A COMPARISON OF APPROXIMATE VITERBI TECHNIQUES AND PARTICLE FILTERING FOR DATA ESTIMATION IN DIGITAL COMMUNICATIONS

Steffen Barembruch
Institut des Telecommunications/Telecom ParisTech
46 rue Barrault F-75634 Paris Cedex 13, France.

ABSTRACT

We consider trellis-based algorithms for data estimation in digital communication systems. We present a general framework which includes approximate Viterbi algorithms like the M-algorithm and the T-algorithm as well as particle filtering algorithms. The algorithmic concepts are very close, since the difference is simply the choice of the norm in the weights calculation. The general framework yields hence a new interpretation of these algorithms and may give rise to a series of new algorithms by using general selection schemes or a different choice for the norm. We show that the (approximate) expectation maximization Viterbi algorithm (EMVA) profits from using Chi Squared optimal selection compared to the standard EMVA.

Index Terms— Viterbi decoding, Monte Carlo methods, Deconvolution, Smoothing methods, Multipath channels

1. INTRODUCTION

We compare approximate Viterbi algorithms (AVAs) and particle filtering algorithms (PFAs) for symbol recovery and for blind identification via the EM algorithm in discrete hidden Markov models (HMM). These two groups of algorithms stem from basically different concepts, i.e. the AVAs intend to calculate the maximum-a-posteriori (MAP) estimate of the data, whereas PFAs approximate the filtering probabilities and apply as well to non-discrete state-space models. We will provide a general framework including both these algorithms.

A lot of research has been devoted to the Viterbi algorithm (VA) [1] which is the most efficient method to derive the MAP estimate of the data. It has been used in speech processing [2] and many other applications. Several approximations like the M-algorithm [3] and the T-algorithm [4] have been described to reduce the computational complexity of the VA.

On the other hand, particle filtering approximations have already proved to be useful in many general HMM problems and in many digital communication scenarios. In a finite state space, the plain particle estimation algorithms may be significantly improved, since it is possible to consider all the potential offsprings of a given particles, i.e. a proposal distribution as in standard particle filtering is not needed. In this context, the PFAs reduce the computational complexity of the filtering part of the Baum–Welch algorithm [5].

If the channel is known, then the algorithms are used for data recovery. However for blind identification, they implement as well the expectation step of an EM algorithm or the EMVA [6]. We show that with the help of this framework the EMVA may be easily changed using Chi Squared optimal selection [7]. We show in simulation how much the EMVA gains from this adaptation.

2. MODEL

Let $\mathcal{X}$ be the alphabet of a linear modulation of size $m = |\mathcal{X}|$ and let the symbols $a_k \in \mathcal{X}$ be independently and uniformly distributed in $\mathcal{X}$.

Let $h = (b_0, \ldots, b_{L-1})^T$ denote the coefficients of the finite impulse response of the frequency selective channel of order $L$. The observation sequence is given by:

$$y_k = \sum_{i=0}^{L-1} a_{k-i} b_i + \varepsilon_k, \quad \varepsilon_k \sim \mathcal{N}_C(0, \sigma^2),$$

where $\varepsilon_k$ is independent complex Gaussian noise of variance $\sigma^2$. The same model in matrix notation to obtain a HMM writes:

$$S_k = QS_{k-1} + [a_k, 0, \ldots, 0]^T,$$
$$y_k = h^T S_k + \varepsilon_k, \quad \varepsilon_k \sim \mathcal{N}_C(0, \sigma^2),$$

where $S_k = [a_{k-1}, \ldots, a_k, 0_{L-1}]^T \in \mathbb{R}^L$ denotes the $L$ current symbols. $Q$ is a shift matrix, which is zero except for the elements underneath the main diagonal.

The elements of the state space are denoted $S = \{s_0, \ldots, s^{M-1}\}$. A lower index always indicates the time step(s), while an upper index indicates an order on the state space or the index of a selected position. The state space has size $M = m^L$. Let $q(s, s')$ denote the transition probability from $s$ to $s'$ for $s, s' \in S$. The likelihood of observation $y_k$ given $S_k = s$ is denoted $g(s, y_k)$. For $k \leq j$, we use the notation $S_{k:j} = (S_k, \ldots, S_j)$ and accordingly for the other quantities, as well as $\pi_{s_{k:j}|s_{i-1}}(\cdot) = P(S_{k:k'} = \cdot | y_{0:i} = y_{0:i})$ for $i \leq n$ and $k \leq k' \leq n$. Let $y_{0:n} = (y_0, \ldots, y_n)$ be a given observation sequence of length $n+1$.

