SPARSE BAYESIAN ESTIMATION OF SUPERIMPOSED SIGNAL PARAMETERS

Dmitriy Shutin, Gernot Kubin

Signal Processing and Speech Communication Laboratory
Graz University of Technology
Graz, Austria
dshutin@tugraz.at, gernot.kubin@tugraz.at

ABSTRACT

This paper addresses parameter estimation of superimposed signals jointly with their number within the Bayesian framework. We combine sparse Bayesian machine learning methods with the state of the art SAGE-based parameter estimation algorithm. Existing sparse Bayesian methods allow to assess model order through priors over model parameters, but do not consider models nonlinear in parameters. SAGE-based parameter estimation does allow nonlinear model structures, but lacks a mechanism for model order estimation. Here we show how Gaussian and Laplace priors can be applied to enforce sparsity and determine the model order in case of superimposed signals, as well as develop an EM-based learning algorithm that efficiently estimate parameters of the superimposed signals as well as prior parameters that control the sparsity of the learned models. Our work extends the existing approaches to complex data and models nonlinear in parameters. We also present new analytical and empirical studies of the Laplace sparsity priors applied to complex data. The performance of the proposed algorithm is analyzed using synthetic data.

Index Terms— Bayesian learning, evidence procedure, SAGE

1. INTRODUCTION

Design and analysis of state of the art communication systems equipped with sensor arrays often require accurate models, which reproduce in a realistic manner the structure and dynamics of the studied phenomenon. Over the last decade a significant amount of efforts has been put into the development of efficient estimation algorithms, capable to *jointly* estimate channel parameters, e.g., relative delays, Doppler frequencies, directions of the impinging wave fronts, etc [1–3]. However, joint estimation of the model parameters along with the number of superimposed signals (i.e., model order) is a particularly difficult task. Often the model order is simply fixed to a certain number. This approach does not always result in realistic models, specially in time-varying environments. Thus, we are typically forced to abandon the "joint" estimation concept. However, within the class of maximum-likelihood estimators it is possible to provide a mechanism that seamlessly incorporates the model selection scheme into the estimation framework. Estimation of the model order can be solved in the spirit of Occham's razor principle, i.e., several models are trained and then those that offer the best balance between model 'simplicity' (the smallest dimension of the parameter space), and model performance (the highest likelihood) are selected. Examples are the celebrated Akaike Information Criterion (AIC) [4], Minimum Description Length (MDL) principle and its variants [5]. Bayesian methods provide the ingredients required to jointly estimate signal parameters and their number.

Bayesian model order estimation, also referred to as sparse Bayesian learning, consists of smoothness or "simplicity" constraints, imposed on the model parameters [6, 7]. These constraints are usually specified in terms of appropriate parameter priors. In Relevance Vector Machines (RVM) [6] this prior is chosen to be zero-mean Gaussian. This choice of priors leads to analytically tractable estimation and was shown in [8] to be equivalent to the Schwarz's model selection [5]. However Gaussian priors are effective only under very special conditions, like high SNR or large number of observations, and require some post-learning rules, or thresholds to reenforce sparseness [8]. Alternatively, one can make use of Laplace priors that were shown to result in better sparsity enforcement [7, 9] at the expense of limited analytical tractability.

The above mentioned methods have been originally developed for sparse regression and classification problems, assuming models linear in parameters. In this paper we propose a sparse Bayesian parameter estimation algorithm that

- allows efficient estimation of model parameters that enter the model structure nonlinearly,
- accounts for non-white additive noise,
- assumes complex measurement data, and,
- unlike the above cited methods, does not require matrix inversions during the computations.

The paper is organized as follows: In Section 2 we introduce the signal model; Section 3 covers the learning algorithm itself, and, finally, Section 4 shows some application results for the simulated models.

