FAST ADAPTIVE VARIATIONAL SPARSE BAYESIAN LEARNING WITH AUTOMATIC RELEVANCE DETERMINATION

Dmitriy Shutin† Thomas Buchgraber* Sanjeev R. Kulkarni† H. Vincent Poor†

† Department of Electrical Engineering, Princeton University, USA
* Signal Processing and Speech Comm. Lab., Graz University of Technology, Austria

ABSTRACT

In this work a new adaptive fast variational sparse Bayesian learning (V-SBL) algorithm is proposed that is a variational counterpart of the fast marginal likelihood maximization approach to SBL. It allows one to adaptively construct a sparse regression or classification function as a linear combination of a few basis functions by minimizing the variational free energy. In the case of non-informative hyperpriors, also referred to as automatic relevance determination, the minimization of the free energy can be efficiently realized by computing the fixed points of the update expressions for the variational distribution of the sparsity parameters. The criteria that establish convergence to these fixed points, termed pruning conditions, allow an efficient addition or removal of basis functions; they also have a simple and intuitive interpretation in terms of a component’s signal-to-noise ratio. It has been demonstrated that this interpretation allows a simple empirical adjustment of the pruning conditions, which in turn improves sparsity of SBL and drastically accelerates the convergence rate of the algorithm. The experimental evidence collected with synthetic data demonstrates the effectiveness of the proposed learning scheme.

1. INTRODUCTION

During the past decade, research on sparse signal representations has received considerable attention [1–5]. With a few minor variations, the general goal of sparse reconstruction is to optimally estimate the parameters of the following canonical model:

\[ t = \Phi w + \xi, \]

where \( t \in \mathbb{R}^N \) is a vector of targets, \( \Phi = [\phi_1, \ldots, \phi_L] \) is a design matrix with \( L \) columns corresponding to basis functions \( \phi_l \in \mathbb{R}^N \), \( l = 1, \ldots, L \), and \( w = [w_1, \ldots, w_L]^T \) is a vector of weights that are to be estimated. The additive perturbation \( \xi \) is typically assumed to be a white Gaussian random vector with zero mean and covariance matrix \( \Sigma = \tau^{-1}I \), where \( \tau \) is a noise precision parameter. Imposing constraints on the model parameters \( w \) is key to sparse signal modeling [3].

In sparse Bayesian learning (SBL) [2, 4, 6] the weights \( w \) are constrained using a parametric prior probability density function (pdf) \( p(w|\alpha) = N(w|0, \text{diag}(\alpha)^{-1}) \), with the prior parameters \( \alpha = [\alpha_1, \ldots, \alpha_L]^T \), also called sparsity parameters, being inversely proportional to the width of the pdf. Naturally, a large value of \( \alpha_l \) will drive the corresponding weight \( w_l \) to zero, thus encouraging a solution with only a few nonzero coefficients.

In the relevance vector machine (RVM) approach to the SBL problem [2] the sparsity parameters \( \alpha \) are estimated by maximizing the marginal likelihood \( p(t|\alpha, \tau) = \int p(t|w, \tau)p(w|\alpha)dw \), which is also termed model evidence [2, 6]; the corresponding estimation approach is then referred to as the Evidence Procedure (EP) [2]. Unfortunately, the RVM solution is known to converge rather slowly and the computational complexity of the algorithm scales as \( O(L^3) \) [2, 7]; this makes the application of RVMs to large data sets impractical. In [7] an alternative learning scheme was proposed to alleviate this drawback. This scheme exploits the structure of the marginal likelihood function to accelerate the maximization via a sequential addition and deletion of candidate basis functions, thus allowing efficient implementations of SBL even for “very wide” matrices \( \Phi \).

