Sphere-Constrained ML Detection for Channels with Memory

Haris Vikalo[†], Babak Hassibi[†], and Urbashi Mitra[‡]
[†]California Institute of Technology, Pasadena CA 91125
[‡]University of Southern California, Los Angeles CA 90089

Abstract

The maximum-likelihood (ML) detection problem for channels with memory is investigated. The Viterbi algorithm (VA) provides an exact solution. Its computational complexity is linear in the length of the transmitted sequence but exponential in the channel memory length. Hence, the VA can be computationally inefficient when employed for detection on long channels. On the other hand, the sphere decoding (SD) algorithm also solves the ML detection problem exactly and has expected complexity which is polynomial (often cubic) in the length of the transmitted sequence over a wide range of signal-to-noise ratios (SNR). We combine the sphere-constrained search strategy of SD with the dynamic programming principles of the VA. The resulting algorithm has the worst-case complexity of the VA, but often significantly lower expected complexity.

1 Introduction

We consider the frequency-selective channel model, with input/output relation given by

$$x_i = \sum_{j=1}^{l} h_j s_{i-j} + v_i,$$

where $h_i, i = 1, \dots, l$ are the coefficients of the channel impulse response, l denotes the channel length, s_i is the i^{th} symbol in the transmitted sequence chosen from an L-PAM constellation $\mathcal{D}_L$, and v_i denotes a Gaussian noise sample $\mathcal{N}(0, \sigma^2)$. Over a horizon of length T, the ML sequence detector minimizes the cost function

$$C_T = \sum_{j=1}^{T} |x_j - \sum_{m=1}^{l} h_m s_{j-m+1}|^2 \tag{1}$$

to find the most likely transmitted symbol sequence $\{s_1, s_2, \dots, s_T\}$. The Viterbi algorithm (VA) [1] finds the sequence which minimizes C_k by using dynamic programming ideas [2]. Typically, the VA is conveniently employed as a breadth-first search on a trellis, a directed graph describing systems with memory illustrated in Figure 1. The key observations is that C_T can be recursively computed as

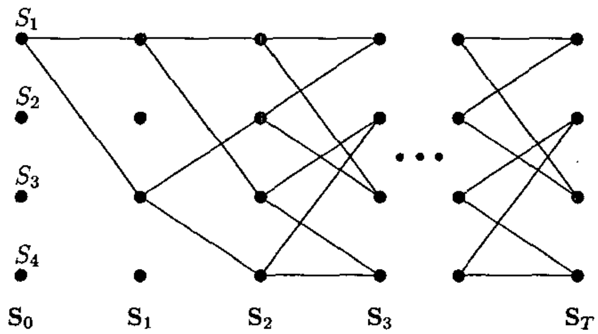

Figure 1. *Trellis illustration*

$$C_{k+1} = C_k + |x_{k+1} - \sum_{j=1}^{l} h_j s_{k-j+1}|^2, \tag{2}$$

$k = 0, \dots, T-1$, $C_0 = 0$. Clearly, the second term on the right-hand side (RHS) of (2) does not depend on $s_{k-l}, \dots, s_1$ but only on the current symbol s_k and the current memory of the channel $s_{k-1}, \dots, s_{k-l+1}$. The L^{l-1} possible states of the channel memory comprise the state set $\mathcal{S}_k$ (see Figure 1). To find the smallest cost path to the j^{th} state in $\mathcal{S}_{k+1}$, denoted by $\mathcal{S}_{k+1}^{[j]}$, it is sufficient to consider all possible state transitions to $\mathcal{S}_{k+1}^{[j]}$ along the L branches emanating from the states in set $\mathcal{S}_k$. This procedure can be done recursively. The trellis path of length T that has the smallest cost C_T is the optimal path. The signal sequence that corresponds to the branch transitions along the optimal trellis path is the solution to the ML detection problem. The complexity of the Viterbi algorithm is proportional to the number of states and thus grows exponentially with the length of the channel. On the other hand, it is linear in the

[†]This work was supported in part by the NSF under grant no. CCR-0133818, by the Office of Naval Research under grant no. N00014-02-1-0578, and by Caltech's Lee Center for Advanced Networking.