2. SIGNAL MODEL

Let us assume that the receiver (Rx) is equipped with an antenna array consisting of P sensors located at $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{P-1} \in \mathbb{R}^2$ with respect to an arbitrary reference point. Let us also assume that the received signal can be represented as:

$$\boldsymbol{z}(t) = \sum_{l=1}^{L} w_l \boldsymbol{c}(\phi_l) R(t - \tau_l) + \boldsymbol{\xi}(t), \tag{1}$$

where $\boldsymbol{z}(t) \in \mathbb{C}^P$ is a vectorized representation of the sensor output at time t, L, is the number of superimposed signals, each having a complex gain w_l, relative delay τ_l, and arriving from a direction ϕ_l. The waveform $R(\cdot)$ incorporates the transmitted signal along with the influence of the transceiver front-end. The P-dimensional steering vector $\boldsymbol{c}(\phi_l)$ is represented as $\boldsymbol{c}(\phi_l) = [c_0(\phi_l), \ldots, c_{P-1}(\phi_l)]^T$,

and, assuming the coupling between the antenna sensors can be neglected, its components are given as

$$c_p(\phi_l) = f_p(\phi_l)\exp(j2\pi\lambda^{-1}\boldsymbol{e}^H(\phi_l)\boldsymbol{x}_p),$$

with λ, $\boldsymbol{e}(\phi_l)$, and $f_p(\phi_l)$ denoting the wavelength, the unit vector in $\mathbb{R}^2$ pointing in the direction ϕ_l, and the complex electric field pattern of the pth sensor, respectively. The additive noise $\boldsymbol{\xi}(t)$ is assumed to be a spatially white P-dimensional vector with each element being a zero-mean wide-sense stationary (WSS) complex Gaussian noise. In practice the output of the sensor array is sampled with the sampling period T_s, resulting in P N-tuples of the MF output, where N is the number of output samples. By stacking the sampled outputs of P sensors in one vector $\boldsymbol{z}$, (1) can be rewritten in the vector form as

$$\boldsymbol{z} = \sum_{l=1}^{L} w_l \boldsymbol{s}(\boldsymbol{\theta}_l) + \boldsymbol{\xi} = \boldsymbol{K}(\boldsymbol{\theta})\boldsymbol{w} + \boldsymbol{\xi}, \tag{2}$$

where we have defined $\boldsymbol{\theta}_l = [\phi_l, \tau_l]$, $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L]$, $\boldsymbol{r}_l = [R(-\tau_l), R(T_s - \tau_l), \ldots, R((N-1)T_s - \tau_l)]^T$,

$$\boldsymbol{s}(\boldsymbol{\theta}_l) = \begin{bmatrix} c_0(\phi_l)\boldsymbol{r}_l \\ \vdots \\ c_{P-1}(\phi_l)\boldsymbol{r}_l \end{bmatrix}, \quad \boldsymbol{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_L \end{bmatrix}, \quad \boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\xi}_0 \\ \vdots \\ \boldsymbol{\xi}_{P-1} \end{bmatrix}, \tag{3}$$

and $\boldsymbol{\xi}_p = [\xi_p(0), \ldots, \xi_p((N-1)T_s)]^T$. We will assume that

$$E\{\boldsymbol{\xi}_p\} = \boldsymbol{0}, E\{\boldsymbol{\xi}_m\boldsymbol{\xi}_k^H\} = \boldsymbol{0}, \text{for } m \neq k, \text{ and} \tag{4}$$

$$E\{\boldsymbol{\xi}_p\boldsymbol{\xi}_p^H\} = \boldsymbol{\Sigma}_p. \tag{5}$$

The goal of the learning algorithm is to estimate the model parameters which are: the order of the model L and parameters $\{w_l, \boldsymbol{\theta}_l\}_{l=1}^L$. Note that we treat the model parameters $\boldsymbol{\theta}$ and $\boldsymbol{w}$ separately. As we will show, it is $\boldsymbol{w}$ that are used to control the model complexity: by setting some of them to zero we realize model selection [1]. Parameters $\boldsymbol{\theta}$, on the other hand, are used to improve the fit of the model to the measured data, and thus control the model order only indirectly.