An alternative approach to SBL is based on approximating the posterior \( p(w, \tau, \alpha|t) \) with a variational proxy pdf \( q(w, \tau, \alpha) = q(w|\tau)q(\tau|\alpha) \) [8] such as to minimize the variational free energy [9]. There are several advantages of the variational solution to SBL as compared to that proposed in [2] and [7]: first, the distributions rather than point estimates of the unobserved variables can be obtained. Second, the variational approach to SBL allows one to obtain analytical approximations to the distributions of interest even when exact inference of these distributions is intractable. Finally, the variational methodology provides a general tool for inference on graphical models that represent extensions of (1), e.g., different priors, parametric design matrices, etc. Unfortunately, the variational approach in [8] is equivalent to RVM in terms of estimation complexity and rate of convergence. Also, due to the nature of the variational approximation, it is no longer possible to exploit the structure of the marginal likelihood function to implement the learning more efficiently: the pdfs \( q(\alpha) \) and \( q(\tau) \) are estimated such as to approximate the true posterior pdfs, thus obscuring the structure of the marginal likelihood that was exploited in [7].

Nonetheless, it can be shown [10] that by computing the fixed points of the update expressions for the variational parameters of \( q(\alpha) \) one can establish a dependency between the optimum of the variational free energy and a sparsity parameter of a single basis function \( \phi_l \). Our goal in this paper is to extend these results by constructing a fast adaptive variational SBL scheme that can be used to implement adaptive SBL by allowing deletion and addition of new basis functions. We show that the criteria that guarantee the convergence of a sparsity parameter for a basis function, which we term the pruning condition, can be used either to prune or to add new basis functions. The computation of the pruning conditions requires knowing only the target vector \( t \) and the posterior covariance matrix of the weights \( w \). Moreover, the proposed adaptive scheme requires only \( O(L^2) \) operations for adding or deleting a basis function. We
2. VARIATIONAL SPARSE BAYESIAN LEARNING

For the purpose of further analysis let us assume that we have a dictionary \( \mathcal{D} \) of some potential basis functions. \( \mathcal{D} \) is assumed to consist of an atomic dictionary \( \Phi \) used in (1) and a passive dictionary \( \Phi^C \) such that \( \mathcal{D} = \Phi \cup \Phi^C \) and \( \Phi \cap \Phi^C = \emptyset \).

In SBL it is assumed that the joint pdf of all the variables factors as \( p(w, \tau, \alpha, t) = p(t|w, \tau)p(w|\alpha)p(\alpha)p(\tau) \) [2, 4, 6]. Under the Gaussian noise assumption, \( p(t|w, \tau) = N(t|\Phi w, \tau^{-1} I) \). The sparsity prior \( p(w|\alpha) \) is assumed to factor as \( p(w|\alpha) = \prod_{l=1}^{L} p(w_l|\alpha_l) \), where \( p(w_l|\alpha_l) = N(w_l|0, \alpha_l^{-1}) \). The choice of the prior \( p(\tau) \) is arbitrary in the context of this work; a convenient choice would be a gamma distribution due to conjugacy properties, e.g., \( p(\tau) = \Gamma(\tau|\varepsilon, d) \). The prior \( p(\alpha_l) \) is also called the hyperprior of the \( l \) th component, is selected as a gamma pdf \( \Gamma(\alpha_l|\hat{a}_l, \hat{b}_l) \). We will however consider an automatic relevance determination scenario, obtained when \( a_l = b_l = 0 \) for all components; this choice renders the hyperpriors non-informative [2, 6].

The variational solution to SBL is obtained by finding an approximating pdf \( q(w, \alpha, \tau) = q(w)q(\tau) \prod_{l=1}^{L} q(\alpha_l) \), where \( q(w) = N(w|\bar{w}, \bar{S}) \), \( q(\alpha) = \Gamma(\alpha|\hat{a}_l, \hat{b}_l) \), and \( q(\tau) = \Gamma(\tau|\varepsilon, d) \) are the variational approximating factors. With this choice of \( q(w, \alpha, \tau) \) the variational parameters \( \{ \bar{w}, \bar{S}, \hat{\varepsilon}, \hat{\alpha}_l, \hat{b}_l, \ldots, \hat{a}_L, \hat{b}_L \} \) can be found in closed form as follows [8]:

\[
\bar{S} = \left( \tilde{\tau} \Phi^T \Phi + \text{diag}(\hat{\alpha}) \right)^{-1}, \quad \bar{w} = \tilde{\tau} \bar{S} (\Phi^T t)
\]

\[
\hat{\alpha}_l = a_l + 1/2, \quad \hat{b}_l = b_l + 1/2, \quad \hat{\varepsilon} = c + \frac{N}{2}, \quad \hat{d} = d + \frac{\| t - \Phi \bar{w} \|^2 + \text{Trace}(\bar{S}\Phi^T \Phi)}{2}.
\]

