Zhou and Nelson *EURASIP Journal on Advances in Signal Processing* (2018) 2018:6
DOI 10.1186/s13634-017-0527-3

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Blind sequential detection for sparse ISI channels

Weiwei Zhou[1] and Jill K. Nelson[2*]

Abstract

We present a computationally efficient blind sequential detection method for data transmitted over a sparse intersymbol interference channel. Unlike blind sequential detection methods designed for general channels, the proposed method exploits the channel sparsity by using estimated channel sparsity to assist in the detection of the transmitted sequence. A Gaussian mixture model is used to describe sparse channels, and two tree-search strategies are applied to estimate the channel sparsity and the transmitted sequence, respectively. To demonstrate the performance improvement achieved by the proposed blind detector, we compare it to conventional joint channel and sequence detection methods that use sparse channel estimation techniques. Simulation results show that the proposed detector not only reduces computational complexity compared to existing methods but also provides superior performance, particularly when the signal to noise ratio is low.

Keywords: Blind sequence detection, Sparse channels, Intersymbol interference, Matching pursuit, Tree search

1 Introduction

Intersymbol interference (ISI) poses a great challenge for reliable high-speed wireless communications, significantly degrading system performance, and hence channel equalization is typically employed at the receiver to mitigate its harmful effects. Due to the wide variety of physical conditions that induce frequency selectivity, wireless channels are often unknown and time-varying, thus posing an additional challenge to the receiver. One approach to combating time-varying distortion is via adaptive equalizers [1] that first use training symbols to estimate the channel and then perform conventional detection using the channel estimate. Transmitting a training sequence consumes bandwidth that could otherwise be used for transmission of information, however, and training sequences must be transmitted at regular intervals when the channel is time varying.

In order to improve spectral efficiency, considerable research effort has been devoted to the application of blind equalization techniques in wireless communication systems in recent decades [2–8]. A popular approach to blind equalization is to jointly estimate the unknown channel and transmitted data based on the maximum likelihood (ML) criterion [9]. The joint estimates are obtained using the least-squares (LS) algorithm for channel estimation and the Viterbi algorithm (VA) for data detection. The LS algorithm is applied iteratively using the data along each surviving path of the VA trellis, generating channel estimates that are used in the search extending from each path. One major drawback of joint estimation methods is their potentially prohibitive complexity, particularly for long channels and/or large symbol alphabets. In order to simplify the blind equalization process, researchers have proposed blind sequential detection approaches that avoid explicit estimation of the channel [10, 11]. In [10], for example, a framework is developed for applying a blind Bayesian maximum likelihood (ML) sequence detector for ISI channels. The channel taps are modeled as stochastic quantities drawn from a known probability distribution, and the tree-based stack algorithm is used to estimate the transmitted sequence based on a Bayesian probability metric that incorporates implicit estimation of the channel. While the Bayesian ML method achieves promising performance results for moderate channel lengths, computational complexity continues to be a drawback for longer channels.

*Correspondence: jnelson@gmu.edu
[2]Department of Electrical and Computer Engineering, George Mason University, 4400 University Dr., Fairfax, VA 22030, USA
Full list of author information is available at the end of the article

Most existing blind equalization approaches have been designed to recover data transmitted over general (non-sparse) ISI channels. In high-speed wireless applications such as underwater acoustic (UWA) communications, ultrawide band (UWB) communications, and high-definition television (HDTV) systems, the discrete-time channel impulse response has a very large channel memory, but only a small number of significant channel coefficients contribute to the distortion of the transmitted signal. When blind equalization approaches that have been designed for general channels are applied to sparse channels, all channel coefficients are treated as significant when estimating the transmitted signal, yielding high-computational burden for the receiver and decreasing the accuracy of the resulting data detection. If we take channel sparsity into consideration when designing the blind equalizer, both computational efficiency and performance can be improved.

In order to blindly estimate a sequence transmitted over a sparse channel, popular sparse channel estimation methods such as matching pursuit (MP), orthogonal matching pursuit (OMP), and basis pursuit (BP) [12–14] can be combined with the VA or stack algorithm using a joint channel estimation and data detection framework. As we show in Section 5, however, these conventional methods result in high complexity, particularly when signal to noise ratio (SNR) is low. In this paper, we propose a computationally efficient blind sequential detection method for recovering data transmitted over sparse ISI channels. Two tree-search-based algorithms are used in the blind detection process: the breadth-first M-algorithm (MA) is used to estimate channel sparsity, and the best-first stack algorithm (SA) exploits the channel sparsity estimate to efficiently detect the transmitted sequence. The two algorithms use different strategies to search a tree structure. The MA extends all possible paths and retains the M paths with the highest likelihood at a given depth, while the SA extends only the path with the highest likelihood but retains unextended paths. The proposed method moves beyond existing blind sequential detection approaches by incorporating a statistical model of the sparse channel characteristics to reduce the number of path explorations required in the tree search. By combining the two tree-search algorithms in an iterative framework, we avoid the need for training data. Additionally, simulations show that computational complexity savings can be achieved when tree-search algorithms are used in place of conventional MP, OMP, and BP for sparse channel estimation.

The remainder of the paper is organized as follows. In Section 2, the overall system model is described, and the sparse channel model is presented. Section 3 provides an overview of the stack-based sequential detection method for unknown sparse ISI channels. The proposed blind sequential detection algorithm, which combines stack-based sequential detection with channel sparsity estimation using the MA, is described in Section 4. In Section 5, the proposed approach is compared to conventional joint channel estimation and data detection methods that use MP, OMP, and BP with respect to both performance and computational complexity. Section 6 concludes the paper.

2 System model

We consider a discrete-time, baseband equivalent communication system, a block diagram of which is shown in Fig. 1. The information bits are passed through an error control encoder with rate $R = \frac{1}{r}$. No training sequence is transmitted. For a block of information bits with length N, denoted by $\mathbf{b}_1^N$, a length-rN sequence of coded bits, $\mathbf{x}_1^{rN}$, is generated. The encoded sequence is transmitted over a length-L_h sparse ISI channel $\mathbf{h} = [h_0, \ldots, h_{L_h-1}]$ with additive white Gaussian noise (AWGN) w_k of variance σ^2. The received sequence $\mathbf{y}_1^{rN} = [y_1, \ldots, y_{rN}]^T$ is the input to a detector that blindly estimates the transmitted information bits $\mathbf{b}_1^N$, where the received signal at time instant k, y_k, $k = 1, 2, \ldots, rN$, is defined as

$$y_k = \sum_{i=0}^{L_h-1} h_i x_{k-i} + w_k. \tag{1}$$

The length-rN received sequence can be expressed in matrix form as $y_1^{rN} = \mathbf{X}^{(rN)}\mathbf{h} + \mathbf{w}$, where the signal matrix $\mathbf{X}^{(rN)}$ is defined as

$$\mathbf{X}^{(rN)} = \begin{bmatrix} x_1 & 0 & \ldots & 0 \\ x_2 & x_1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ x_{L_h} & x_{L_h-1} & \ldots & x_1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{rN} & x_{rN-1} & \ldots & x_{rN-L_h+1} \end{bmatrix}.$$

We assume that the channel has only L_a active (non-zero) taps, $\bar{\mathbf{h}}_a = [h_0, h_1, \ldots, h_{L_a-1}]^T$, where $L_a \ll L_h$. In order to better describe the channel sparsity, we use $d_i \in \{0, 1\}$ as an indicator to denote whether the i-th tap, h_i, is active or inactive:

$$d_i = \begin{cases} 0, & h_i \text{ is inactive,} \\ 1, & h_i \text{ is active.} \end{cases} \tag{2}$$

Hence, $\mathbf{d} = \{d_0, \ldots, d_{L_h-1}\}$ is a binary vector that denotes the sparsity of the channel. We assume that d_i is equally likely to be 0 and 1. The number of active channel taps, L_a, therefore, can be expressed as $L_a = ||\mathbf{d}||_0$, where the l_0-norm of the vector $\mathbf{d}$ counts the number of non-zero elements in $\mathbf{d}$.

