ITERATIVE LEAST SQUARES DETECTION USING THE EM-ALGORITHM

Patrik Bohlin
Department of Signals and Systems,
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
E-mail: pbohlin@s2.chalmers.se

ABSTRACT

In this paper an algorithm based on Expectation-Maximization (EM) is derived for the problem of separating superimposed digitally modulated signals impinging on an antenna array. It is found that this algorithm closely resembles previously proposed methods based on Iterative Least Squares (ILS) techniques. Using an extension to the EM-algorithm known as SAGE, improvements to the algorithms are proposed for increasing performance and convergence rate as well as handling unknown noise covariance matrices. These improvements can in the ILS-framework be seen as doing Gauss-Seidel optimization instead of fix-point optimization as is done in the original algorithm.

INTRODUCTION

Increased capacity is one of the main design concerns in mobile communication systems. As a result, much research is targeted at increasing capacity by exploiting the spatial domain, using an antenna array at the base station. Typically, these methods use sophisticated signal processing techniques to obtain estimates of the transmitted signals by taking advantage of the additional degrees of freedom offered by several receiving antennas. Many algorithms rely on high resolution estimates of the direction of arrival of the signals in order to combine the contributions from the different antennas in an optimal way. This approach will be a challenging task in a wireless system since, due to multipath, both the signal of interest and the interfering signals will arrive randomly distributed in both time and space, and estimating all these paths will be prohibitively complex.

If little or no spatial knowledge is present, one solution is to rely on so-called property restoral algorithms, i.e. methods that make use of some known property of the unknown signal to be detected. One possibility is to use the fact that the symbols belong to a finite alphabet (such as BPSK). This approach has previously been explored under the name Iterative Least Squares (ILS) [1, 2]. The ILS-methods iterates between two steps: First, tentative symbols are used to get an estimate of the channel. This estimate is then used to get new estimates of the transmitted symbols (which are known to belong to the finite alphabet). In this paper we will derive a similar method based on the EM-framework. Improvements based on an extension to the EM-algorithm will also be proposed in order to reduce the computational load and increase the performance.

The Expectation-Maximization(EM) algorithm is a method that was first introduced in the statistics literature as an iterative procedure to produce Maximum Likelihood (ML)-parameter estimates [3]. EM is often useful when evaluation of the likelihood is made difficult by the absence of certain data. Suppose we wish to estimate a parameter $\boldsymbol{\theta} \in \Omega$ based on an observation $\mathbf{x}$, the ML-estimate would be $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} f(\mathbf{x};\boldsymbol{\theta})$. In some cases it is difficult or impossible to obtain a solution to this problem. It may on the other hand be possible to form a larger dataset $\mathbf{y}$ by appending a hidden or unknown variable to the observed data $\mathbf{x}$. We call $\mathbf{y}$ the *complete* data. If we let $\boldsymbol{\theta} \in \Omega$ denote the parameter vector of interest and if the Probability Density Function (PDF) $f_Y(\mathbf{y};\boldsymbol{\theta})$ is well defined, the PDF of $\mathbf{X}$ parameterized by $\boldsymbol{\theta}$ is given by $f_X(\mathbf{x};\boldsymbol{\theta}) = \int_{\mathcal{Y}(\mathbf{x})} f_Y(\mathbf{y};\boldsymbol{\theta})d\mathbf{y}$. In order to find the ML-estimate of the parameter $\boldsymbol{\theta}$ we seek to maximize the likelihood function $f_X(\mathbf{x};\boldsymbol{\theta})$ or equivalently the log-likelihood function $L(\boldsymbol{\theta}) = \log f_X(\mathbf{x};\boldsymbol{\theta})$. In order to this the EM-algorithm uses the expected value of the log-likelihood function for the complete data, conditioned on the observed data and the previous parameter estimate, i.e.

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{[k]}) = E\left[\log f(\mathbf{Y};\boldsymbol{\theta})|\mathbf{X}=\mathbf{x};\boldsymbol{\theta}^{[k]}\right]. \tag{1}$$

The function $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{[k]})$ is maximized iteratively with respect to the first parameter, given the second one. Each iteration $\boldsymbol{\theta}^{[k]} \rightarrow \boldsymbol{\theta}^{[k+1]}$ can then be described as follows:

E-step: Determine $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{[k]})$.