The marginal filtering probabilities $\pi_{k|k}(s) = P(S_k = s|y_{0:k})$ for $s \in S$ may be decomposed into

$$\pi_{k|k}(s) \propto \sum_{s' \in S} \pi_{k|k-1}(s, s') g(s, y_k),$$

where $\pi_{k|k-1}(s, s') = \pi_{k-1|k-1}(s') q(s', s)$.
3. APPROXIMATE ALGORITHMS

We focus on methods that reduce the complexity of the Viterbi or the Baum-Welch algorithm by exploiting only a part of the state space. We will first present a general framework including the approximate Viterbi algorithms as well as particle filtering methods, then we will place each method into this framework.

3.1. General Framework

At each time step \( k \), we store a set of \( N \) positions \( s_k^i \in S \) considered to be a representative sample of \( S \) at time \( k \). They are assigned probability weights \( w_k^i \), whose interpretation depends on the specific algorithm. If necessary, the information \( a_k^i \) on the most probable predecessor is also stored, leading to the system \( (s_k^i, w_k^i, a_k^i)_{i \in S} \).

**Algorithm 1**: Marginal framework, update step

| Input: \( (s_k^i, w_k^i, a_k^i)_{i \in \{0, \ldots, N_k-1\}} \) |
| Output: \( (s_{k+1}^i, w_{k+1}^i, a_{k+1}^i)_{i \in \{0, \ldots, N_{k+1}-1\}} \) |

- **Exploitation step**
  
  for \( j \in \{0, \ldots, M-1\} \) (loop over \( S \)) do
  
  \[ w_{k+1}^j = \left\| \left( w_k^j q(s_k^j, s^j), \sum_{i \in S} g(s^j, y_{k+1}) \right) \right\|_p \]  

  Optional: \( \tilde{a}_{k+1}^j = \arg \max_{i \in S} \left( w_k^j q(s_k^j, s^j) \right) \)

  end

- **Selection step**

  | Input: \( (s^j, \tilde{w}_{k+1}^j)_{j \in \{0, \ldots, M-1\}} \) |
  | Output: \( N_{k+1} \) positions \( s_{k+1}^i \) with weights \( w_{k+1}^i \) (and \( a_{k+1}^i \))

Updating to time step \( k+1 \) is done in two steps. The first one is the exploitation step, where a weight for each state in \( S \) is calculated, followed by a selection step, in which a part of the states is neglected. The schematics of this iterative procedure may be seen in Fig. 1.

The weight calculation of the VA and of the Baum-Welch algorithm which corresponds to Decomposition (1) are very similar. While the latter is averaging over the look-ahead probabilities \( \pi_{k|k-1} \), the VA is maximizing over them. We may interpret the averaging as taking the \( \ell_1 \)-norm of \( \pi_{k|k-1} \) and the maximizing as \( \infty \)-norm. This may be generalized to any \( p \)-norm for \( p \geq 1 \). The algorithm for a fixed \( p \geq 1 \) is given in Alg. 1.

Observe that the weights calculation in (2) is an approximate analog of the VA update if we choose \( p = \infty \) and of (1) if we choose \( p = 1 \). The displayed calculations allow for transformations (e.g. operating in the log-domain) reducing the computational cost.

Several procedures have been developed to implement the selection step, e.g.

1. **Best-Weights Selection** [8] is a purely deterministic scheme keeping the \( N \) highest weights.

2. **Threshold-Comparison Selection** [4] keeps the weights which are larger than a certain fraction \( T \) of the maximal weight.

3. **L2-Optimal Random Selection** [9] minimizes the expected \( L_2 \) distance of the selected weights to the original weights.

4. **Chi-Square-Optimal Random Selection** [7] minimizes the expected Chi-Square distance.

3.2. Approximate Viterbi Algorithms: M-Algorithm and T-Algorithm

The approximate Viterbi algorithms (AVAs) reduce the complexity of the original VA by neglecting a certain part of the states after each time step. They may be considered as marginal algorithms since at time step \( k \) a weight \( \tilde{w}_k^j \) is assigned to each single state \( s_j \) for \( j < M \), and not to the trajectory, although the weights describe scores of the best paths to these states. The history of the best path to each state is therefore stored in the auxiliary information \( \tilde{a}_k^j \).