0-7803-8104-1/03/$17.00 ©2003 IEEE

length of the data sequence.

The SD algorithm [3] can also be used to perform finite-length sequence ML detection on channels with memory [4]. To employ SD, we need to write the channel model as $\mathbf{x} = H\mathbf{s} + \mathbf{v}$, where $\mathbf{s} = [s_1\ s_2\ \dots\ s_T]'$ is the vector of transmitted data sequence, and $\mathbf{v} = [v_1\ v_2\ \dots\ v_{T+l-1}]'$ is the vector of additive white Gaussian noise. The Toeplitz matrix $H \in \mathcal{R}^{(T+l-1)\times T}$ is given by

$$H = \begin{bmatrix} h_1 & & & \\ \vdots & \ddots & 0 & \\ h_l & \dots & h_1 & \\ & \ddots & & \ddots \\ & & h_l & \dots & h_1 \\ & 0 & & \ddots & \vdots \\ & & & & h_l \end{bmatrix}. \quad (3)$$

The ML detection can now be expressed as an integer least-squares problem,

$$\min_{\mathbf{s}\in\mathcal{D}_L^T} \|\mathbf{x} - H\mathbf{s}\|^2. \quad (4)$$

This problem has a geometric interpretation: given a point $\mathbf{x}$, find the closest lattice point in a skewed lattice $H\mathbf{s}$. The SD algorithm solves (4) by performing a search over only those points $H\mathbf{s}$ that belong to a sphere around $\mathbf{x}$. The radius r of the sphere is chosen so that we find a point inside the sphere with a high probability. In particular, note that $\|\mathbf{x} - H\mathbf{s}\|^2 = \|\mathbf{v}\|^2 = v_1^2 + \dots + v_T^2$ is a chi-square random variable with $2T$ degrees of freedom. Thus the radius $r^2 = \alpha T\sigma^2$ can be chosen probabilistically so that

$$\int_0^{\alpha T} \frac{\lambda^{T-1}}{\Gamma(T)} e^{-\lambda} d\lambda = 1 - \epsilon, \quad (5)$$

where $\epsilon \ll 1$. The condition that a point $H\mathbf{s}$ belongs to the sphere of radius r is given by

$$r^2 \geq \|\mathbf{x} - H\mathbf{s}\|^2. \quad (6)$$

The summation on the RHS of (6) can be expanded to yield a set of conditions on the components of $\mathbf{s}$,

$$(x_1 - h_1 s_1)^2 \leq r_1^2,\ \ (x_2 - h_1 s_2 - h_2 s_1)^2 \leq r_2^2,$$

$$(x_3 - h_1 s_3 - h_2 s_2 - h_3 s_1)^2 \leq r_3^2, \quad \text{etc.},$$

where $r_1^2 = r^2, r_2^2 = r_1^2 - (x_1 - h_1 s_1)^2$, and so on. Note that this gives T conditions on the components of $\mathbf{s}$ which are necessary but still not sufficient. Only if the additional constraint,

$$\begin{aligned} r_{T+1}^2 \ \geq\ & (x_{T+1} - h_l s_{T-l+2} - \dots - h_2 s_T)^2 \\ & + \dots + (x_{T+l-1} - h_l s_T)^2, \end{aligned}$$

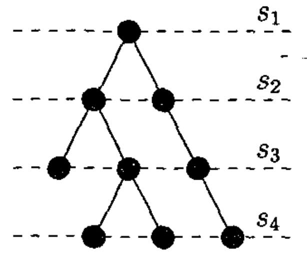

Figure 2. *Tree search of the sphere decoding algorithm.*

where $r_{T+1}^2 = r^2 - \sum_{j=1}^{T}(x_l - \sum_{m=1}^{l} h_m s_{j-m+1})^2$, is satisfied, does the point s indeed belong to the sphere, i.e., satisfies condition (6). The SD algorithm performs a depth-first search on a tree, as illustrated in Figure 2. A trace leading to a surviving node on the k^{th} level of the tree corresponds to a vector $[s_1 \dots s_k]'$ inside the k-dimensional sphere. With the probabilistic choice of r, the computational complexity of the SD algorithm is a random variable [5], with the mean often significantly below the complexity of the VA [4].