3. LEARING ALGORITHM

Before we begin explaining the estimation algorithm, let us now outline the probabilistic structure of the variables involved in the analysis. From (2) it follows that $\boldsymbol{z}|\boldsymbol{w}, \boldsymbol{\theta}$ is complex Gaussian, with the mean $\boldsymbol{K}(\theta)\boldsymbol{w}$ and covariance matrix $\boldsymbol{\Sigma}$, i.e.,

$$\boldsymbol{z}|\boldsymbol{w}, \boldsymbol{\theta} \sim \mathcal{CN}(\boldsymbol{K}(\theta)\boldsymbol{w}, \boldsymbol{\Sigma}).$$

When additive noise is spatially white, $\boldsymbol{\Sigma}$ is simply a block-diagonal covariance matrix $\boldsymbol{\Sigma} = \text{diag}\{\boldsymbol{\Sigma}_0, \ldots, \boldsymbol{\Sigma}_{P-1}\}$.

The sparsity is enforced through the zero-mean prior $p(\boldsymbol{w}|\boldsymbol{\alpha})$ over the model coefficients $\boldsymbol{w}$. Parameters $\boldsymbol{\alpha}$, also called evidence parameters, or hyperparameters, are inversely proportional to the width of the corresponding pdf. Large values of α_l render the contribution of the corresponding column in the matrix $\boldsymbol{K}(\boldsymbol{\theta})$ 'irrelevant', since the corresponding weights w_l are then likely to have a very small value. We will consider the learning algorithm for the cases when $p(\boldsymbol{w}|\boldsymbol{\alpha})$ is either Gaussian, or Laplace pdf.

[1] The considered approach allows to control the complexity of the model by removing some of the contributions $\boldsymbol{s}(\boldsymbol{\theta}_l)$. The learning algorithm, however, does allow to increase the model order if necessary, but the development of this idea stays outside the scope of this paper.

In order to stay within the Bayesian framework we also need to define the hyperprior $p(\boldsymbol{\alpha})$. To avoid additional free parameters, we assume this prior to be noninformative, i.e., flat, which corresponds to the automatic relevance determination (ARD) concept, proposed in [10, 11]. Similarly, we chose a prior $p(\boldsymbol{\theta})$ to be flat. However, as we will see, the latter does not have a dramatic effect on the learning algorithm and more elaborated priors can be easily integrated. This completes the description of densities involved in the algorithm.

3.1. EM-based learning algorithm

Our ultimate goal is to obtain the model parameters $\{\boldsymbol{w}, \boldsymbol{\theta}\}$, and hyperparameters $\boldsymbol{\alpha}$ that maximize the posterior $p(\boldsymbol{w}, \boldsymbol{\theta}, \boldsymbol{\alpha}|\boldsymbol{z})$. We rewrite it as $p(\boldsymbol{w}, \boldsymbol{\theta}, \boldsymbol{\alpha}|\boldsymbol{z}) = p(\boldsymbol{w}, \boldsymbol{\theta}|\boldsymbol{\alpha}, \boldsymbol{z})p(\boldsymbol{\alpha}|\boldsymbol{z})$ and maximize each term on the right-hand side sequentially from right to left, which is known as the *marginal estimation method* [12, ch. 5]. The term $p(\boldsymbol{w}, \boldsymbol{\theta}|\boldsymbol{\alpha}, \boldsymbol{z})$ is of interest when we only estimate model parameters assuming fixed model order. The second term $p(\boldsymbol{\alpha}|\boldsymbol{z})$ – the "penalty", comes into play when the model order is to be estimated.

We begin with the maximization of $p(\boldsymbol{w}, \boldsymbol{\theta}|\boldsymbol{\alpha}, \boldsymbol{z})$ assuming that $\boldsymbol{\alpha}$ is known and fixed. Using Bayes theorem we rewrite this posterior as

$$p(\boldsymbol{w}, \boldsymbol{\theta}|\boldsymbol{\alpha}, \boldsymbol{z}) = \frac{p(\boldsymbol{z}|\boldsymbol{w}, \boldsymbol{\theta})p(\boldsymbol{w}|\boldsymbol{\alpha})p(\boldsymbol{\theta})}{p(\boldsymbol{z}|\boldsymbol{\alpha})} \tag{6}$$

and maximize the numerator term on the right-hand side. This is a classical parameter estimation approach, and if it were not for the $\boldsymbol{\theta}$ that nonlinearly enters the likelihood $p(\boldsymbol{z}|\boldsymbol{w}, \boldsymbol{\theta})$, the optimal solution would have been trivial. Here, in order to facilitate the optimization we appeal to the EM algorithm [13]. The major steps of this algorithm are summarized below.

