\kappa_1^\infty(i) := \pi(i)$. Assume that

$$\kappa_{t-1}^\infty(i) = \lim_{\mu\to\infty}\{\kappa_{t-1}^\mu(i)\}$$

exists and using the identity $\ln(ab) = \ln(a) + \ln(b)$ observe that

$$\kappa_t^\infty(i) := \lim_{\mu\to\infty}\left\{ b_i(O_t)(1+(N-1)e^{-\mu})\left[\frac{1}{\mu}\ln\left(\frac{1}{N}\right) + \frac{1}{\mu}\ln\left(\sum_{j=1}^N [\exp(\mu\kappa_{t-1}^\mu(j)a_{ji})]\right)\right]\right\}. \tag{36}$$

Taking limit of the right-hand side, noting that $\lim_{\mu\to\infty}\{e^{-\mu}\} = 0$, and using Proposition 1, Part b) gives

$$\begin{aligned}\kappa_t^\infty(i) &= \max_{1\le j\le N}\left[\lim_{\mu\to\infty}\{\kappa_{t-1}^\mu(j)\}a_{ji}\right]b_i(O_t) \\ &= \max_{1\le j\le N}[\kappa_{t-1}^\infty(j)a_{ji}]b_i(O_t).\end{aligned} \tag{37}$$

Thus by induction $\kappa_t^\infty(k)$ exists for all $t = 1,\cdots,T$ and, in fact, $\kappa_{\hat{t}}^\infty(i) = \lim_{\mu\to\infty}\{\kappa_{\hat{t}}^\mu(i)\}$.
Comparing (37) and (7) it follows that

$$\kappa_{\hat{t}}^\infty(i) \equiv \delta_t(i). \tag{38}$$

□

Lemma 2: The backward information probability measure $\tau_t^\mu(i)$ limits to

a) $\lim_{\mu\to 0}\{\tau_t^\mu\} \to \beta_t(i)$—the backward information probability measure of the HFBA, and
b) $\lim_{\mu\to\infty}\{\tau_t^\mu\} \to \tilde{\beta}_t(i)$—the backward probability measure of the VFBA.

Proof: The proof is analogous to that for Lemma 1. □

As a consequence of these results we show that the state estimates $\hat{q}_t^\mu$ obtained by the hybrid algorithm limits to those obtained by the HFBA and VFBA (and thus the VA) for $\mu \to 0$ and $\mu \to \infty$, respectively.

Theorem 1: The hybrid algorithm interpolates between the HFBA and the VFBA. That is,

a) $\hat{q}_t^{\text{HFBA}} = \lim_{\mu\to 0}\{\hat{q}_t^\mu\}$, and (39)

b) $\hat{q}_t^{\text{VFBA}} = \lim_{\mu\to\infty}\{\hat{q}_t^\mu\}$ (40)

are the state estimates that would be obtained via the HFBA and the VFBA (and hence the VA), respectively.

Proof:
Part a):
Taking the limit as $\mu \to 0$ of (24), using Lemmas 1 and 2, Part a) proves that $\theta_t^0(i) \equiv \gamma_t(i)$, and thus

$$\hat{q}_t^0 = \lim_{\mu\to 0}\{\hat{q}_{\hat{t}}^\mu\} \equiv \hat{q}_t^{\text{HFBA}}. \tag{41}$$

□

Part b):
Taking the limit as $\mu \to \infty$ of (24), using Lemmas 1 and 2, Part b) proves that $\theta_t^\infty(i) \equiv \tilde{\gamma}_t(i)$, thus

$$\hat{q}_t^\infty = \lim_{\mu\to\infty}\{\hat{q}_i^\mu\} \equiv \hat{q}_t^{\text{VFBA}} \equiv \hat{q}_t^{\text{VA}}. \tag{42}$$

□

Practicalities: Observe that since $\hat{q}_t^\mu$ is a hard decision from a finite set, then for sufficiently small $\mu \to 0, \hat{q}_t^\mu = \hat{q}_t^{\text{HFBA}}$ and similarly for sufficiently large $\mu \to \infty, \hat{q}_t^\mu = \hat{q}_t^{\text{VFBA}}$. The computational complexity and numerical implementation issues associated with the hybrid algorithm can be overcome using the Jacobian logarithm (cf. [20, eq. (16)]).

The hybrid algorithm provides a mathematical connection between the widely used Viterbi and HMM forward-backward estimation methods. By use of the tuning parameter, the hybrid algorithm may also prove useful in the development of adaptive algorithms. That is, algorithms which adaptively change from estimating ML sequences to estimating MAP states (or *vice versa*) without having to swap between two different algorithms instantaneously. The hybrid algorithm may also be used to obtain state estimates in which the rigid path constraints imposed via ML sequence estimation could be relaxed by varying degrees. However, determining practical applications where the hybrid estimates are meaningful and/or advantageous, is the subject of continuing research.

V. Conclusion

In this correspondence, a hybrid Viterbi/HMM forward-backward algorithm was developed. This algorithm provides a mathematical connection between maximum-likelihood sequence estimation, obtained via the Viterbi forward-backward algorithm (and thus the Viterbi algorithm), and maximum *a posteriori* probability state estimation, obtained via the classical HMM forward-backward algorithm.

Acknowledgment

The authors wish to thank the anonymous reviewers for their helpful comments and suggestions.

References

[1] L. E. Baum and T. Petrie, “Statistical inference for probabilistic functions of finite state Markov chains,” *Ann. Math. Stat.*, vol. 37, pp. 1554–1563, 1966.
[2] L. E. Baum and J. A. Egon, “An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology,” *Bull. Amer. Meteorol. Soc.*, vol. 73, pp. 360–363, 1967.
[3] L. E. Baum and G. R. Sell, “Growth functions for transformations on manifolds,” *Pac. J. Math.*, vol. 27, no. 2, pp. 211–227, 1968.