M-step: Choose $\boldsymbol{\theta}^{[k+1]} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta};\boldsymbol{\theta}^{[k]})$.

The algorithm keeps alternating between the two steps (Expectation and Maximization) until no significant improvement of $Q\left(\boldsymbol{\theta}^{[k+1]};\boldsymbol{\theta}^{[k]}\right)$ is observed. It can be proved that each step increases the likelihood, implying that the algorithm converges to a (local) maximum of the likelihood function.

As we will later see, this procedure closely resembles the ILS-algorithms. The expectation step can then be seen as the estimation step (i.e. estimating the channel) and the maximization step as the enumeration or projection step in the ILS-context. Due to space limitations we will only consider the case when one user at a time is treated, i.e., DWILSP [1] and not the joint multiuser case ILSE [2].

Despite the versatility of the EM-algorithm it has a few drawbacks, where the most severe are the slow convergence rate and the computational complexity of the M-step. To overcome these problems one solution that has been proposed is the Space Alternating Generalized EM (SAGE) [4]. This method is especially suited for problems where it is possible to sequentially update small groups of the parameter vector. An example is a communication system with an unknown channel, where the information bits can be updated sequentially. Instead of using one large complete-data set and update all parameters simultaneously, each group of parameters is assigned a *hidden*-space $\mathbf{Y}^S$, each a complete data set in the EM-sense if all other parameters were known. This leads to less informative data sets and hence a higher convergence rate.

The E-step in the SAGE algorithm is given by the following conditional expectation:

$$Q^S(\boldsymbol{\theta}_S;\boldsymbol{\theta}^{[k]}) = \mathrm{E}\left\{\log k(\mathbf{Y}^S;\boldsymbol{\theta}_S,\boldsymbol{\theta}_{S'}^{[k]})|\mathbf{X}=\mathbf{x};\boldsymbol{\theta}^{[k]}\right\} \tag{2}$$

Here S is used to indicate the subset of parameters that we are optimizing over, S' its complement, and $\mathbf{Y}^S$ is the corresponding hidden-space. The M-step is defined exactly as in the EM-case, but with reduced number of parameters. After finishing the two steps the algorithm then continues with the next set of parameters, until convergence.

SIGNAL MODEL

Assume that d users simultaneously transmit over a channel to an antenna array with M elements. The discrete-time complex output vector at time k can, after down conversion, matched filtering and symbol spaced sampling be written as

$$\mathbf{x}_k = \mathbf{H}\mathbf{s}_k + \mathbf{v}_k. \tag{3}$$

Here $\mathbf{H} = [\mathbf{h}_1 \dots \mathbf{h}_m]$ is the channel matrix of all the users (spatial signatures), $\mathbf{s}_k = [s_k^{(1)} \dots s_k^{(m)}]^T$, $s_k^{(i)} = \pm 1$ (BPSK), and $\mathbf{v}_k$ is spatially and temporally complex white Gaussian noise with variance $N_0/2$ in each dimension. By defining one signal (at a time) to be the signal of interest (SOI), a received block of length N, denoted by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$, can be written in the following way

$$\mathbf{X} = \mathbf{h}_1\mathbf{s}^{(1)} + \sum_{l=2}^{m}\mathbf{h}_l\mathbf{s}^{(l)} + \mathbf{V} = \mathbf{h}_1\mathbf{s}^{(1)} + \mathbf{J}. \tag{4}$$

Without loss of generality the first signal is taken to be the SOI. The term $\mathbf{J}$ corresponds to interfering signals plus noise and is modeled as a temporally white but spatially colored complex Gaussian process, i.e. $\mathbf{J}_j \in \mathcal{CN}(0,\mathcal{R}_{jj})$, $j = 1,\dots,N$.