E-Step. As an unobserved data we chose $\boldsymbol{x} = [\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_L^T]^T$ where

$$\boldsymbol{x}_l = w_l\boldsymbol{s}(\boldsymbol{\theta}_l) + \boldsymbol{\xi}_l, \; l = 1, \ldots, L, \tag{7}$$

and $\boldsymbol{\xi}_l$ are obtained by arbitrarily decomposing the total noise $\boldsymbol{\xi}$ such that $\boldsymbol{\xi} = \sum_{l=1}^L \boldsymbol{\xi}_l$. It follows that $\boldsymbol{x}_l$ is conditionally Gaussian, i.e., $\boldsymbol{x}_l|w_l, \boldsymbol{\theta}_l \sim \mathcal{N}(w_l\boldsymbol{s}(\boldsymbol{\theta}_l), \boldsymbol{\Sigma}_l)$ where $\boldsymbol{\Sigma}_l = E\{\boldsymbol{\xi}_l\boldsymbol{\xi}_l^H\} = \delta_l\boldsymbol{\Sigma}$, and $\delta > 0$ is chosen so that $\sum_l \delta_l = 1$. Based on (7) the Q-function $Q(\boldsymbol{\theta}, \boldsymbol{w}|\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{w}}) = E\Big\{ \log[p(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{\theta})]\big|\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{w}}, \boldsymbol{z}\Big\}$ is given as [13]:

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{w}|\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{w}}) &= \sum_l Q_l(\boldsymbol{\theta}_l, w_l|\hat{\boldsymbol{\theta}}_l, \hat{w}_l) = \\ & c - \sum_{l=1}^{L}(\hat{\boldsymbol{x}}_l - w_l\boldsymbol{s}(\boldsymbol{\theta}_l))^H\boldsymbol{\Sigma}_l^{-1}(\hat{\boldsymbol{x}}_l - w_l\boldsymbol{s}(\boldsymbol{\theta}_l)) \end{aligned} \tag{8}$$

where c is a constant independent of $\boldsymbol{\theta}$ and $\boldsymbol{w}$, and $\hat{\boldsymbol{x}}_l$ is given as

$$\hat{\boldsymbol{x}}_l = \hat{w}_l\boldsymbol{s}(\hat{\boldsymbol{\theta}}_l) + \delta_l(\boldsymbol{z} - \boldsymbol{K}(\hat{\boldsymbol{\theta}})\hat{\boldsymbol{w}}), \tag{9}$$

with $\hat{\boldsymbol{w}}$ and $\hat{\boldsymbol{\theta}}$ being some current parameter estimates. We now see that the maximization of (8) with respect to $\boldsymbol{\theta}$ and $\boldsymbol{w}$ is equivalent to L smaller optimizations of $Q_l(\boldsymbol{\theta}_l, w_l|\hat{\boldsymbol{\theta}}_l, \hat{w}_l)$ with respect to $\boldsymbol{\theta}_l$ and w_l only. This is how the matrix inversions, usually appearing during the maximization of (6), can be avoided.