In practical communication systems, channels cannot be exactly sparse; the inactive taps have very small (but

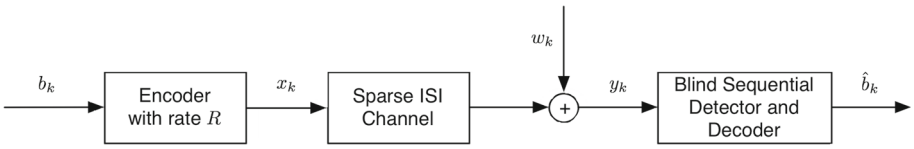

Fig. 1 System model for sparse ISI channel with blind sequential detector. The data b_k is passed through an error control encoder with rate $R = \frac{1}{r}$. The coded sequence x_k is transmitted over a sparse ISI channel with additive white Gaussian noise w_k. The received sequence y_k is processed by a blind sequential detector and decoder to detect the transmitted data sequence

non-zero) values. To reflect this, we use a Gaussian mixture to describe the channel tap values. Active taps are drawn from a zero-mean Gaussian distribution with relatively large variance, while inactive taps are drawn from a zero-mean Gaussian distribution with much smaller variance [15, 16]:

$$h_i \sim (1-d_i)\cdot\mathcal{N}\left(0,\sigma_0^2\right)+d_i\cdot\mathcal{N}\left(0,\sigma_1^2\right), \tag{3}$$

where σ_0^2 denotes the variance of h_i in the inactive state, and $\sigma_1^2 \gg \sigma_0^2$ denotes the variance of h_i in the active state.

We further assume that the L_h channel taps are independent. This assumption is well justified, especially for wireless fading channels that are rich in scattering [17–19]. The distribution of the channel $\boldsymbol{h}$ is then given by

$$\begin{aligned}
P(\mathbf{h}) &= \sum_{\mathbf{d}} p(\mathbf{h}|\mathbf{d})p(\mathbf{d}) = \sum_{\mathbf{d}}\prod_{i=0}^{L_h-1} p(h_i|d_i)p(d_i)\\
&= \frac{1}{2^{L_h}}\sum_{\mathbf{d}}\prod^{||\mathbf{d}||_0} P(h_i|d_i=1)\prod^{L_h-||\mathbf{d}||_0} P(h_i|d_i=0)\\
&= \frac{1}{2^{L_h}}\sum_{\mathbf{d}}\left(\frac{\exp\left(-\frac{h_i^2}{2\sigma_1^2}\right)}{\sqrt{2\pi\sigma_1^2}}\right)^{||\mathbf{d}||_0}\left(\frac{\exp\left(-\frac{h_i^2}{2\sigma_0^2}\right)}{\sqrt{2\pi\sigma_0^2}}\right)^{L_h-||\mathbf{d}||_0}\\
&= \frac{1}{2^{L_h}}\sum_{\mathbf{d}}\left(\frac{1}{\sqrt{2\pi\sigma_1^2}}\right)^{||\mathbf{d}||_0}\left(\frac{1}{\sqrt{2\pi\sigma_0^2}}\right)^{L_h-||\mathbf{d}||_0}\exp\left(-\frac{\mathbf{h}^T\Gamma_{\mathbf{d}}\mathbf{h}}{2\sigma^2}\right),
\end{aligned} \tag{4}$$

where

$$\begin{aligned}
\Gamma_{\mathbf{d}} &= \mathrm{Diag}\{\{\alpha_i\}\},\\
\alpha_i &= \begin{cases}\gamma_0=\frac{\sigma_0^2}{\sigma^2}, & \text{for } d_i=0\\ \gamma_1=\frac{\sigma_1^2}{\sigma^2}, & \text{for } d_i=1\end{cases}, \quad i=0,\ldots,L_h-1.
\end{aligned} \tag{5}$$

Under this model, the channel is treated as a stochastic random vector at the receiver. The receiver has no knowledge of the channel taps h_i nor of the sparsity vector $\mathbf{d}$. The diagonal matrix $\Gamma_{\mathbf{d}}$ that describes the channel sparsity in (4) varies with realizations of the vector $\mathbf{d}$. The diagonal element $\alpha_i = \gamma_1$ indicates that the channel tap h_i is active, and $\alpha_i = \gamma_0$ indicates that h_i is inactive. We assume that the receiver has knowledge of the variance of active and inactive taps σ_0^2 and σ_1^2. We also assume that the receiver has knowledge of the code generator polynomial C, the variance of the AWGN σ^2, and the channel length L_h. For simplicity, only binary phase shift keying (BPSK) encoded data is considered. Extension to transmission of data from larger constellations is straightforward.

3 Stack algorithm for unknown sparse ISI channels

The stack algorithm (SA), which was originally proposed for decoding convolutional and tree codes [20], is a best-first tree-search technique that can be used to approximate the ML solution to sequential detection problems [21]. When the SA is applied to sequential detection, the tree represents the space spanned by a sequence of transmitted bits. Each path in the tree represents a possible realization of the transmitted sequence. A metric is computed for each path; the path metric represents the likelihood that the corresponding bit sequence was transmitted, conditioned on the observations. At each time step, the SA extends the path with the highest likelihood (metric) and stores a set of possible paths and their associated metrics in a stack (or list) in order of decreasing metric value. The SA terminates when the top path in the stack reaches a leaf of the tree, or equivalently when the top path represents a full block of transmitted bits.

To navigate a tree using the SA, paths are progressively extended from depth 1 to depth N. Hence, the algorithm must compute the path metric for a partial-length sequence given a full block of observations; this partial-path metric is used to decide which path in the stack will be extended in the next stage. The metric derivation for unknown general ISI channels is introduced in [10]. For sparse ISI channels, we derive a new form for the path metric using the model defined in [10]. The probability that a length-n sequence $\mathbf{b}_1^n = [b_1, b_2, \ldots, b_n]^T$ was transmitted given the complete received sequence $\mathbf{y}_1^{rN}$ and the code thus far, $C^{(n)}$, is expressed as

$$\begin{aligned}
P\left(\mathbf{b}_1^n|\mathbf{y}_1^{rN},C^{(n)}\right) &= \int_{\mathbf{h}} P\left(\mathbf{b}_1^n,\mathbf{h}|\mathbf{y}_1^{rN},C^{(n)}\right)\mathrm{d}\mathbf{h}\\
&= \frac{P\left(\mathbf{b}_1^n\right)}{P\left(\mathbf{y}_1^{rN}|C^{(n)}\right)}\int_{\mathbf{h}} P\left(\mathbf{y}_1^{rN}|\mathbf{b}_1^n,\mathbf{h},C^{(n)}\right)P(\mathbf{h})\mathrm{d}\mathbf{h}.
\end{aligned} \tag{6}$$

We can eliminate the term $P\left(\mathbf{y}_1^{rN}|C^{(n)}\right)$ since it is equal for all paths, and the path metric can be written as

$$m\left(\mathbf{b}_1^n\right) = P\left(\mathbf{b}_1^n\right)\int_{\mathbf{h}} P\left(\mathbf{y}_1^{rN}|\mathbf{b}_1^n,\mathbf{h},C^{(n)}\right)P(\mathbf{h})\,\mathrm{d}\mathbf{h}. \tag{7}$$

We take the sparsity of the channel into consideration and find a closed-form expression for the integral in (7) using the channel model given in (4). Applying the techniques used to derive the path metric for general unknown ISI channels in [10], we assume that $\mathbf{y}_1^{rn}$ is independent of $\mathbf{y}_{rn+1}^{rN}$ and that $\mathbf{y}_{rn+1}^{rN}$ is independent of $\mathbf{b}_1^n$ to obtain the following expression for the path metric:

$$\begin{aligned} m(\mathbf{b}_1^n) &= \frac{1}{2^{L_h}}\left(\frac{1}{2\sqrt{2\pi\sigma^2}}\right)^{rn}\left(\frac{1}{\sqrt{2\pi(\sigma^2+1)}}\right)^{r(N-n)} \\ &\quad\times\prod_{i=rn+1}^{rN}\exp\left(-\frac{y_i^2}{2(\sigma^2+1)}\right)\int_{\mathbf{h}}\prod_{i=1}^{rn}\exp\left(-\frac{\left(y_i-\mathbf{h}^T\mathbf{x}_{i-L+1}^i\right)^2}{2\sigma^2}\right)P(\mathbf{h})\,\mathrm{d}\boldsymbol{h} \\ &= A(y_i)\int_{\mathbf{h}}\exp\left(-\frac{1}{2\sigma^2}\left(R_{yy}^{(rn)}[0]-2\mathbf{h}^T\mathbf{r}_{yx}^{(rn)}+\mathbf{h}^T R_{xx}^{(rn)}\mathbf{h}\right)\right)P(\mathbf{h})\,\mathrm{d}\mathbf{h}, \end{aligned} \tag{8}$$

where

$$\begin{aligned} A(y_i) &= \frac{1}{2^{L_h}}\left(\frac{1}{2\sqrt{2\pi\sigma^2}}\right)^{rn}\left(\frac{1}{\sqrt{2\pi(\sigma^2+1)}}\right)^{r(N-n)} \\ &\quad\times\prod_{i=rn+1}^{rN}\exp\left(-\frac{y_i^2}{2(\sigma^2+1)}\right), \end{aligned} \tag{9}$$

$$R_{yy}^{(rn)}[0] = \sum_{i=1}^{rn} y_i^2, \tag{10}$$

$$\mathbf{r}_{yx}^{(rn)} = \sum_{i=1}^{rn} y_i\mathbf{x}_{i-L_h+1}^i, \tag{11}$$

and

$$R_{xx}^{(rn)} = \sum_{i=1}^{rn}\left(\mathbf{x}_{i-L_h+1}^i\right)\left(\mathbf{x}_{i-L_h+1}^i\right)^T. \tag{12}$$