2 Combining sphere decoding and Viterbi algorithm

Complexity of the VA is linear in the length of the data sequence but is exponential in the channel memory size, where the base of the exponent is the symbol alphabet size. Thus for long channels and/or large symbol alphabets, the VA is often inefficient and occasionally non-feasible. On the other hand, over a wide range of SNR, expected complexity of SD is polynomial in the data block length and the degree of the polynomial does not vary significantly with the channel memory size. However, the SD algorithm does not exploit special structure of the channel matrix and has exponential worst-case complexity. Therefore, a hybrid receiver structure that combines the constrained search strategy of SD with the trellis based decoding of the VA, is desired. This can be obtained by either modifying the SD algorithm to include the channel memory state constraints or by adding the sphere constraints to the trellis search of the VA. The two approaches, essentially equivalent although one is depth-first and the other breadth-first, are briefly described here.

Algorithm 1 [SD modified with VA]: Consider the SD algorithm and the tree search illustrated in Figure 1. The SD algorithm does not account for the banded Toeplitz structure of the lattice generating matrix (3). We propose the following modification: assume that the algorithm is currently examining a node on the k^{th} level of the tree. Based on the current, and on up to $l-2$ tree nodes on levels $k-1, k-2, \ldots, k-l+1$ along which the algorithm descended to the current node, we identify the current state $S_k^{[j]} = s_k s_{k-1} \ldots s_{k-l+1}$. The meaning of the state is as it is on the trellis, i.e., it is the current state of the channel memory and we assign a cost $\mathcal{C}_k(S_k^{[j]})$ to it. By writing out the recursion for r_k^2, it is easy to see that

$$\mathcal{C}_k(S_k^{[j]}) = r^2 - r_{k+1}^2.$$

Now, in addition to the standard steps that the SD algorithm undertakes, it also compares this $\mathcal{C}_k(S_k^{[j]})$ with the previously stored minimum cost $\min \mathcal{C}_k(S_k^{[j]})$. If the current $\mathcal{C}_k(S_k^{[j]})$ is greater than $\min \mathcal{C}_k(S_k^{[j]})$, the algorithm prunes the tree, i.e., it discards the current tree node. If the current $\mathcal{C}_k(S_k^{[j]})$ is smaller than the previously stored $\min \mathcal{C}_k(S_k^{[j]})$ (or there are no previously stored $\min \mathcal{C}_k(S_k^{[j]})$), the algorithm assigns $\min \mathcal{C}_k(S_k^{[j]}) := \mathcal{C}_k(S_k^{[j]})$ and proceeds with the other SD steps. Note that the algorithm is still depth-first. Clearly, its complexity will be lower than the complexity of the original SD.

■

Algorithm 2 [VA modified with SD]: Consider the trellis representation of a frequency-selective channel and a finite data block transmission. We impose the constraint (6) that the transmitted signal belongs to a sphere of radius r defined by (5). As we have shown in the previous section, an obvious necessary condition that the transmitted signal needs to satisfy is given by $(x_1 - h_1 s_1)^2 \leq r_1^2$. However, from (1), this condition is equivalent to the constraint $\mathcal{C}_1(S_k) \leq r_1^2$. Similarly, condition $(x_2 - h_1 s_2 - h_2 s_1)^2 \leq r_2^2$ is equivalent to the constraint $\mathcal{C}_2 \leq r_2^2$. In general,

$$\mathcal{C}_k(S_k^{[j]}) \leq r_k^2, \quad k = 1, 2, \ldots, T, \; j = 1, 2 \ldots, L^{l-1}. \tag{7}$$

On the trellis, condition (7) means that the cost $\mathcal{C}_k$ should, for each state and time index k, be smaller than the radius of the sphere. The states $S_k^{[j]}$ that violate condition (7) can be neglected, i.e., no branches emanating from such states need to be considered when searching for the optimal trellis path. Therefore, the search on trellis can, on average, be performed faster that the Viterbi algorithm. The worst case complexity, on the other hand, coincides with the complexity of the Viterbi algorithm. The sphere-constrained trellis search is illustrated in Figure 3.