The channel is assumed to have negligible delay spread and to exhibit slow fading, i.e the channel is constant during one block. The signal models for the two algorithms only differ in how the channel is modeled. In the EM-context the channel is modeled as a stochastic variable. The channel of each user is modeled as a zero mean Gaussian vector possibly correlated in space, but assumed independent among the users, i.e. $\mathbf{h} \in \mathcal{CN}(0,\mathcal{R}_{hh})$. This is the well known Rayleigh fading model which is widely used in the literature. In the ILS approach the channel is assumed deterministic but unknown.

EM BASED ALGORITHMS

By choosing the the complete data $\mathbf{Y}$ as the union of the observed data $\mathbf{X}$ and the unknown channel $\mathbf{h}$, the PDF for $\mathbf{Y}$ as a function of the parameter of interest, $\mathbf{s}$, can be written as the product between the likelihood

and the prior distribution of $\mathbf{h}$, i.e.,

$$f(\mathbf{Y};\mathbf{s}) = f(\mathbf{X},\mathbf{h};\mathbf{s}) = f(\mathbf{X}|\mathbf{h};\mathbf{s})f(\mathbf{h};\mathbf{s}). \tag{5}$$

Using the assumptions stated above $f(\mathbf{X}|\mathbf{h};\mathbf{s})$ and $f(\mathbf{h};\mathbf{s})$ are given by

$$f(\mathbf{X}|\mathbf{h};\mathbf{s}) = \frac{1}{\pi^{MN}|\mathcal{R}_{jj}|^N} \exp\left\{-\operatorname{Tr}\left[(\mathbf{X}-\mathbf{hs})^* \mathcal{R}_{jj}^{-1}(\mathbf{X}-\mathbf{hs})\right]\right\} \tag{6}$$

$$f(\mathbf{h};\mathbf{s}) = \frac{1}{\pi^{M}|\mathcal{R}_{hh}|} \exp\left\{-\mathbf{h}^*\mathcal{R}_{hh}^{-1}\mathbf{h}\right\}. \tag{7}$$

Assuming that the true covariance matrix $\mathcal{R}_{jj}$ is known, it is possible to evaluate the E-step. As we have conditioned on the received data $\mathbf{X}$, the expectation is only taken with respect to the channel $\mathbf{H}$. After some tedious but straight forward calculations [5] and by dropping terms not depending on $\mathbf{s}$ we have

$$Q(\mathbf{s};\mathbf{s}^{[k]}) = \mathrm{E}\left\{\log f(\mathbf{Y};\mathbf{s})|\mathbf{X};\mathbf{s}^{[k]}\right\} = -||\mathbf{X}-\hat{\mathbf{h}}\mathbf{s}||^2_{\mathcal{R}_{jj}^{-1}}, \tag{8}$$

. Here $\hat{\mathbf{h}}$ is the conditional mean or MMSE-estimate of the channel, given by

$$\hat{\mathbf{h}} = \left(\mathcal{R}_{jj}^{-1} + \mathcal{R}_{hh}^{-1}/N\right)^{-1} \mathcal{R}_{jj}^{-1}\mathbf{X}\mathbf{s}^{[k]^*}/N. \tag{9}$$

We can now see that if the number of samples tends to infinity the influence of the channel covariance $\mathcal{R}_{hh}$ disappears. If have a finite amount of data we can chose a flat prior distribution of the channel, i.e. $||\mathcal{R}_{hh}||_F \rightarrow \infty$. This choice of prior can be interpreted as a way to design an algorithm that will perform equally well for all realizations of $\mathbf{h}$. The channel estimate will tend to $\hat{\mathbf{h}} = \mathbf{X}\mathbf{s}^{[k]^*}/N$ and the M-step can be evaluated as

$$\mathbf{s}^{[k+1]} = \operatorname{sgn}\operatorname{Re}\left\{\hat{\mathbf{h}}^*\mathcal{R}_{jj}^{-1}\mathbf{X}\right\}. \tag{10}$$

We note that these are the steps performed by the DWILSP-algorithm [1] except that in [1] the sample covariance matrix was used as weighting matrix instead of the true noise covariance matrix. Nevertheless, it was shown that asymptotically it renders the same result. Hence, the DWILSP-algorithm can asymptotically be seen as an instance of the EM-algorithm.