Substituting (4) into (8) and integrating the quadratic exponential form yields:

$$\begin{aligned} m(\mathbf{b}_1^n) &= B(y_i)\sum_{\mathbf{d}}\left(\frac{1}{\sqrt{2\pi\sigma_1^2}}\right)^{||\mathbf{d}||_0}\left(\frac{1}{\sqrt{2\pi\sigma_0^2}}\right)^{L_h-||\mathbf{d}||_0} \\ &\quad\times\left|R_{xx}^{(rn)}+\Gamma_{\mathbf{d}}\right|^{-\frac{1}{2}}\exp\left(\frac{1}{2\sigma^2}\left(\mathbf{r}_{yx}^{(rn)}\right)^T\left(R_{xx}^{(rn)}+\Gamma_{\mathbf{d}}\right)^{-1}\mathbf{r}_{yx}^{(rn)}\right), \end{aligned} \tag{13}$$

where

$$B(y_i) = \sigma^{rn}A(y_i)\exp\left(-\frac{1}{2\sigma^2}R_{yy}^{(rn)}[0]\right). \tag{14}$$

Note that the term $\left(\mathbf{r}_{yx}^{(rn)}\right)^T\left(R_{xx}^{(rn)}+\Gamma_{\mathbf{d}}\right)^{-1}\mathbf{r}_{yx}^{(rn)}$ takes a form similar to the least-squares estimate of the channel, demonstrating the implicit learning of the channel vector that is integrated in the metric of the tree-search-based detection approach.

Using the sparse channel model described in Section 2, we are able to incorporate prior information about channel sparsity into the path metric of the SA. However, the computation of $m(\mathbf{b}_1^n)$ in (13) considers all possible realizations of the vector $\mathbf{d}$. Without simplification, we would need to evaluate and sum 2^{L_h} exponential terms, which is computationally impractical for channels with long delay spread. For a given sparse channel, the sparsity vector $\mathbf{d}$ is just one of the 2^{L_h} possible binary vectors. Therefore, if the channel sparsity can be determined, the path metric can be computed for a unique vector $\mathbf{d}$. In order to achieve this, we propose a computationally efficient sparsity estimation technique, described in the following section.

4 Computationally efficient blind sequential detection

4.1 Channel sparsity detection using the M-algorithm

The proposed method combines the M-algorithm (MA) and the stack algorithm (SA) to perform blind sequential detection; we refer to the overall algorithm as the MA-SA for simplicity. The MA is used to find the active channel tap indices and estimate the unique binary vector $\mathbf{d}$ corresponding to the sparse channel. The path metric of the SA is then computed using the channel sparsity estimated by the MA.

Both the MA [22] and the SA are built upon a tree structure representation of the space spanned by a sequence of transmitted bits. In contrast to the SA, however, the MA employs a breadth-first strategy for searching the tree. At a given depth of the tree, it extends all possible paths and retains the M paths with the highest likelihood (or largest path metrics). In the tree structure of the SA, each path represents a possible realization of the transmitted sequence; the MA can be used to estimate the channel sparsity along each path. The SA path metrics can be computed using the estimated channel sparsity in combination with the observations and the symbol sequence associated with each path. The metrics are used to guide the tree search to find the most likely transmitted sequence. Such a mechanism establishes a foundation for combining the MA and the SA. A flow chart of the MA-SA algorithm is shown in Fig. 2.

To better illustrate the proposed algorithm, we provide a step-by-step procedure under the assumption that $M = 1$.

Step 0. (Initialization) Consider a partial path through the tree that represents a length-n sequence $\mathbf{b}_1^n$. The MA-SA algorithm is initialized with the assumption that all channel taps are inactive, i.e., $\mathbf{d}^{(0)} = [0, 0, \ldots, 0]$. The corresponding path metric can be computed as

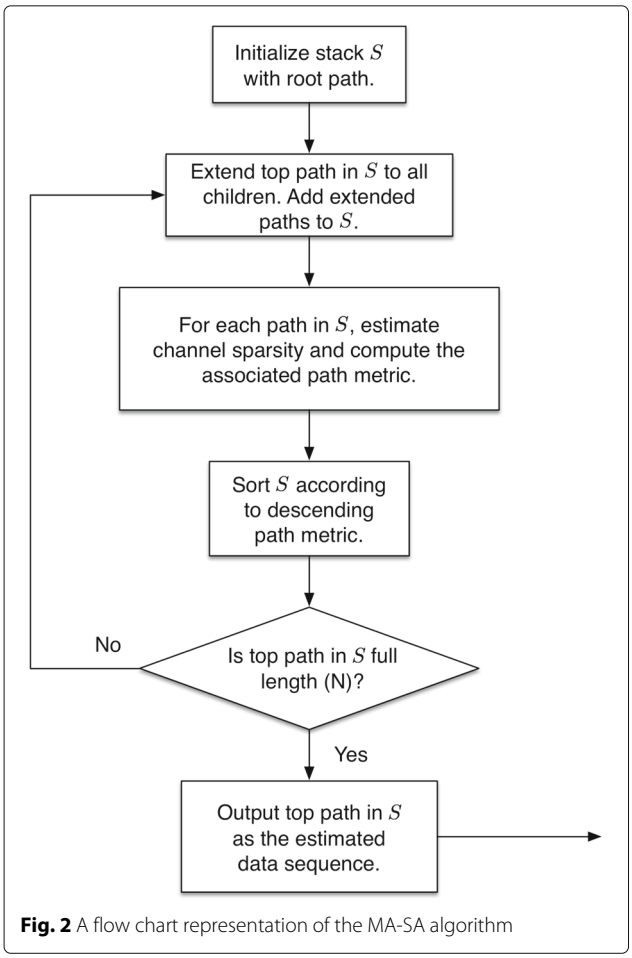

Fig. 2 A flow chart representation of the MA-SA algorithm

$$
\begin{aligned}
m\left(\mathbf{b}_1^n\right)_{\mathbf{d}^{(0)}} = \; & B(y_i)\frac{\left|R_{xx}^{(rn)}+\Gamma_{\mathbf{d}^{(0)}}\right|^{-\frac{1}{2}}}{\left(\sqrt{2\pi\sigma_0^2}\right)^{L_h}} \\
& \times \exp\left(\frac{\left(\mathbf{r}_{yx}^{(rn)}\right)^T\left(R_{xx}^{(rn)}+\Gamma_{\mathbf{d}^{(0)}}\right)^{-1}\mathbf{r}_{yx}^{(rn)}}{2\sigma^2}\right) \\
= \; & B(y_i)\left(\frac{1}{\sqrt{2\pi\sigma_0^2}}\right)^{L_h}\left|\mathbf{R}\left(\mathbf{d}^{(0)}\right)\right|^{-\frac{1}{2}} \\
& \times \exp\left(\frac{\left(\mathbf{r}_{yx}^{(rn)}\right)^T\mathbf{R}^{-1}\left(\mathbf{d}^{(0)}\right)\mathbf{r}_{yx}^{(rn)}}{2\sigma^2}\right),
\end{aligned}
\tag{15}
$$

where

$$
\mathbf{R}\left(\mathbf{d}^{(0)}\right) = R_{xx}^{(rn)} + \Gamma_{\mathbf{d}^{(0)}}, \tag{16}
$$

and

$$
\Gamma_{\mathbf{d}^{(0)}} = \mathrm{Diag}\{\gamma_0\}. \tag{17}
$$

The vector $\mathbf{d}^{(0)}$ is assigned to $\mathbf{d}_*^{(0)}$, where the vector $\mathbf{d}_*^{(p)}$ is used to store the best (highest metric) vector obtained at the p-th iteration. A set $\mathcal{S}$ is defined to store the associated indices of the active channel taps at each iteration. Initialize iteration counter to $p = 0$ and index vector to $\mathcal{S} = \{\}$.