■

The following points are worthy of mention. Algorithm 2 is employed on the trellis and essentially reduces complexity of the VA by discarding states which violate certain (sphere) constraints. Hence this algorithm can be counted among the reduced-state detection algorithms (for

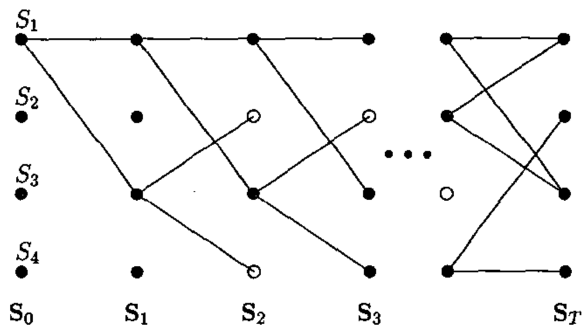

Figure 3. *Sphere-constrained search on trellis*

some recent results see, e.g., [6] and the references therein). However, the combined VA and SD on trellis does not sacrifice performance, as many state-reduced algorithms do, but rather solves the ML detection problem exactly (though this may require increasing the sphere radius if no point is found inside). Also, note that the VA can be employed for sequence detection by forcing the detector to make decisions once sufficiently deep inside the trellis (common heuristic suggests that 5 times the channel length is sufficiently deep). Algorithm 2 can be slightly modified to employ the same heuristic. In this case, one can think of a sliding window (or a "sliding sphere") of length (dimension) $5l$ that imposes a sphere constraint of the form (6) on the states of the trellis.

3 Expected complexity of the combined VA/SD algorithm

In this section, we consider the sphere-constrained modification of the VA and, using the approach originally proposed in [5]-[4], analytically find its expected complexity. Clearly, the expected complexity of the combined algorithm is proportional to the expected number of the states that survive the pruning process,

$$E_c = \sum_{d=1}^{T} \underbrace{E(\#\text{ of states that survive at dimension } d)}_{E_s(d)} \cdot (\#\text{ of flops per state}).$$

Since all states are equally likely, the expected number of surviving states in each dimension is given by

$$E_s(d) = n_s(d) \cdot P(\mathcal{C}_d^{[1]} = 00\ldots0 \text{ survives at dimension } d) \tag{8}$$

where for $d \geq l-1$ (i.e., once the memory "fills"), the number of states is $n_s(d) = L^{l-1}$. In what follows, we determine $E_s(d)$ for $d \geq l-1$. [For the other case ($d < l-1$), $E_s(d)$ can be either found similarly or simply approximated by the total number of states. This will be a good approximation since early in the trellis/tree there is not much pruning and most of the states survive.] To this end, the approach used to calculate the expected complexity of the SD algorithm in [5]-[4] needs to be modified so that the Toeplitz structure of matrix H is taken into account. [Note that the stand-alone SD (with no VA modification which we propose) only exploits banded but does not make use of Toeplitz structure of H.] We start by finding the probability that a state at dimension d survives the pruning. Consider the following thought experiment: assume that the sequence $\mathbf{s}_t$ of length d was transmitted and that $\mathbf{x} = H\mathbf{s}_t + \mathbf{v}$ is observed. We wish to determine the probability that for an arbitrary sequence $\mathbf{s}_a$ of length d, it holds that

$$\begin{aligned} 0 &\leq r^2 - \|\mathbf{x} - H_d\mathbf{s}_a\|^2 = r^2 - \|\mathbf{v} + H_d(\mathbf{s}_t - \mathbf{s}_a)\|^2 \\ &= r^2 - \|\mathbf{v} + (\mathbf{S}_t - \mathbf{S}_a)\mathbf{h}\|^2 = \xi, \end{aligned} \tag{9}$$

where the $d \times d$ matrix H_d and the l-dimensional vector $\mathbf{h}$ are given by