M-Step. Since parameters $\boldsymbol{\theta}_l$ enter the Q-function nonlinearly, the M-step can be solved using the SAGE algorithm [14], i.e., we suggest to update one parameter (or a subset of parameters) at a time, while keeping the other fixed:

$$\hat{\boldsymbol{\theta}}_l' \leftarrow \underset{\theta_l}{\text{argmax}}\, Q_l'(\boldsymbol{\theta}_l, \hat{w}_l|\hat{\boldsymbol{\theta}}_l, \hat{w}_l), \tag{10}$$

$$w_l' \leftarrow \underset{w_l}{\operatorname{argmax}}\, Q_l'(\hat{\boldsymbol{\theta}}_l', w_l|\hat{\boldsymbol{\theta}}_l, \hat{w}_l) \tag{11}$$

where

$$Q_l'(\boldsymbol{\theta}_l, w_l|\hat{\boldsymbol{\theta}}_l, \hat{w}_l) = Q_l(\boldsymbol{\theta}_l, w_l|\hat{\boldsymbol{\theta}}_l, \hat{w}_l) + \log[p(w_l|\alpha_l)p(\boldsymbol{\theta}_l)],$$

and $\hat{\boldsymbol{\theta}}_l'$ and $\hat{w}_l'$ are improved versions of the $\hat{\boldsymbol{\theta}}_l$ and $\hat{w}_l$, respectively.

3.2. Sparsity priors

Inferring hyperparameters constitutes the essence of the model selection. This is achieved through the "penalty" posterior

$$p(\boldsymbol{\alpha}|\boldsymbol{z}) \propto p(\boldsymbol{z}|\boldsymbol{\alpha})p(\boldsymbol{\alpha}). \tag{12}$$

Since $p(\boldsymbol{\alpha})$ is assumed to be flat, the hyperparameters $\boldsymbol{\alpha}$ that maximize $p(\boldsymbol{\alpha}|\boldsymbol{z})$ can be found by maximizing the $p(\boldsymbol{z}|\boldsymbol{\alpha})$ alone. The latter term is known as *evidence*. Maximization of the evidence with respect to $\boldsymbol{\alpha}$ can also be accomplished by means of complete data $\boldsymbol{x}$, which also leads to the simpler optimization procedure. Indeed,

$$p(\boldsymbol{x}|\boldsymbol{\alpha}) = \int\int p(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{\theta})p(\boldsymbol{w}|\boldsymbol{\alpha})p(\boldsymbol{\theta})d\boldsymbol{w}d\boldsymbol{\theta} \tag{13}$$

can be used to locally maximize the evidence $p(\boldsymbol{z}|\boldsymbol{\alpha})$. However, the nonlinear dependency of $p(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{\theta})$ on $\boldsymbol{\theta}$ will still cause difficulties in solving the integral (13). To simplify the computation we adopt several approximations. First we assume that $\boldsymbol{\theta}$ is fixed at $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}'$, which makes $\boldsymbol{x}$ functionally independent of $\boldsymbol{\theta}$. Thus

$$p(\boldsymbol{x}|\boldsymbol{\alpha}) \approx \int p(\boldsymbol{x}|\boldsymbol{w}, \hat{\boldsymbol{\theta}}')p(\boldsymbol{w}|\boldsymbol{\alpha})d\boldsymbol{w}. \tag{14}$$

Integral (14) can be solved using Laplace approximation method [5], which consists in approximating the integrand using second order Taylor series around $\hat{\boldsymbol{w}}'$ and then computing the integral.

3.2.1. Gaussian prior

Should the prior $p(\boldsymbol{w}|\boldsymbol{\alpha})$ be chosen to be complex Gaussian, i.e., $\boldsymbol{w}|\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{0}, \operatorname{diag}\{\boldsymbol{\alpha}\}^{-1})$, then Laplace approximation is exact:

$$p(\boldsymbol{x}|\boldsymbol{\alpha}) = p(\boldsymbol{x}|\hat{\boldsymbol{w}}', \hat{\boldsymbol{\theta}}')p(\hat{\boldsymbol{w}}'|\boldsymbol{\alpha})\pi^L|\boldsymbol{\Phi}|, \tag{15}$$

where $\boldsymbol{\Phi}$ is a diagonal matrix with the lth element on the main diagonal given as $\Phi_{ll} = (\alpha_l + \boldsymbol{s}(\hat{\boldsymbol{\theta}}')^H\boldsymbol{\Sigma}_l^{-1}\boldsymbol{s}(\hat{\boldsymbol{\theta}}'))^{-1}$. Note, that $\boldsymbol{\Phi}$ becomes diagonal only thanks to the unobserved data $\boldsymbol{x}$. The same approach applied to the incomplete data $\boldsymbol{z}$ would result in the full matrix $\boldsymbol{\Phi}$, and thus computationally heavier matrix inversion. Now, by taking the logarithm of (15), and setting the partial derivatives with respect to α_l to zero, we find