Step 1. Change a single 0 element of the binary vector $\mathbf{d}_*^{(p)}$ to 1 to generate a set of $L_h - p$ possible binary vectors, $\mathbf{d}_j^{(p+1)}$, where $j \notin \mathcal{S} \in \{1, \ldots, L_h\}$ denotes that $\mathbf{d}_j^{(p+1)}$ is identical to $\mathbf{d}_*^{(p)}$ with the exception of the j-th element.

Step 2. Compute the path metric $m\left(\mathbf{b}_1^n\right)_{\mathbf{d}_j^{(p+1)}}$ for each of the possible weight-$(p+1)$ vectors, and assign the vector that corresponds to the largest metric to the vector $\mathbf{d}_*^{(p+1)}$, i.e., $\mathbf{d}_*^{(p+1)} = \mathbf{d}_{j^*}^{(p+1)}$, where

$$
j^* = \underset{j}{\mathrm{argmax}}\; m\left(\mathbf{b}_1^n\right)_{\mathbf{d}_j^{(p+1)}},\; j \notin \mathcal{S} \in \{1, \ldots, L_h\}. \tag{18}
$$

Step 3. Update the index vector $\mathcal{S}$ by including j^*, and increment p by 1:

$$
\mathcal{S} = \mathcal{S} \cup j^*, \qquad p = p + 1. \tag{19}
$$

Step 4. If all active channel tap indices are located or a stopping criterion (described in Section 4.3) is satisfied, terminate the algorithm; otherwise, return to step 1.

The final vector $\mathbf{d}_*^{(\hat{L}_a)}$ provides the estimate of channel sparsity, where $\hat{L}_a$ denotes the estimate of the active channel length. $\hat{L}_a = L_a$ if L_a is known at the receiver. The corresponding metric $m\left(\mathbf{b}_1^n\right)_{\mathbf{d}_*^{(\hat{L}_a)}}$ will be used by the SA to perform the tree search. For a given L_a, the total number of the metric computations is given by $1 + (L_a + 1)\left(L_h - \frac{L_a}{2}\right)$. We will show in the next section that the number of metric computations can be further reduced. The proposed MA-based sparsity estimation is summarized in a flow chart in Fig. 3.

An example of the sparsity estimation procedure is shown in Fig. 4. The example considers a length-4 channel with two active taps, $\mathbf{h} = [h_0, 0, h_1, 0]$, and the figure shows the k-th branch of the SA tree. The red boxes show the best sparsity vectors obtained at each iteration. We initialize the vector $\mathbf{d}^{(0)} = [0, 0, 0, 0]$ and compute the path metric $m\left(\mathbf{b}_1^n\right)_{\mathbf{d}^{(0)}}$ using (15). At step 1, there are four possibilities for the sparsity vector: $\mathbf{d}_1^{(1)}, \ldots, \mathbf{d}_4^{(1)}$, and the corresponding path metric can be obtained similar to using (15) for each vector. We assume that $m\left(\mathbf{b}_1^n\right)_{\mathbf{d}_1^{(1)}}$ is the largest path metric at this step, and thus $\mathbf{d}_*^{(1)} = \mathbf{d}_1^{(1)}$, indicating that the first channel tap is active. From $\mathbf{d}_*^{(1)}$, three possibilities for the sparsity vector, $\mathbf{d}_2^{(2)}, \mathbf{d}_3^{(2)}$, and $\mathbf{d}_4^{(2)}$, can be generated. The corresponding path metric for each vector is computed, and $m\left(\mathbf{b}_1^n\right)_{\mathbf{d}_3^{(2)}}$ is assumed to be largest.

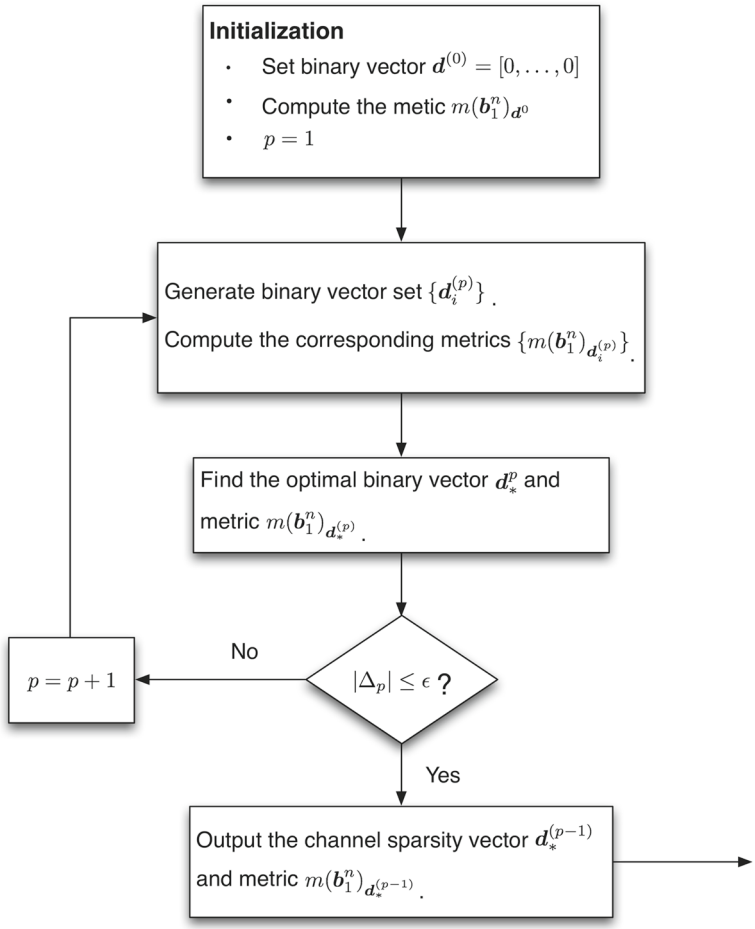

Fig. 3 A flow chart representation of the MA-based technique for updating the path metric

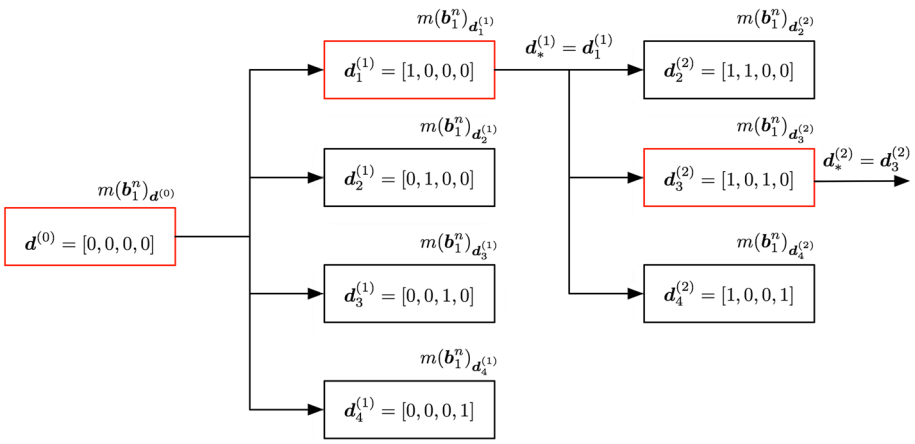

Fig. 4 An example of channel sparsity estimation using the MA. A length-4 channel with two active taps: $\mathbf{h} = [h_0, 0, h_1, 0]$ is used. The red boxes show the sparsity vectors identified as best at each iteration of the algorithm

Therefore, $\mathbf{d}_*^{(2)} = \mathbf{d}_3^{(2)}$, indicating that the third channel tap is also active. Now that the two active taps have been located (and assuming the number of active taps is known), the sparsity estimation algorithm can be terminated. The quantity $m\left(\mathbf{b}_1^n\right)_{\mathbf{d}_3^{(2)}}$ serves as the path metric for the SA at this branch.