The AVAs fit therefore exactly into the marginal general framework if we choose \( p = \infty \). After having reached the final time step \( n \), the best path is found by backtracking through the auxiliary information variables.

The difference between the M-Algorithm [3] and the T-Algorithm [4] is the selection procedure. The first one employs the Best-Weights Selection, while the latter one the Threshold-Comparison Selection. Furthermore, it is also possible to employ the random selection schemes, which has never been considered so far. In this case, it is however not longer possible to operate in the log-domain, because the random selection schemes work with the (normalized) probability weights.

3.3. Particle Filtering

Marginal PFAs estimate the distribution \( \pi_{k|k} \) and are thus an adaptation of the Baum-Welch filtering algorithm. The update procedure from time step \( k \) to \( k+1 \) is based on Decomposition 1. We only consider the current state and the weight, the past of each particle is irrelevant. Therefore, \( a_k^i \) is not needed. Since we want to use Decomposition 1 to approximate the filtering probabilities, we
choose \( p = 1 \) in Alg. 1, such that the norm averages the predictive probabilities, instead of taking the maximal weight of the predecessors as in the AVAs. This turns out to be the only algorithmic difference between them.

Having conducted Alg. 1 to obtain \((\xi_k, w_k^i)_{k=0}^{n} \) for each time step \( k \leq n \), the marginal filtering distributions \( \pi_{k|n} \) are approximated by

\[
\hat{\pi}_{k|n}(s) = \left( \Omega_k \right)^{-1} \sum_{i=0}^{N_k-1} w_k^i \delta_{\xi_k^i}(s) \tag{3}
\]

for each \( s \in \mathcal{S} \) and where \( \Omega_k = \sum_{i=0}^{N_k-1} w_k^i \) and for each \( s, s' \in \mathcal{S} \) we define \( \delta_s(s') = 1 \) if \( s = s' \) and 0 otherwise.

### 4. SMOOTHING TECHNIQUES AND APPROXIMATE EM ALGORITHM

The previously described algorithms may also help to approximate the smoothing distributions \( \pi_{k|n} \), which may improve the data estimation or serve to estimate unknown parameters via the EM algorithm.

The EMVA [6] adapts this EM Algorithm by approximating the smoothing probabilities \( \pi_{0:n|n}(s_{0:n}) \) of the paths \( s_{0:n} \in \mathcal{S}^{n+1} \) by

\[
\hat{\pi}_{0:n|n}(s_{0:n}) = \Omega^{-1} \sum_{i=0}^{M} w^i \delta_{\xi^i_{0:n}}(s_{0:n})
\]

where \( \xi^i_{0:n} = (\xi^i_0, \ldots, \xi^i_n) \) denotes the \( i \)-th survivor path of the VA, i.e. the shortest path to the \( i \)-th symbol at time \( n \). It is assigned a probability weight proportional to

\[
w^i = g(\xi_0^i, y_0) \prod_{k=0}^{n-1} g(\xi_k^i, s_{k+1}^i) g(\xi_{k+1}^i, y_{k+1}) \tag{4}
\]

The normalization constant is given by \( \Omega = \sum_{i=0}^{M} w^i \). The marginals \( \pi_{k|n} \) are then approximated by marginalization, which risks strong degeneracy for \( k \ll n \). If necessary, the authors suggest to reduce the complexity of the EMVA by replacing the VA with the M-algorithm. As mentioned before, the AVA may as well be coupled with one of the random selection schemes.

There are as well several smoothing approximations based on particle filtering [7]. We use here the fixed-interval smoothing approximation [10].

### 5. COMPUTATIONAL RESULTS

For data recovery with the channel assumed known, we compare the approximate Viterbi algorithms (AVA) - the M-algorithm when using the Best-Weights selection and the T-algorithm for the Threshold selection - the marginal Particle-Filtering Algorithm (mPFA) and a joined Particle-Filtering Algorithm (jPFA) [10] in terms of the symbol error rate (SER). The model of a linear modulation allows only \( m \) offsprings at time \( k+1 \) for each position at time \( k \), such that the complexity of the updating step is reduced to \( O(N_k m^k) \) instead of \( O(N_k m^k) \) for general models. Furthermore, because of the specific structure, the complexity of the Fixed-Interval Smoothing (FIS) is not higher than that of the AVAs or the PFAs [7]. We take it hence as a comparison, together with a Block CMA. The symbols are sent over a channel consisting of three taps with delays \([0T, 0.9T, 3.2T] \), where \( T \) is the symbol period. The attenuations were randomly chosen. A channel order \( L = 5 \) was sufficient to cover the relevant peaks in the corresponding channel impulse response. We fixed the length of the observations to \( n = 2000 \).