$$\alpha_l = \frac{1}{|\hat{w}_l'|^2 + \Phi_{ll}}, \tag{16}$$

where $\hat{w}_l' = \Phi_{ll}\boldsymbol{s}(\boldsymbol{\theta}_l')\boldsymbol{\Sigma}_l^{-1}\hat{\boldsymbol{x}}_l$ is a closed-form solution to (11) in case of Gaussian sparsity priors. Note, that Gaussian priors need additional thresholds to decide when α_l is large enough to remove the corresponding contribution. A useful approach that requires only the statistics of the additive noise $\boldsymbol{\xi}$ has been proposed in [8]. Once the value of α_l is computed, we can decide if the corresponding contribution should remain in the model, or should be pruned, thus implementing model order estimation.

3.2.2. Laplace priors

In case when $p(\boldsymbol{w}|\boldsymbol{\alpha})$ is complex Laplace, i.e.,

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \prod_{l=1}^{L}\frac{2\alpha_l^2}{\pi}\exp\{-2\alpha_l|w_l|\},$$

the Laplace approximation to the evidence integral (14) is no longer exact. The resulting approximation has the form identical to (15), but for Laplace priors Φ_{ll} is given as

$$\Phi_{ll} = \left(\frac{\boldsymbol{s}(\hat{\boldsymbol{\theta}}_l')^T\boldsymbol{\Sigma}_l^{-1}\boldsymbol{s}(\hat{\boldsymbol{\theta}}_l')}{2} + \frac{\alpha_l}{4|\hat{w}_l|}\right)^{-1}, \tag{17}$$

where

$$\hat{w}_l' = \frac{\operatorname{sign}(\boldsymbol{s}(\hat{\boldsymbol{\theta}}_l')^T\boldsymbol{\Sigma}_l^{-1}\hat{\boldsymbol{x}}_l)(|\boldsymbol{s}(\hat{\boldsymbol{\theta}}_l')^T\boldsymbol{\Sigma}_l^{-1}\hat{\boldsymbol{x}}_l| - \alpha_l)_+}{\boldsymbol{s}(\hat{\boldsymbol{\theta}}_l')^T\boldsymbol{\Sigma}_l^{-1}\boldsymbol{s}(\hat{\boldsymbol{\theta}}_l')}. \tag{18}$$

In (18) $\operatorname{sign}(\cdot)$ is a sign function defined as $\operatorname{sign}(x) = x/|x|$, and $(\cdot)_+$ is a positive part operator defined as: $(a)_+ = a$, if $a \geq 0$, and $(a)_+ = 0$, if $a < 0$. Here, unlike the Gaussian prior case, there is no need to define any additional thresholds– the hyperparameter plays the role of such a threshold directly. It can be found by equating to zero the partial derivatives of (15) with respect to α_l, which, using (17) and (18), leads to hyperparameter update expression:

$$\alpha_l = \frac{1}{|\hat{w}_l'| + \Phi_{ll}/8|\hat{w}_l'|}. \tag{19}$$

Once α_l is found, evaluation of the (18) will automatically remove the "irrelevant" contributions.

The last ingredient that we need in our algorithm is an initialization strategy.

3.3. Algorithm initialization

As with any EM-based estimation algorithm, proper initialization plays an important role here. Initialization includes defining the initial number of components L as well as the corresponding model parameters $\{\boldsymbol{w}, \boldsymbol{\theta}\}$, and hyperparameters $\boldsymbol{\alpha}$. We also need to chose constants δ_l to evaluate the complete data $\boldsymbol{x}_l$ in (9). Below, we propose a simple initialization procedure that deduce the initial values from the measured data $\boldsymbol{z}$. First, we need to define the initial number L of contributions in the model (2). According to our model selection strategy, L should be initially chosen to overestimate the true model order so that the "irrelevant" contributions could be pruned at the learning stages of the algorithm. Note that the complexity of our algorithm increases only linearly with the number of contributions. Concerning the selection of the factors δ_l, we refer to [2] where it was shown that by choosing $\delta_l = 1$ the convergence rate of the algorithm is maximized. Initialization of other parameters is a bit more elaborated. We begin by setting w_l and θ_l for all L components to zero. Then, a single iteration of the learning algorithm is performed, i.e., (9) followed by (10), with the latter optimization performed incoherently for each element of $\boldsymbol{\theta}_l$.