4.2 Efficient metric computation

As described in the proposed MA-based sparsity estimation, a new path metric is computed at each iteration, involving both inversion of a matrix and computing its determinant. The total $1 + (L_a + 1)\left(L_h - \frac{L_a}{2}\right)$ metric computations are still a heavy burden for the detector. By exploiting the properties of matrices, we can further reduce the computational complexity and accelerate the metric update.

In order to develop a general expression for efficient metric computation, consider the binary sparsity vector obtained at the $(p-1)$-th iteration: $\mathbf{d}_*^{(p-1)}$. The matrix $\mathbf{R}\left(\mathbf{d}_*^{(p-1)}\right)$ in the path metric is computed as

$$\mathbf{R}\left(\mathbf{d}_*^{(p-1)}\right) = R_{xx}^{(rn)} + \Gamma_{\mathbf{d}_*^{(p-1)}}. \tag{20}$$

At the p-th iteration, a set of $L_h - p$ vectors $\mathbf{d}_i^{(p)}$ is generated by turning just one 0 element of the vector $\mathbf{d}_*^{(p-1)}$ to 1. The matrix $\mathbf{R}\left(\mathbf{d}_i^{(p)}\right)$ is given by

$$\mathbf{R}\left(\mathbf{d}_i^{(p)}\right) = R_{xx}^{(rn)} + \Gamma_{\mathbf{d}_i^{(p)}}. \tag{21}$$

It is clear that $\mathbf{R}\left(\mathbf{d}_i^{(p)}\right)$ differs from $\mathbf{R}\left(\mathbf{d}_*^{(p-1)}\right)$ in only one position: (i,i). Thus, by using the properties of matrix inversion, the inverse and determinant of $\mathbf{R}\left(\mathbf{d}_i^{(p)}\right)$ can be easily obtained from the inverse and determinant of $\mathbf{R}\left(\mathbf{d}_*^{(p-1)}\right)$. We assume that we already have the inverse and determinant of $\mathbf{R}\left(\mathbf{d}_*^{(p-1)}\right)$, denoted by $V = \mathbf{R}^{-1}\left(\mathbf{d}_*^{(p-1)}\right)$ and $U = \left|\mathbf{R}\left(\mathbf{d}_*^{(p-1)}\right)\right|$, respectively. $\mathbf{R}\left(\mathbf{d}_i^{(p)}\right)$ can then be expressed as

$$\mathbf{R}\left(\mathbf{d}_i^{(p)}\right) = \mathbf{R}\left(\mathbf{d}_*^{(p-1)}\right) + \beta\mathbf{u}\mathbf{u}^T, \tag{22}$$

where $\beta = \gamma_1 - \gamma_0$ and $\mathbf{u} = [0,\ldots,1,\ldots,0]^T$. The single non-zero element in the vector $\mathbf{u}$ is at index i. Applying the Sherman-Morrison formula [23], the inverse of $\mathbf{R}\left(\mathbf{d}_i^{(p)}\right)$ can be expressed as

$$\begin{aligned}\mathbf{R}^{-1}\left(\mathbf{d}_i^{(p)}\right) &= \left(\mathbf{R}\left(\mathbf{d}_*^{(p-1)}\right) + \beta\mathbf{u}\mathbf{u}^T\right)^{-1}\\ &= \mathbf{R}^{-1}\left(\mathbf{d}_*^{(p-1)}\right) - \frac{\beta\mathbf{R}^{-1}\left(\mathbf{d}^{(p-1)}\right)\mathbf{u}\mathbf{u}^T\mathbf{R}^{-1}\left(\mathbf{d}_*^{(p-1)}\right)}{1+\beta\mathbf{u}^T\mathbf{R}^{-1}\left(\mathbf{d}_*^{(p-1)}\right)\mathbf{u}}\\ &= V - \frac{\beta V\mathbf{u}\mathbf{u}^T V}{1+\beta\mathbf{u}^T V\mathbf{u}}\\ &= V - \frac{\beta}{1+\beta V(i,i)}V(:,i)V(i,:),\end{aligned} \tag{23}$$

where $V(:,i)$ is the ith column of the matrix V.

The determinant of $\mathbf{R}\left(\mathbf{d}_i^{(p)}\right)$ can be computed as

$$\begin{aligned}\left|\mathbf{R}\left(\mathbf{d}_i^{(p)}\right)\right| &= \left|\mathbf{R}\left(\mathbf{d}_*^{(p-1)}\right) + \beta\mathbf{u}\mathbf{u}^T\right|\\ &= \left(1+\beta\mathbf{u}^T\mathbf{R}^{-1}\left(\mathbf{d}_*^{(p-1)}\right)\mathbf{u}\right)\left|\mathbf{R}\left(\mathbf{d}_*^{(p-1)}\right)\right|\\ &= (1+\beta V(i,i))U.\end{aligned} \tag{24}$$

Combining Eqs. (23) and (24), the path metric associated with the sparsity vector $\mathbf{d}_i^{(p)}$ can be computed as

$$\begin{aligned}m\left(\mathbf{b}_1^n\right)_{\mathbf{d}_i^{(p)}} &= \frac{B(y_i)}{\sqrt{(1+\beta V(i,i))U}}\left(\frac{1}{\sqrt{2\pi\sigma_1^2}}\right)^{||\mathbf{d}_i^{(p)}||_0}\left(\frac{1}{\sqrt{2\pi\sigma_0^2}}\right)^{L_h-||\mathbf{d}_i^{(p)}||_0}\\ &\quad\times\exp\left(\frac{\left(\mathbf{r}_{yx}^{(rn)}\right)^T}{2\sigma^2}\left(V - \frac{\beta}{1+\beta V(i,i)}V(:,i)V(i,:)\right)\mathbf{r}_{yx}^{(rn)}\right)\\ &= m\left(\mathbf{b}_1^n\right)_{\mathbf{d}_*^{(p-1)}}\frac{\sigma_0}{\sigma_1\sqrt{\theta}}\exp\left(-\frac{\beta}{2\theta\sigma^2}\left(\mathbf{r}_{yx}^{(rn)}\right)^T V(:,i)V(i,:)\mathbf{r}_{yx}^{(rn)}\right),\end{aligned} \tag{25}$$

where $\theta = 1 + \beta V(i,i)$.

From (25), we can see that the path metric associated with the vector $\mathbf{d}_i^{(p)}$ can be computed via a simple update from the metric for $\mathbf{d}_*^{(p-1)}$. The computation for the metric update is dominated by the vector multiplications, $\left(\mathbf{r}_{yx}^{(rn)}\right)^T V(:,i)V(i,:)\mathbf{r}_{yx}^{(rn)}$. The inverse and determinant computations are no longer needed at every iteration; they are performed only once to obtain $R^{-1}\left(\mathbf{d}^{(0)}\right)$ and $\left|R\left(\mathbf{d}^{(0)}\right)\right|$ when the initial metric $m\left(\mathbf{b}_1^n\right)_{\mathbf{d}^{(0)}}$ is computed.

4.3 Stopping criterion

The MA-based sparsity detection described in Section 4.1 relies on searching for active channel indices and terminates when all the active indices have been identified. In practical communication systems, the number of active taps, L_a, is usually an unknown parameter. A stopping criterion must be implemented to terminate the iterative sparsity detection process without knowledge of L_a. To address this, we compare the path metrics we obtain at the L_a-th and (L_a+1)-th iterations and use the difference between them to identify a stopping criterion.