Unknown Noise Covariance Matrix

We will now derive a way to properly (in the EM-context) handle the unknown covariance matrix. Here the covariance matrix is considered as a (nuisance) parameter that has to be estimated along with the signal. Following the SAGE-algorithm the parameters are divided into two sets, the signal and the covariance matrix. The hidden spaces for both sets are chosen as the complete space used in the EM-algorithm, i.e. $\mathbf{Y} = \{\mathbf{X},\mathbf{h}\}$. The approach is to iteratively estimate the covariance matrix and the user symbols. The evaluation of the E-step is done in a similar way as above, but the conditioning is on the previous estimate of the covariance matrix $\mathcal{R}_{jj}^{[k]}$ instead. This gives

$$Q(\mathcal{R}_{jj};\mathcal{R}_{jj}^{[k]},\mathbf{s}^{[k]}) = -N\log|\mathcal{R}_{jj}^{-1}| - \operatorname{Tr}\left\{\mathcal{R}_{jj}^{-1}(\mathbf{X}\mathbf{X}^* - N\hat{\mathbf{h}}\hat{\mathbf{h}}^* + \mathcal{R}_{jj}^{[k]})\right\} \tag{11}$$

and the following M-Step

$$\mathcal{R}_{jj}^{[k+1]} = \arg\max_{\mathcal{R}_{jj}}(Q(\mathcal{R}_{jj};\mathcal{R}_{jj}^{[k]},\mathbf{s}^{[k]})) = \hat{\mathcal{R}}_{xx} - \hat{\mathbf{h}}\hat{\mathbf{h}}^* + \frac{\mathcal{R}_{jj}^{[k]}}{N}. \tag{12}$$

Here, $\hat{\mathcal{R}}_{xx} = \mathbf{X}\mathbf{X}^*/N$ is the sample covariance matrix. The next step is to estimate the symbols. Given the estimate of $\mathcal{R}_{jj}$, the M-step (for the symbols) will be the same as in (10), but with $\mathcal{R}_{jj}$ replaced by $\mathcal{R}_{jj}^{[k]}$:

$$\mathbf{s}^{[k+1]} = \arg\max_{\mathbf{s}\in\{\pm1\}^N} Q(\mathbf{s};\mathbf{s}^{[k]}\mathcal{R}_{jj}^{[k]}) = \operatorname{sgn}\operatorname{Re}\left\{\hat{\mathbf{h}}^*\mathcal{R}_{jj}^{-1^{[k]}}\mathbf{X}\right\}. \tag{13}$$

Again $\hat{\mathbf{h}} = \mathbf{X}\mathbf{s}^{[k]^*}/N$ and $\hat{\mathcal{R}}_{jj}$ is given by (12). The algorithm iterates between estimating the covariance matrix and the symbols respectively, until convergence. Although this should be a proper way to handle the covariance matrix under finite frame length, no performance increase, compared to the original DWILSP-algorithm has been seen in any of the simulations conducted.