2.1. Adaptive fast variational SBL

Although expressions (2)-(4) reduce to those obtained in [2] when the approximating factors \( q(\tau) \) and \( q(\alpha_l) \) are chosen as Dirac measures on the corresponding domains, they do not reveal the structure of the marginal likelihood function that leads to an efficient SBL algorithm in [7]. Nonetheless, an analysis similar to [7] can be performed by computing the fixed points of the update expression for the variational parameters of a single factor \( q(\alpha_l) \) [10]. In [10] we have shown that, for a given basis function \( \phi_l \in \Phi \), the sequence of estimates \( \{ \hat{\alpha}_l(m) \}_{m=1}^{\infty} \), obtained by successively updating pdfs \( q(w) \) and \( q(\alpha_l) \), converges to the following fixed point \( \hat{\alpha}_l(\infty) \) as \( M \rightarrow \infty^{2}:

\[
\hat{\alpha}_l(\infty) = \left\{ \begin{array}{ll}
(\omega_l^2 - \xi_l)^{-1} & \omega_l^2 > \xi_l \\
\omega_l^2 & \omega_l^2 \leq \xi_l
\end{array} \right.,
\]

where \( \xi_l \) and \( \omega_l^2 \) are the pruning parameters defined as

\[
\xi_l = (\hat{\tau} \phi_l^T \phi_l - \hat{\tau}^2 \phi_l^T \Phi_l \bar{S}_l \Phi_l^T \phi_l)^{-1},
\]

\[
\omega_l^2 = (\hat{\tau} \phi_l^T \phi_l - \hat{\tau}^2 \phi_l^T \Phi_l \bar{S}_l \Phi_l^T \phi_l)^{-1}.
\]

The parameters \( \xi_l \) and \( \omega_l^2 \) depend on \( \Phi_l = \Phi \setminus \phi_l, \alpha_l = [\hat{\alpha}]_l, \text{ and } \bar{S}_l = (\hat{\tau} \phi_l^T \phi_l + \text{diag}(\hat{\alpha}_l))^{-1} \left[ \bar{S} - \frac{\bar{S} \phi_l^T \phi_l \bar{S}}{\phi_l^T \phi_l} \right] \frac{1}{\hat{\tau}} \),

where \( \Phi_l \) is the posterior covariance matrix of the weights obtained when the basis function \( \phi_l \) is removed from \( \Phi \). The result (5) provides a simple criterion for pruning a basis function from the active dictionary: a finite value of \( \hat{\alpha}_l(\infty) \) instructs us to keep the \( l \) th component in the model since it should minimize the free energy, while the infinite value of \( \hat{\alpha}_l(\infty) \) indicates that the basis function \( \phi_l \) is superfluous. It can be shown [10] that the test parameters \( \omega_l^2 \) and \( \xi_l \) are the squared weight of the basis function \( \phi_l \) and the weight’s variance computed when \( \hat{\alpha}_l \) equals zero—a fact that will become more evident later when we consider an inclusion of a basis function in the active dictionary. The pruning test is applied sequentially to all the basis functions in the active dictionary \( \Phi \) to determine which components should be pruned. Algorithm 1 summarizes the key steps of this procedure. For further details we refer the reader to [10].

**Algorithm 1** A test for a deletion of a basis function \( \phi_l \in \Phi \)

<table>
<thead>
<tr>
<th>Function</th>
<th>TestComponentPrune(l)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute</td>
<td>( \xi_l ) and ( \omega_l^2 ) from (7)</td>
</tr>
<tr>
<td>if ( \omega_l^2 &gt; \xi_l ) then</td>
<td></td>
</tr>
<tr>
<td>( \Delta \hat{\alpha}_l \leftarrow (\omega_l^2 - \xi_l)^{-1} - \hat{\alpha}_l )</td>
<td></td>
</tr>
<tr>
<td>( \bar{S} \leftarrow \bar{S} - \frac{\bar{S} \phi_l^T \phi_l \bar{S}}{\phi_l^T \phi_l} \hat{\alpha}_l \leftarrow (\omega_l^2 - \xi_l)^{-1} )</td>
<td></td>
</tr>
<tr>
<td>else</td>
<td></td>
</tr>
<tr>
<td>( \bar{S} \leftarrow \bar{S}_l, \hat{\alpha}_l \leftarrow [\hat{\alpha}]_l, \Phi \leftarrow \Phi \setminus \phi_l )</td>
<td></td>
</tr>
<tr>
<td>end if</td>
<td></td>
</tr>
</tbody>
</table>