![Fig. 2](image_url) Symbol error rate, 16-QAM with \( N = 250 \) particles and Best-Weights Selection

The model PFAs is not able to recover the symbols, since the marginal filtering distributions only involve information on the past observations but not on the future. However, the marginal smoothing algorithm, which is based on these marginal filtering distributions plus a backward weight correction, yields - together with the AVA - the lowest SER. The joined PFA is slightly inferior.

<table>
<thead>
<tr>
<th>Alg.</th>
<th>Selection</th>
<th>Best W.</th>
<th>Threshold</th>
<th>L2</th>
<th>Chi Sq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVA</td>
<td></td>
<td>96.8%</td>
<td>98.0%</td>
<td>95.8%</td>
<td></td>
</tr>
<tr>
<td>mPFA</td>
<td></td>
<td>94.6%</td>
<td>98.6%</td>
<td>95.2%</td>
<td></td>
</tr>
<tr>
<td>jPFA</td>
<td></td>
<td>94.8%</td>
<td>75.0%</td>
<td>98.8%</td>
<td></td>
</tr>
<tr>
<td>FIS</td>
<td></td>
<td>94.6%</td>
<td>88.6%</td>
<td>93.0%</td>
<td></td>
</tr>
<tr>
<td>Block CMA</td>
<td></td>
<td>100%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Probability of \( SER < 0.01 \), SNR 20dB, 16-QAM, for different particle sizes \( N \)

Fig 2 shows the performance of the different algorithms coupled with the Best-Weights Selection for the 16-QAM. The marginal PFA is not able to recover the symbols, since the marginal filtering distributions only involve information on the past observations but not on the future. However, the marginal smoothing algorithm, which is based on these marginal filtering distributions plus a backward weight correction, yields - together with the AVA - the lowest SER. The joined PFA is slightly inferior.

Table 1 presents the percentages of Monte Carlo runs for which the final MSE was smaller than 0.01. The SNR is 20dB, while the number of particles varies \( (N = 100, 250) \). The threshold selection appears to be slightly inferior to the other selection schemes. Apart from the marginal PFA (mPFA) the algorithms are almost equivalent.

We now turn to blind identification of the unknown channel and evaluate the mean-squared error (MSE) performance of the EMVA (with beam-search) compared to the EMVA where the Beam-search has been replaced by an AVA with a random selection scheme. The initial channel estimates are chosen according to the BCA.
initialization method [6]. We simulate the data directly from the model in 2 with $L = 3$ and $n = 300$. For each Monte-Carlo run, the channel coefficients are drawn randomly from a uniform distribution. Fig. 3 shows the median over 500 Monte-Carlo runs of the MSE of the channel coefficients from the first iteration up to 50 iterations for a 16-QAM model with channel order $L = 3$ and an SNR of 16dB.

The difference between the $L^2$- and the Chi-Square-optimal selection is due to the fact, that in the $L^2$ selection all the weights $\leq \lambda$ are resampled and assigned the same new weight $\lambda$. Especially during the first iterations, the filtering distributions are flat, i.e. often very few weights are larger than $\lambda$. Hence, the remaining particles often have the same weight. In the update update to the next time step in 2, the maximization is then over a set of equal weights. The Chi-Square optimal selection is better adapted, since it assigns a weight proportional to their square root. These are hence never equal.

The Chi-Square selection is superior to the original EMVA (with Best-Weights selection) since during the EM, the states with high smoothing probabilities often have small filtering probabilities. In contrast to the random selection schemes the Best-Weights selection will never select these states. Fig. 4 shows that this effect gets even more important in a 64-QAM model. Both random schemes become clearly superior now.

6. CONCLUSION

We have presented a flexible algorithmic framework covering approximate Viterbi algorithms (VA) and particle-filtering algorithms (PFA). They are conceptually similar, although their estimation purpose is unlike. The difference is the choice of the norm. In this respect, alongside to $p = 1$ and $p = \infty$, this gives rise to a new, more general class of algorithms by choosing $p \geq 1$. It is clear that each algorithm, e.g. the approximate VA, may be coupled with each selection scheme, leading to a modified EMVA that outperforms the standard approximate EMVA.

7. REFERENCES