At the L_a-th iteration, the best estimated sparse vector with L_a non-zero elements is $\mathbf{d}_*^{(L_a)}$, and the corresponding path metric can be written as

$$m\left(\mathbf{b}_1^n\right)_{\mathbf{d}_*^{(L_a)}} = A(y_i)\sigma^{rn}\left(\frac{1}{\sqrt{2\pi\sigma_1^2}}\right)^{L_a}\left(\frac{1}{\sqrt{2\pi\sigma_0^2}}\right)^{L_h-L_a}\left|R_{xx}^{(rn)}+\Gamma_{\mathbf{d}_*^{(L_a)}}\right|^{-\frac{1}{2}} \times \exp\left(-\frac{1}{2\sigma^2}\left(R_{yy}^{(rn)}[0]-\left(\mathbf{r}_{yx}^{(rn)}\right)^T\left(R_{xx}^{(rn)}+\Gamma_{\mathbf{d}_*^{(L_a)}}\right)^{-1}\mathbf{r}_{yx}^{(rn)}\right)\right). \tag{26}$$

In the metric, the term $\left(R_{xx}^{(rn)}+\Gamma_{\mathbf{d}_*^{(L_a)}}\right)^{-1}\mathbf{r}_{yx}^{(rn)}$ in the exponential function takes a form that is very similar to the LS estimation of the channel, $\hat{\mathbf{h}}$. Thus, $R_{yy}^{(rn)}[0]-\left(\mathbf{r}_{yx}^{(rn)}\right)^T\left(R_{xx}^{(rn)}+\Gamma_{\mathbf{d}_*^{(L_a)}}\right)^{-1}\mathbf{r}_{yx}^{(rn)}$ measures the error $\mathbf{e}$ between the received signal $\mathbf{y}_1^{rn}$ and the noise-free output predicted by the channel estimate $\hat{\mathbf{h}}$ and assumed bit sequence $\mathbf{x}_1^{rn}$. At the (L_a+1)-th iteration, the metric $m\left(\mathbf{b}_1^n\right)_{\mathbf{d}_*^{(L_a+1)}}$ associated with the best estimated sparse vector $\mathbf{d}_*^{(L_a+1)}$ is updated from the metric $m\left(\mathbf{b}_1^n\right)_{\mathbf{d}_*^{(L_a)}}$ according to

$$\begin{aligned} m\left(\mathbf{b}_1^n\right)_{\mathbf{d}_*^{(L_a+1)}} &= m\left(\mathbf{b}_1^n\right)_{\mathbf{d}_*^{(L_a)}}\frac{\sigma_0}{\sigma_1\sqrt{\theta}} \\ &\quad\times\ \exp\left(-\frac{\beta}{2\theta\sigma^2}\left(\mathbf{r}_{yx}^{(rn)}\right)^T V(:,j^*)V(j^*,:)\mathbf{r}_{yx}^{(rn)}\right) \\ &= m\left(\mathbf{b}_1^n\right)_{\mathbf{d}_*^{(L_a)}}\frac{\sigma_0}{\sigma_1\sqrt{\theta}}\exp\left(\frac{\beta}{2\theta\sigma^2}\Delta_{L_a+1}\right), \end{aligned} \tag{27}$$

where $\Delta_{L_a+1} = \left(\mathbf{r}_{yx}^{(rn)}\right)^T V(:,j^*)V(j^*,:)\mathbf{r}_{yx}^{(rn)}$ denotes the contribution of the (L_a+1)-th estimated active tap to the error $\mathbf{e}$, and j^* denotes the index of the L_a-th estimated active channel tap.

Assuming that the L_a active channel taps are correctly located at the L_a-th iteration, no more active taps are contributing to the computation of the metric $m\left(\mathbf{b}_1^n\right)_{\mathbf{d}_*^{(L_a+1)}}$. Therefore, the corresponding Δ_{L_a+1} will depend only on additive noise. Given this, we can establish a stopping criterion as follows: For a sufficiently small ϵ, at the p-th iteration, $|\Delta_p| \le \epsilon$, the algorithm is terminated. The number of active channel taps can be estimated as $\hat{L}_a = p-1$. The estimated sparse vector is $\mathbf{d}_*^{(p-1)}$.

Assuming the algorithm estimates L_a correctly, one additional round of iterations for $p = L_a+1$ is implemented (relative to the number of iterations implement when L_a is known). This involves $L_h - L_a$ metric updates, which is only a small increase in computational complexity. When the noise level is high, the algorithm is more likely to estimate L_a incorrectly. However, since the channel sparsity is estimated using path metrics that take noise into consideration, we are still able to achieve better channel sparsity estimation than is achieved using conventional methods, as we show in Section 5.

5 Simulation results and discussion

5.1 Performance comparison

In this section, we compare the proposed method to several conventional blind sequential detection methods, all of which are based on joint sparse channel estimation and data detection. The method introduced in [24], for example, finds an ML solution for a joint single-input multiple-output channel and sequence estimation problem using a two-step procedure: (1) channel estimates are obtained for every possible data sequence, and (2) the ML data sequence and corresponding channel estimate are selected. A least-squares (LS) approach is used to implement step 1, and the VA is used to implement step 2. The LS channel estimation algorithm performs a pseudo-inverse of the channel matrix. When the channel is long but sparse, LS does not provide reliable channel estimation since each tap estimate will have a non-zero value [25]. Additionally, for such channels, the VA is prohibitively complex due to the long-channel memory. Therefore, we use sparse channel estimation techniques such as matching pursuit (MP), orthogonal matching pursuit (OMP), and basis pursuit (BP) to replace the LS approach in the first step. In the second step, we use the SA, which is considerably more efficient than the VA for channels with long delay spread and suffers only minimal performance degradation [26]. The combined methods we consider are denoted by MP-SA, OMP-SA, and BP-SA. In addition to these three methods, we also consider a recently proposed joint sparse channel and sequence detection method using the expectation-maximization (EM) algorithm, SC-EM [27].

MP and OMP, two greedy algorithms originally proposed for sparse recovery problems, sequentially estimate the sparse channel by using a training sequence [14, 28]. MP applies sequential forward selection to determine a sparse representation of the channel $\mathbf{h}$ by using a training sequence and its corresponding received signal. The main difference between OMP and MP is that OMP avoids the re-selection problem that occurs in each iteration of MP, thereby accelerating convergence of the algorithm and enhancing the accuracy of sparse channel estimation, but requiring extra computational effort. BP, also referred to as a form of compressed sensing (CS), obtains the sparse channel estimate by solving an $l_1 - l_2$ optimization problem defined in [29]. Numerical results in [30] show that BP delivers a better channel estimate than MP and OMP but requires increased complexity. In order to use MP, OMP, and BP for blind detection, MP-SA, OMP-SA, and BP-SA employ the two-step procedure, described in detail in Section 4, to compute the metric at each path of the tree. On each branch of the tree, MP, OMP, and BP are first used to estimate the sparse channel using the realization of the coded sequence associated with a path. Each path metric is then computed using the estimated channel.

The path metric governs the search through the tree to find the most likely transmitted sequence. The tree search continues until a leaf of the tree is reached [31].

The SC-EM algorithm uses a different two-step procedure than the three methods described above [27]. The expectation (E) step performs sequential detection using forward-backward recursions (BCJR). ML sparse channel estimation is performed in the maximization (M) step using the estimated sequence of symbols. Iteration between these two steps increases the joint likelihood until convergence is reached.

Simulations have been conducted for a length $L_h = 10$ sparse channel with $L_a = 3$ active taps. Information bits are encoded using a rate $R = \frac{1}{2}$ convolutional code with generator polynomials $\mathbf{g}_0 = [1, 1, 1]$ and $\mathbf{g}_1 = [1, 1, 0]$. The initial state of the register is set to $[0, 0]$. Ten thousand blocks of 100 data bits each are transmitted over the sparse channel. For each data block, the active channel taps are drawn from a Gaussian distribution with zero mean and variance $\sigma_1^2 = 1$, and the inactive taps are drawn from a Gaussian distribution with zero mean and variance $\sigma_0^2 = 0.01$. The channel energy is normalized to 1. The locations of active and inactive channel taps are randomly generated from a discrete uniform distribution for each block. For the three methods, the stack size of the SA is set to 10^5 to make erasures rare. We assume that both the channel response and the number of active channel taps L_a are unknown to the receiver. M is set to 1 when we use MA to estimate the channel sparsity in MA-SA.

A performance comparison of the five methods described above is shown in Fig. 5. The proposed MA-SA technique outperforms the competing methods, most substantially when SNR is low. This performance gap can be attributed to the fact that MA-SA's approach of estimating the channel sparsity based on the path metric is more robust to noise than are the competing approaches. As expected, OMP-SA yields better performance than MP-SA due to the introduction of the orthogonalization process. MP/OMP do not take the channel noise into consideration and therefore are more likely to incorrectly estimate the sparse channel. Thus, MP-SA and OMP-SA are outperformed by the other three methods. The BP-SA and SC-EM methods have very similar performance due to the similar procedures they adopt to solve $l_1 - l_2$ optimization problems.