[4] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Stat.*, vol. 41, no. 1, pp. 164–171, 1970.

[5] L. E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, vol. 3, pp. 1–8, 1972.

[6] L. R. Rabiner and B. H. Juang, "An Introduction to Hidden Markov Models," *IEEE ASSP Mag.*, pp. 4–16, Jan. 1986.

[7] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–285, Feb. 1989.

[8] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.

[9] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *IEEE Globecom '89* (Dallas, TX, vol. 3, Nov. 1989), pp 47.1.1–47.1.7.

[10] B. Vucetic and Y. Li, "A survey of soft-output algorithms," in *Proc. Int. Symp. Information Theory and Tts Applications* (Sydney, Australia, Nov. 1994), pp. 863–867.

[11] J. S. Baras and M. R. James, "Robust and risk-sensitive output feedback control for finite state machines and hidden Markov models," *J. Math. Syst. Est. Contr.*, vol. 7, no. 3, pp. 371–374, 1997.

[12] M. I. Friedlin and A. D. Wentzell, *Random Perturbations of Dynamical Systems.* New York: Springer-Verlag, 1984.

[13] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, Mar. 1973.

[14] J. K. Omura, "On the Viterbi algorithm," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 177–179, Jan, 1969.

[15] V. Krishnamurthy and J. B. Moore, "On-line estimation of hidden Markov model parameters based on the Kullback-Leibler information measure," *IEEE Trans. Signal Processing*, vol. 41. pp. 2557–2573, Aug. 1993.

[16] M. V. Eyuboğlu and S. U. Qureshi, "Reduced-state estimation with set partitioning and decision feedback," *IEEE Trans. Commun.*, vol. 36, pp. 13–20, Jan. 1988.

[17] A. Duel-Hallen and C. Heegard, "Delayed decision-feedback sequence estimation," *IEEE Trans. Commun.*, vol. 37, pp. 428–436, May 1989.

[18] G. D. Brushe, V. Krishnamurthy, and L. B. White, "A reduced complexity on-line state sequence and parameter estimator for superimposed convolutional coded signals," *IEEE Trans. Commun.*, vol. 45, pp. 1565–1574, Dec. 1997.

[19] J. F. Hayes, T. M. Cover, and J. B. Riera, "Optimal sequence detection and optimal symbol-by-symbol detection: Similar algorithms," *IEEE Trans. Commun.*, vol. COM-30, pp. 152–157, Jan. 1982.

[20] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. IEEE Int. Conf. Communications '95* (Seattle, WA, June 1995), pp. 1009–1013.

[21] G. Fettweis and H. Meyr, "High-speed parallel Viterbi decoding: Algorithm and VLSI-architecture," *IEEE Commun. Mag.*, pp. 46–55, May 1991.

Correlation-Based Testing for the Convergence of Decision Feedback Equalizers

Kutluyıl Doğançay, *Member, IEEE*

***Abstract*— It has been thought that convergence tests developed for linear decision-directed equalizers based on the correlation matching criterion may not be directly applicable to decision feedback equalizers (DFE's). Indeed, the correlation-based convergence criteria proved for linear equalizers have so far been accepted as conjectural for DFE's. In this correspondence we show that identical channel input and DFE output autocorrelations is a necessary and sufficient condition for the channel eye to be open, if the channel input is binary phase-shift keyed. A threshold test based on the correlation matching criterion and assuming *a priori* knowledge of the channel input autocorrelation is also constructed and analyzed.**

***Index Terms*—Blind channel equalization, convergence tests, decision feedback equalizers, hypothesis testing, statistical threshold tests.**

I. Introduction

In this correspondence we extend the notion of correlation-based testing for the convergence of linear decision-directed equalizers (LDDE's) to the case of decision feedback equalizers (DFE's). Previously proposed off-line tests for the convergence of LDDE's to open-eye parameters [1], [2] are essentially predicated on a comparison between the channel input and (detected) equalizer output autocorrelation values. For binary phase-shift keyed channel inputs taking on values ± 1, the only *a priori* knowledge required by these tests is the autocorrelation of the channel input sequence. Testing for the convergence is, therefore, carried out in a blindfolded fashion (i.e., without any explicit knowledge of the channel input sequence). Attempts have been made to construct similar tests for DFE's in the past, but no formal proof has been available to show their viability [1].

The aim of this correspondence is to prove that the testing criterion of correlation matching which hitherto has been used to test for the convergence of LDDE's to open-eye parameters can be applied directly to DFE's without any modification. The proof is based on showing that if there are discrepancies between the channel input and equalizer output sequences, it is not possible for their autocorrelations to be identical. In this sense, the correspondence is an extension of [1] and [2] to DFE's and provides a proof for Conjecture 2 in [3] whilst extending its application to correlated channel inputs.

The correspondence is organized as follows. Section II gives a formal definition of the testing problem at hand and states our main result. In Section III we implement the test criterion as a statistical threshold test along the lines of [2] and provide a performance analysis. The effect of channel noise on the test criterion is considered in Section IV. Section V presents simulation examples which demonstrate the application and statistical properties of the test. Conclusions are drawn in Section VI.

Manuscript received April 21, 1997; revised February 7, 1998. This work was supported by the Australian Research Council (ARC) and the Cooperative Research Centre for Sensor, Signal and Information Processing (CSSIP).

The author was with the Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, Victoria 3052, Australia. He is now with the Institute of Information Sciences and Technology, Massey University, Palmerston North, New Zealand.

Publisher Item Identifier S 0018-9448(98)06904-1.