$$H_d = \begin{bmatrix} h_1 & & & & 0 & \\ h_2 & h_1 & & & & \\ \vdots & \vdots & \ddots & & & \\ h_l & \dots & & h_1 & & \\ & \ddots & & & \ddots & \\ 0 & & h_l & \dots & & h_1 \end{bmatrix},$$

$$\mathbf{h} = [h_1 \; h_2 \; \dots \; h_l]'.$$

Furthermore, the structure of the sequence vectors $\mathbf{s}_t$ and $\mathbf{s}_a$ is of the form

$$\mathbf{s}_t = [\underbrace{s_{t,1} \; \cdots \; s_{t,d-l}}_{\mathbf{s}_{t,past}} \,|\, \underbrace{s_{t,d-l+1} \; \cdots \; s_{t,d}}_{\mathbf{s}_{t,state}}]',$$

$$\mathbf{s}_a = [\underbrace{s_{a,1} \; \cdots \; s_{a,d-l}}_{\mathbf{s}_{a,past}} \,|\, \underbrace{s_{a,d-l+1} \; \cdots \; s_{a,d}}_{\mathbf{s}_{a,state}=00\ldots0}]'.$$

while the $d \times l$ dimensional matrices $\mathbf{S}_t$ is given by

$$\mathbf{S}_t = \begin{bmatrix} s_{t,1} & & 0 \\ \vdots & \ddots & \\ s_{t,d-l+1} & \cdots & s_{t,d-2l} \\ \vdots & & \vdots \\ s_{t,d} & \cdots & s_{t,d-l+1} \end{bmatrix},$$

and where

$$\mathbf{S}_a = \begin{bmatrix} s_{a,1} & & 0 \\ \vdots & \ddots & \\ s_{a,d-l+1} & \cdots & s_{a,d-2l} \\ & \ddots & \vdots \\ 0 & & s_{a,d-l+1} \end{bmatrix}.$$

To simplify the expressions, denote $\Lambda = \mathcal{S}_t - \mathcal{S}_a$. The characteristic function of ξ is given by

$$\Phi(\omega) = \int_{-\infty}^{\infty} p(\xi)e^{-j\omega\xi}d\omega = \mathrm{E}[e^{-j\omega\xi}],$$

The vector $[\mathbf{v} \;\; \mathbf{h}]'$ in (9) is Gaussian and thus we can obtain (see [4])

$$\Phi(\omega) = \frac{e^{j\omega r^2}}{(1 + j\omega\sigma^2)^{T-1}\prod_{k=1}^{l}[1 + j\omega(\sigma^2 + \rho_k)]},$$

where $\rho_k, k = 1, \dots, l$ are the eigenvalues of the matrix $\Lambda^*\Lambda$. Thus we obtain the desired probability as

$$\begin{aligned} &P(\mathcal{C}_d^{[1]} = 00\ldots0 \text{ survives at dimension } d) = \\ &\frac{1}{L^{l-1}} \frac{1}{L^{2(d-l+1)}} \sum_{\mathbf{s}_{t,state}} \sum_{(\mathbf{s}_{t,past};\mathbf{s}_{a,past})} \mathcal{F}^{-1}[\Phi(\omega)]. \end{aligned} \tag{10}$$

Expression (10) essentially states how the probability that an all-zero state survives the pruning is equal to the probability that an arbitrary sequence terminated in the all-zero state belongs to a sphere around the transmitted sequence, averaged over all possible transmitted sequences. The outer summation in (10) is performed over states in which the transmitted sequence may terminate (all L^{l-1} of them). The inner summation is performed over all possible pairs $(\mathbf{s}_{t,past}; \mathbf{s}_{a,past})$. An efficient enumeration (similar in spirit to the one proposed in [5] in the context of multi-antenna systems) that might ease the computation of the inner sum and possibly result in a closed-form expression for the probability, so far appears hard to obtain. Thus we leave (10) in its current form to be used numerically in evaluating the expected complexity (8).

4 Simulation results

We consider a channel of length $l = 6$, transmitting 4-PSK modulated ($L = 4$) data in blocks of length $T = 12$ at $SNR = 16$dB, and employ Algorithm 2 for ML detection on receiver. Figure 4 shows the empirical distribution of the complexity exponent, defined as $e = \log_T F$, where F denotes the total number of operations (flop count) performed when detecting s.