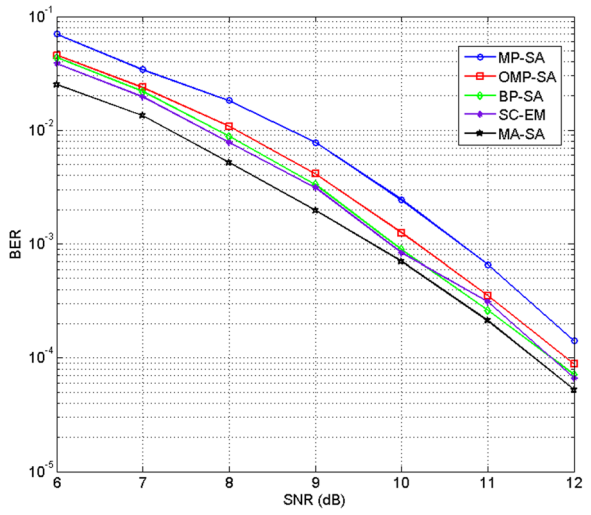

Fig. 5 Performance comparison between MP-SA, OMP-SA, BP-SA, SC-EM, and MA-SA. The performance for each method is computed using 10^4 length-100 sequences transmitted over a 10-tap channel with three active taps

In order to further investigate the performance of the proposed method, we consider a scenario in which a short training sequence is transmitted. In this case, we compare the MA-SA with methods that combine greedy algorithms with the VA, which attains the ML solution in the presence of AWGN. We can use MP, OMP, and BP to estimate the sparse channel using the training sequence. When training is available, each channel tap value can be estimated (rather than estimating only channel sparsity). Thus, sequential detection methods such as the VA can be applied using the estimated channel. Here, we compare the performance of the proposed MA-SA to that of BP-VA, MP-SA, and OMP-SA. (The BP-VA approach is selected for performance comparison because the individual algorithms show strong performance. SC-EM is not included in this simulation, since it uses iterative EM other than the combination of greedy algorithms and sequential detection methods using graph structure.) Simulations are constructed under the same conditions described above. A known length-10 training sequence is appended to each length-100 block of data. Performance results for the four methods are shown in Fig. 6. We observe that the proposed MA-SA method still achieves the best performance among the four methods due to its iterative update of the channel within the tree-search process. BP-VA outperforms the MP/OMP-SA; by solving $l_1 - l_2$ optimization problems, BP can provide a better channel estimate than the MP/OMP, and the VA is able to achieve the ML estimation of the transmitted sequence.

5.2 Efficiency comparison

To illustrate the computational efficiency of the MA-SA method, we compare its computational requirements to those of the MP-SA approach. (We consider only MP-SA for complexity comparison because it is more computationally efficient than the OMP-SA, BP-SA, and SC-EM approaches.) Since both methods operate on possible sequences represented by paths through the tree, we adopt the complexity of computing the metric for a length-n path as the comparison criterion. In order to

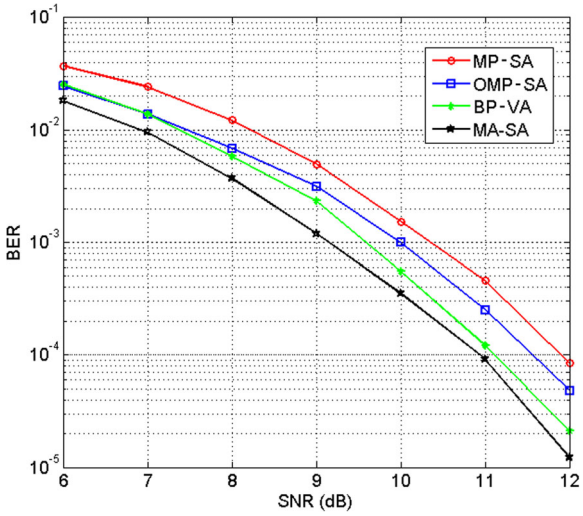

Fig. 6 Performance comparison between MP-SA, OMP-SA, BP-VA, and MA-SA. The performance for each method is computed using a length-10 training sequence for a 10-tap channel with three active taps

compute the metric for a given path, the MP-SA method detects the channel sparsity first and computes the metric using the estimated channel sparsity, while the MA-SA method computes the initial metric for $\mathbf{d}^{(0)} = [0, \ldots, 0]$ and updates the metric with the sparsity detection. The metric computation for MP-SA using the estimated channel sparsity and the initial metric computation for MA-SA have the same complexity. Therefore, to compare the complexity of MP-SA and MA-SA, we need only compare their computational complexity in channel sparsity detection. This complexity depends on the estimate of the active channel length, $\hat{L}_a$, which is affected by the channel itself and by the noise level.

For a path with length n, MP-SA constructs an $n \times L_h$ signal matrix $\mathbf{X}^{(n)}$. At the p-th iteration, a length-n residual vector $\mathbf{b}_p$ is projected onto the direction of each column of $\mathbf{X}^{(n)}$, $\mathbf{a}_i$, $i = 1, \ldots, L_h$. For each iteration, L_h inner product computations are performed. Furthermore, L_h norm computations (inner products) of $\mathbf{a}_i$ must be performed to scale the projection as shown in [12]. Therefore, after the estimated active channel length $\hat{L}_a^{(\text{MP-SA})}$ is obtained, $\hat{L}_a^{(\text{MP-SA})}$ iterations must be performed to estimate the sparsity. Hence, MP-SA must perform $\hat{L}_a^{(\text{MP-SA})} L_h + L_h = \left(\hat{L}_a^{(\text{MP-SA})} + 1\right) L_h$ inner product computations between two length-n vectors.

For a length-n path, at each iteration, the MA-SA computes the metric update $\frac{\sigma_0}{\sigma_1\sqrt{\theta}} \exp\left(-\frac{\beta}{2\theta\sigma^2}\left(\mathbf{r}_{yx}^{(n)}\right)^T V(:,j)V(j,:)\mathbf{r}_{yx}^{(n)}\right)$. We do not consider the multiplications of the constants $\frac{\sigma_0}{\sigma_1\sqrt{\theta}}$ and $\frac{\beta}{2\theta\sigma^2}$, since they are just a one-time operation. Thus, the complexity for each metric update is the inner product of two length-n vectors, $V(j,:)$ and $\mathbf{r}_{yx}^{(n)}$. To obtain the final metric, the metric update is computed $\left(\hat{L}_a^{(\text{MA-SA})} + 1\right)\left(L_h - \frac{1}{2}\hat{L}_a^{(\text{MA-SA})}\right)$ times, so MA-SA must perform $\left(\hat{L}_a^{(\text{MA-SA})} + 1\right)\left(L_h - \frac{1}{2}\hat{L}_a^{(\text{MA-SA})}\right)$ inner product computations between two length-n vectors. The complexity comparison between the MP-SA and MA-SA methods is summarized in Table 1.

Table 1 Computational complexity of the MP-SA and MA-SA methods

Complexity for channel sparsity detection at each path	
	Inner product computations
MP-SA	$\left(\hat{L}_a^{(\text{MP-SA})} + 1\right) L_h$
MA-SA	$\left(\hat{L}_a^{(\text{MA-SA})} + 1\right)\left(L_h - \frac{1}{2}\hat{L}_a^{(\text{MA-SA})}\right)$

The computational complexity for each method is measured in terms of the number of inner product computations required for channel sparsity detection along each path of the tree

When L_a is known at the receiver, both methods will terminate after the L_a-th loop, $\hat{L}^{(\text{MP-SA})} = \hat{L}^{(\text{MA-SA})} = L_a$. Comparing the complexity expressions we derived above, MA-SA performs $\frac{L_a(L_a+1)}{2}$ fewer inner product computations than MP-SA at each path of the tree. While $\frac{L_a(L_a+1)}{2}$ may not be a large number, the computational savings occur along each path. In the application of the SA, a large number of paths will be extended, particularly when SNR is low, making the complexity reduction achieved by MA-SA significant.

When L_a is unknown, we use numerical results to make complexity comparisons. We evaluate the complexity of the MP-SA and MA-SA methods empirically by comparing the average number of paths extended and the total number of inner product computations for each data block. As in the earlier simulations, we consider a length $L_h = 10$ channel with $L_a = 3$ active taps. Results from 10^3 simulations are averaged, and the complexity comparison for various SNR values is shown in Table 2.

The results show that MA-SA extends fewer paths than MP-SA, particularly when SNR is low, due to the robustness of the MA-SA method to noise in channel sparsity detection. As we can see from Table 2, for an SNR of 6 dB, MA-SA performs about 3×10^5 fewer inner product computations than MP-SA, resulting in a roughly 75% reduction in complexity.