In case of Gaussian sparsity priors, hyperparameters α_l are initialized as in [8]: if $|\boldsymbol{s}(\hat{\boldsymbol{\theta}}_l)^H\boldsymbol{\Sigma}_l^{-1}\hat{\boldsymbol{x}}_l)|^2 > \boldsymbol{s}(\hat{\boldsymbol{\theta}}_l)^H\boldsymbol{\Sigma}_l^{-1}\boldsymbol{s}(\hat{\boldsymbol{\theta}}_l)$, then $\alpha_l = \frac{(\boldsymbol{s}(\hat{\boldsymbol{\theta}}_l)^H\boldsymbol{\Sigma}_l^{-1}\boldsymbol{s}(\hat{\boldsymbol{\theta}}_l))^2}{(|\boldsymbol{s}(\hat{\boldsymbol{\theta}}_l)^H\boldsymbol{\Sigma}_l^{-1}\hat{\boldsymbol{x}}_l)|^2 - \boldsymbol{s}(\hat{\boldsymbol{\theta}}_l)^H\boldsymbol{\Sigma}_l^{-1}\boldsymbol{s}(\hat{\boldsymbol{\theta}}_l))}$ [8, eq. (29)]; otherwise, the corresponding basis $\boldsymbol{s}(\hat{\boldsymbol{\theta}}_l)$ is pruned at the initialization stage. Having found α_l, the gain w_l is estimated using (11).

SNR	L	RMSE		$\mathrm{Avr}\{\hat{L}\}$	
		Gaussian	Laplace	Gaussian	Laplace
5dB	$L=3$	0.0493	0.1095	2.80	17.82
	$L=9$	0.1120	0.1241	7.75	22.87
15dB	$L=3$	0.0164	0.0352	3.13	18.82
	$L=9$	0.0364	0.0422	9.19	28.06
25dB	$L=3$	0.0075	0.0123	3.75	23.19
	$L=9$	0.0179	0.0161	12.64	44.08

Table 1: Averaged channel estimation performance.

In case of Laplace priors, the initial values of hyperparameter α_l are first set to zero. Then, (18) is evaluated to obtain initialization $\hat{w}_l$, followed by the estimation of the corresponding hyperparameter α_l according to (19).

4. SIMULATION RESULTS

We begin demonstaring performance of the algorithm with the synthetic data, generated according to the model (1). It is assumed that the data is recorded over the time window $T = 0.31\mu sec$ and sampled at $1/T_s = 400$MHz. Signal parameters are chosen by drawing L samples from the corresponding distributions: delays τ_l and angles ϕ_l are drawn uniformly from the interval $[0.1, 0.2]\mu sec$ and $[-\pi/2, \pi/2]$, respectively. Signal gains w_l are generated as $w_l = \mathrm{e}^{j\psi_l}$, where ψ_l is uniformly distributed in the interval $[0, 2\pi]$. This ensures that all components have the same power, and thus equal chance of being detected under the same noise constraints. The initial number of components is set equal to the number of available signal samples. The performance of the algorithm is assessed based on the averaged number of detected components as well as on the achieved RMSE between the synthetic and reconstructed sensor data. The corresponding simulation results are summarized in the Table 1.

From the Table 1 we can see that with Gaussian priors and additional thresholding the resulting models are sufficiently sparse, and approximate well the simulated channel. Laplace priors on the other hand do not achieve the same sparseness and the number of estimated components is significantly overestimated. This clearly indicates that the Laplace approximation to the evidence integral (14) is not adequate, i.e., the resulting values do not lead to the maximum of the evidence. We have observe that increasing empirically the value of the hyperparameter does lead to the improvement in the estimation performance. Thus, additional rules to compensate for the insufficient approximation of the evidence might improve the algorithm performance.