As evident from Figure 4, the complexity exponent is always smaller than the complexity exponent corresponding to the VA (denoted by the vertical dashed line).

Figure 5 shows the expected complexity exponent as a function of SNR. The expected complexity is roughly cubic in the considered SNR range.

5 Summary and conclusion

We proposed combining the sphere-constrained search of the SD and the dynamic programming principles of the

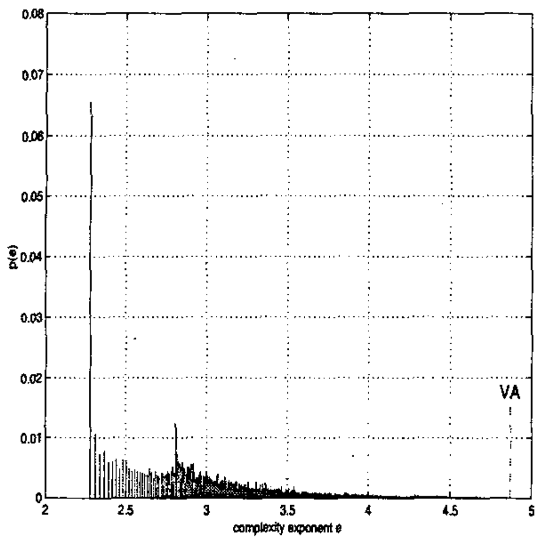

Figure 4. *Distribution of complexity exponent, $l = 6$, $T = 12$, $L = 4$, $SNR = 16dB$.*

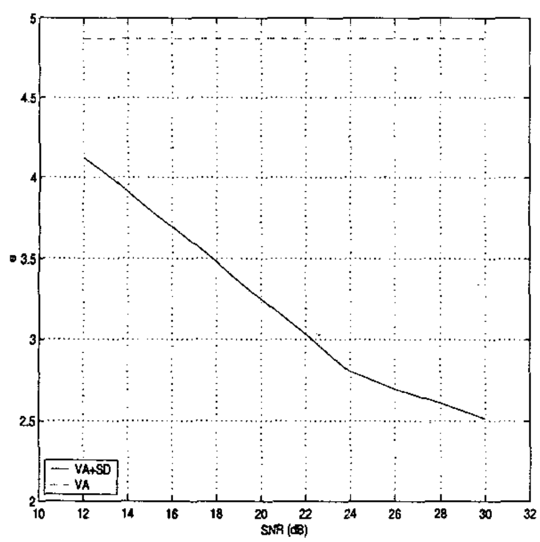

Figure 5. *Expected complexity exponent, $l = 6$, $T = 12$, $L = 4$.*

VA for ML detection for channels with memory. The hybrid algorithm is either the SD modified so to speed-up the search for the closest-point in the lattice or the VA with the imposed sphere-constraints resulting in state-reduction on trellis. The algorithm has expected complexity which is polynomial in the data block length over a wide range of SNR. We found the analytic expression for the expected complexity of the algorithm and illustrated its performance via simulations. The hybrid algorithm is particularly useful when decoding finite data blocks transmitted over channels with long memory.

References

[1] A. D. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithms," *IEEE Trans. on Inf. Theory*, vol. 13, pp. 260-269, 1967.

[2] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, 1957.

[3] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of Computation*, vol. 44, pp. 463-471, April 1985.

[4] H. Vikalo and B. Hassibi, "On Sphere Decoding Algorithm. II. Generalizations, Second-order Statistics, and Applications to Communications," submitted to *IEEE Trans. on Signal Processing.*

[5] B. Hassibi and H. Vikalo, "On Sphere Decoding Algorithm. I. Expected Complexity," submitted to *IEEE Trans. on Signal Processing.*

[6] H. Zamiri-Jafarian and S. Pasupathy, "Complexity reduction of the MLSD/MLSDE receiver using the adaptive state allocation algorithm," *IEEE Trans. on Wireless Comm.*, vol.1, pp. 101-111, January 2002.