6 Conclusions

We have developed and evaluated a tree-based sequential detection method for detecting data transmitted over a sparse ISI channel. We have focused on scenarios in which

Table 2 Computational complexity comparison between MP-SA and MA-SA

	Complexity comparison			
	MP-SA		MA-SA	
	Paths extended	Inner product computations	Paths extended	Inner product computations
6 dB	17,184	945,120	8279	255,997
8 dB	9695	484,752	5183	202,137
10 dB	4748	237,904	3090	128,150
12 dB	3606	151,452	1343	45,976

The comparison is made for length $L_h = 10$ sparse channels with $L_a = 3$ active taps

no training sequence is used in the system and proposed a computationally efficient blind sequential detection method using the MA and SA, both tree-search-based algorithms. A Gaussian mixture model is used to describe the sparse ISI channel, and the MA is applied to blindly estimate the channel sparsity. The estimated channel sparsity is then used to compute the path metrics within the SA, which guides the search in a tree. The MA and SA are combined to achieve blind sequential detection.

For performance comparison, we combined conventional sparse channel estimation techniques such as MP, OMP, and BP with the SA to jointly estimate the sparse channel and transmitted sequence; we considered both blind and semi-blind (when a short training sequence is available) cases. Simulation results show that the proposed MA-SA method is more likely to correctly estimate the channel sparsity along each extended path of the tree than are the comparison methods. Conventional methods are more likely to generate inaccurate channel estimates when SNR is low, which not only limits the performance of the detector but also increases the computational burden. Additionally, we have shown the computational efficiency of the MA-SA approach when the number active taps, L_a is known or unknown. Numerical results for unknown L_a show that proposed MA-SA method achieves both significant computational savings and performance improvement when compared to conventional methods.

Future work will consider alternate channel models and simplifications to accommodate longer channels. We will also investigate the design of a modified SA-MA approach that exploits a multiple tree structure [32] to further improve the efficiency of blind sequential detection.

Acknowledgements
The authors would like to thank the editor and anonymous reviewers for their constructive comments which helped to improve the quality of this paper.

Funding
This material is based upon work supported by the National Science Foundation under Grant No. CCF-1053932 (CAREER).

Authors' contributions
The authors declare that all authors have materially participated in the research and have contributed equally to this paper. Both authors read and approved the final manuscript.

Ethics approval and consent to participate
Not applicable

Consent for publication
Not applicable

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Author details
[1] Department of Bioengineering, George Mason University, 4400 University Dr., Fairfax, VA 22030, USA. [2] Department of Electrical and Computer Engineering, George Mason University, 4400 University Dr., Fairfax, VA 22030, USA.

Received: 19 July 2017 Accepted: 27 December 2017
Published online: 16 January 2018

References

1. SUH Qureshi, Adaptive equalization. Proc. IEEE. **73**(9), 1349–1387 (1985)
2. M Chen, J Tuqan, D Zhi, A quadratic programming approach to blind equalization and signal separation. IEEE Trans. Signal Process. **57**(6), 2232–2244 (2009)
3. R Liu, Y Inouye, Blind equalization of MIMO-FIR channels driven by white but higher order colored source signals. IEEE Trans. Inf. Theory. **48**(5), 1206–1214 (2002)
4. I Santamaria, C Pantaleon, L Vielva, J Ibanez, Blind equalization of constant modulus signals using support vector machines. IEEE Trans. Signal Process. **52**(6), 1773–1782 (2004)
5. V Zarzoso, P Comon, Blind and semi-blind equalization based on the constant power criterion. IEEE Trans. Signal Process. **53**(11), 4363–4375 (2005)
6. L Tong, G Xu, T Kailath, Blind identification and equalization based on second-order statistics: a time domain approach. IEEE Trans. Inf. Theory. **40**(2), 340–349 (1994)
7. S Chen, W Yao, L Hanzo, Semi-blind adaptive spatial equalization for MIMO systems with high-order QAM signalling. IEEE Trans. Wireless Commun. **7**(11), 4486–4491 (2008)
8. H Nguyen, BC Levy, The expectation-maximization Viterbi algorithm for blind adaptive channel equalization. IEEE Trans. Commun. **53**(10), 1671–1678 (2005)
9. N Seshadri, Joint data and channel estimation using blind trellis search techniques. IEEE Trans. Commun. **42**(234), 1000–1011 (1994)
10. JK Nelson, AC Singer, in *The 40th IEEE Annual Conference on Information Sciences and Systems*. Bayesian ML sequence detection for ISI channels, (Princeton, 2006), pp. 693–698
11. RJ Johnson, P Schniter, TJ Endres, JD Behm, DR Brown, RA Casas, Blind equalization using the constant modulus criterion: a review. Proc. IEEE. **86**(10), 1927–1950 (1998)
12. TT Cai, L Wang, Orthogonal matching pursuit for sparse signal recovery with noise. IEEE Trans. Inf. Theory. **57**(7), 4680–4688 (2011)

13. T Kang, RA Iltis, in *IEEE International Conference on Acoustics, Speech and Signal Processing*. Matching pursuits channel estimation for an underwater acoustic OFDM modem, (Las Vegas, 2008), pp. 5296–5299
14. Z Hussain, J Shawe-Taylor, DR Hardoon, C Dhanjal, Design and generalization analysis of orthogonal matching pursuit algorithms. IEEE Trans. Inf. Theory. **57**(8), 5326–5341 (2011)
15. D Middleton, Non-Gaussian noise models in signal processing for telecommunications: new methods an results for class A and class B noise models. IEEE Trans. Inf. Theory. **45**(4), 1129–1149 (1999)
16. T Zhang, J Dan, G Gui, IMAC: Impulsive-mitigation adaptive sparse channel estimation based on Gaussian-mixture model. Comput Res Repository. **abs/1503.00800** (2015). https://arxiv.org/abs/1503.00800
17. GG Raleigh, JM Cioffi, Spatio-temporal coding for wireless communications. IEEE Global Telecommun. Conf. **3**, 1809–18143 (1996)
18. IE Telatar, Capacity of multi-antenna Gaussian channels. Eur. Trans. Telecommun. **10**, 585–595 (1999)
19. GJ Foschini, MJ Gans, On limits of wireless communications in a fading environment when using multiple antennas. Wirel. Pers. Commun. **6**, 311–335 (1998)
20. R Johannesson, K Zigangirov, *Fundamentals of convolutional coding*. (IEEE Press, New York, 1999)
21. RE Blahut, *Algebraic codes for data transmission*. (Cambridge University Press, New York, 2003)
22. JB Anderson, Limited search trellis decoding of convolutional codes. IEEE Trans. Inf. Theory. **35**(5), 944–955 (1989)
23. JE Gentle, *Matrix algebra: theory, computations, and applications in statistics*. (Springer, New York, 2007)
24. S Chen, XC Yang, L Hanzo, in *6th IEE International Conference on 3G and Beyond*. Blind joint maximum likelihood channel estimation and data detection for single-input multiple-output systems, (Washington, DC, 2005), pp. 1–5
25. SF Cotter, BD Rao, Sparse channel estimation via matching pursuit with application to equalization. IEEE Trans. Commun. **50**(3), 374–377 (2002)
26. F Xiong, A Zerik, E Shwedyk, Sequential sequence estimation for channels with intersymbol interference of finite or infinite length. IEEE Trans. Commun. **38**(6), 795–804 (1990)
27. G Mileounis, N Kalouptsidis, B Babadi, V Tarokh, in *17th IEEE International Conference on Digital Signal Processing (DSP)*. Blind identification of sparse channels and symbol detection via the EM algorithm, (Corfu, 2011), pp. 1–5
28. M Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*. (Springer, New York, 2010)
29. CR Berger, S Z, JC Preisig, P Willett, Sparse channel estimation for multicarrier underwater acoustic communication: from subspace methods to compressed sensing. IEEE Trans. Signal Process. **58**(3), 1708–1721 (2010)
30. J Huang, CR Berger, S Zhou, J Huang, Comparison of basis pursuit algorithms for sparse channel estimation in underwater acoustic OFDM. IEEE OCEANS. 1–6 (2010). http://ieeexplore.ieee.org/document/5603522/
31. W Zhou, Computationally efficient equalizer design. PhD thesis, George Mason University (2014)
32. W Zhou, JK Nelson, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. Sequential detection for sparse channels via a multiple tree algorithm, (Vancouver, 2013), pp. 4673–4677