5. CONCLUSIONS

We have proposed an estimation algorithm that unites sparse Bayesian learning with SAGE-based parameter estimation algorithm. The proposed approach allows to efficiently estimate parameters of linearly superimposed complex signals, as well as hyperparameters through which the sparsity of the model is controlled and model order estimation is realized. We have obtained the closed-form expressions for estimating hyperparameters that control the sparsity of the learned model. Thus, estimation of hyperparameters along with the other signal parameters does not increase much the computational complexity, but does allow to jointly estimate the model order. The obtained analytical results for Gaussian priors extend the previous works to models nonlinear in parameters. The estimation results do show that Gaussian priors with post-learning pruning can estimation model parameters as well as the model order quite well. We have also shown some new analytical results of using Laplace priors with complex data to enforce sparsity. However, Laplace approximations we use to obtain the closed-form expression for the hyperparameter update does not approximate the maximum of the corresponding evidence integral. This indicates a need to either invent corrections rules, which will compensate for the insufficiency of the Laplace approximations, or find another computationally efficient way to maximize the evidence integral.

6. REFERENCES

[1] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE Signal Processing Mag.*, pp. 67–94, July 1996.

[2] B.H. Fleury, M. Tschudin, R. Heddergott, D. Dahlhaus, and K. I. Pedersen, "Channel parameter estimation in mobile radio environments using the SAGE algorithm," *IEEE Journal on Sel. Areas in Comm.*, vol. 17, no. 3, pp. 434–450, March 1999.

[3] O. Besson and P. Stoica, "Decoupled estimation of DOA and angular spread for a spatially distributed source," *IEEE Trans. on Sig Proc.*, vol. 48, no. 7, pp. 1872–1882, 2000.

[4] H. Akaike, "A new look at the statistical model identification," *Trans. on Autom. Control*, vol. 19, no. 6, pp. 716–723, Dec. 1974.

[5] A. Lanterman, "Schwarz, Wallace, and Rissanen: Intertwining themes in theories of model order estimation," *International Statistical Review*, vol. 69, no. 2, pp. 185–212, 2000.

[6] Michael Tipping, "Sparse Bayesian Learning and The Relevance Vector Machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, June 2001.

[7] M.A.T. Figueiredo, "Adaptive sparseness for supervised learning," *IEEE Trans. on Pattern Analysis and Machine Intel.*, vol. 25, no. 9, pp. 1150–1159, 2003.

[8] Dmitriy Shutin, Gernot Kubin, and Bernard H. Fleury, "Application of the Evidence procedure to the analysis of wireless channels," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–23, 2007.

[9] Peter M. Williams, "Bayesian regularisation and pruning using a Laplace prior," Tech. Rep. CSRP-312, University of Sussex, February 1994.

[10] R.M. Neal, *Bayesian Learning for Neural Networks*, vol. 118 of *Lecture Notes in Statistics*, New York: Springer-Verlag, 1996.

[11] D. J. C. MacKay, "Bayesian Methods for Backpropagation Networks," in *Models of Neural Networks III*, E. Domany, J. L. van Hemmen, and K. Schulten, Eds., chapter 6, pp. 211–254. Springer-Verlag, New York, 1994.

[12] Simon Haykin, Ed., *Kalman Filtering and Neural Networks*, John Wiley & Sons, Inc., 2001.

[13] M. Feder and E. Weinstein, "Parameter Estimation of Superimposed Signals Using the EM Algorithm," *IEEE Trans. on Acoustics, Speech, and Sig. Proc.*, vol. 36, no. 4, pp. 477–489, April 1988.

[14] J.A. Fessler and A.O. Hero, "Space-alternating generalized Expectation-Maximization algorithm.," *IEEE Transactions on Signal Processing*, vol. 42, pp. 2664–2677, Oct. 1994.