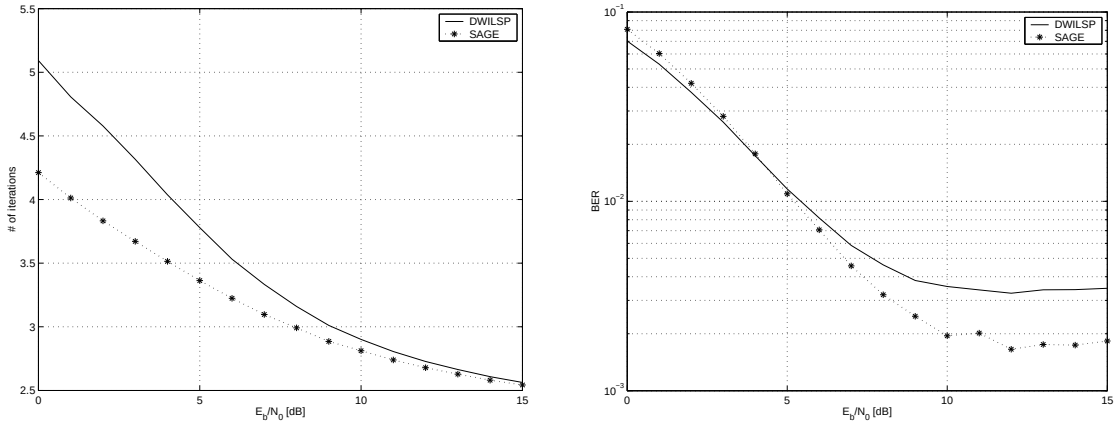

Figure 1: (Left) The number of iterations needed. (Right) The performance in terms of BER versus SNR. A four element uniform linear array is used. Three equally strong users, each transmitting frames of 100 symbols. 10 training bits were used to initialize the algorithm.

Convergence Improvements

Another way to utilize the SAGE-algorithm is to to derive an algorithm with faster convergence rate by estimating only one single symbol before updating the channel estimate, instead of updating the channel only after all symbols in one frame have been estimated. This requires us to define a hidden space for each symbol s_l. The natural hidden-data space for this parameter is the observed data at that time instant together with the channel, $\mathbf{Y}_l = \{\mathbf{X}_{:,l}, \mathbf{h}\}, \quad l = 1, \ldots, N$. Following the reasoning above the algorithm can be stated as:
for $l = \{1 + (k \mod N)\}$ do

$$\textbf{The E-step: } Q_l(\mathbf{s}_l; \mathbf{s}^{[k]}) = \mathrm{E}\left\{\log f(\mathbf{Y}_l; \mathbf{s}_l) | \mathbf{X}; \mathbf{s}^{[k]}\right\} = \mathrm{E}\left\{\log f(\mathbf{X}_{:,l}, \mathbf{h}; \mathbf{s}_l) | \mathbf{X}; \mathbf{s}^{[k]}\right\}$$

$$\textbf{The M-step: } \mathbf{s}_l^{[k+1]} = \arg\max_{\mathbf{s}_l \in \{\pm 1\}} Q_l(\mathbf{s}_l; \mathbf{s}^{[k]}) = \mathrm{sgn}\,\mathrm{Re}\left\{\hat{\mathbf{h}}^* \mathcal{R}_{jj}^{-1} \mathbf{X}_{:,l}\right\}$$

end

Using this approach might seem like a significant increase in complexity, since the $\mathbf{h}$-vector needs to be estimated in each step. Nevertheless, since only one symbol at a time might change, the channel estimate only has to be corrected if the symbol changed. In that case we do not need to explicitly evaluate the full matrix-vector multiplication, as was previously done, but instead only update the contribution from that time index. The computational load in each iteration is in fact even lower when the SAGE approach is used, since an update of $\mathbf{h}$ is only calculated if the corresponding bit has changed, and not for all bits as is done in the original version. In the original ILS-algorithms the proplem is posed as a fix point optimization problem. The algorithm derived here can then be intepreted as a non-linear Gauss-Seidel optimization.

In Fig. 1 the performance as well as the number of iterations needed for the two different algorithms are illustrated. As we can see from the figures, the number of iterations needed decrease by applying SAGE, and for high SNR we can also see a clear improvement in performance due to better convergence properties.

REFERENCES

[1] A. Ranheim. "A Decoupled Approach to Adaptive Signal Separation using an Antenna Array". *IEEE Transactions on Vehicular Technology*, pages 676–682, May 1999.

[2] S. Talwar, M. Viberg, and A. Paulraj. "Blind Separation of Synchronous Co-Channel Digital Signals Using an Antenna Array - Part I: Algorithms". *IEEE Transactions on Signal Processing*, Vol.44(5):1184–1197, May 1996.

[3] N. M. Dempster, A.P. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm". *J. R. Statist. Soc. B*, Vol. 39:185–197, 1977.

[4] J.A. Fessler and A.O. Hero. "Space-Alternating Generalized Expectation-Maximization Algorithm". *IEEE Transactions on Signal Processing*, Vol.42(10):2664–2677, October 1994.

[5] P. Bohlin. *Iterative Least Squares Techniques with Applications to Adaptive Antennas and CDMA-Systems*. Licentiate thesis, Chalmers Univ. of Tech., Göteborg, Sweden, 2001.