It is natural to ask whether this scheme can be made fully adaptive by also allowing inclusion of new basis functions from the passive dictionary \( \Phi^C \). Let us assume that at some iteration of the al-

---

1. In [2] the posterior pdf of the weights \( w \) is Gaussian; its parameters coincide with the variational parameters of \( q(w) \) in (2).
2. Notice that since \( a_l = b_l = 0 \), the parameters of \( q(\alpha_l) \) in (3) can be specified as \( \hat{\alpha}_l = a_l + 1/2 = 1/2(b_l) \), where \( a_l = 1/(\omega_l^2 + \hat{\alpha}_l) \). Thus, it makes sense to study the fixed point of the variational update expressions in terms of \( \hat{\alpha}_l \) rather than in terms of \( \hat{\alpha}_l \) and \( b_l \).
gorithm we have $L$ basis functions in the active dictionary $\Phi$ and an estimate of $\hat{S}$ and $\hat{\alpha}$. Our goal is to test whether a basis function $\phi_{L+1} \in \Phi^C$ should be included in $\Phi$.

Assume for the moment that $\phi_{L+1}$ is in the active dictionary and that $\Phi_{L+1} = [\Phi, \phi_{L+1}]$, $\alpha_{L+1} = [\alpha^T, \alpha_{L+1}]^T$ and $\hat{S}_{L+1} = (\hat{\Phi}_{L+1}^T \Phi_{L+1} + \text{diag}(\alpha_{L+1}))^{-1}$ are available. Then we can compute $\hat{\alpha}_{L+1}^{[\infty]}$ from (5) and determine whether the basis function $\phi_{L+1}$ should be kept in the model. This can be done efficiently using only the current active dictionary $\Phi$ and the corresponding covariance matrix $\hat{S}$. First, consider a matrix $\hat{S}_{L+1}$ obtained from $\hat{S}_{L+1}$ by setting $\hat{\alpha}_{L+1} = 0$:

$$
\hat{S}_{L+1} = \begin{pmatrix}
\hat{\Phi}^T \hat{\Phi} + \text{diag}(\hat{\alpha}) & \hat{\Phi}^T \phi_{L+1} \\
\hat{\phi}_{L+1}^T \hat{\Phi} & \hat{\phi}_{L+1}^T \phi_{L+1} + \text{diag}(\alpha_{L+1})
\end{pmatrix}^{-1}
= \begin{pmatrix}
X_{L+1}^{-1} & -\hat{S} \hat{\Phi}^T \phi_{L+1} y_{L+1}^{-1} \\
-\hat{S} \hat{\Phi}^T \phi_{L+1} y_{L+1}^{-1} & y_{L+1}^{-1}
\end{pmatrix},
$$

(9)

where $X_{L+1} = \hat{S}^{-1} - \hat{\Phi}^T \phi_{L+1} (\phi_{L+1}^T \phi_{L+1})^{-1} \phi_{L+1}^T \hat{\Phi}$ and

$$
y_{L+1} = \hat{\Phi}^T \phi_{L+1} - \hat{\phi}_{L+1}^T \phi_{L+1} \hat{S} \hat{\Phi}^T \phi_{L+1}. 
$$

(10)

By comparing (6) and (10) we immediately notice that $\varsigma_{L+1} = y_{L+1}^{-1}$, which is the variance of the weight for $\phi_{L+1}$ when $\hat{\alpha}_{L+1} = 0$. Thus, $\omega_{L+1}^2$ can be computed from $\hat{\Phi}_{L+1}$ and $\hat{S}_{L+1}$ as

$$
\omega_{L+1}^2 = e_{L+1}^T (\hat{S}_{L+1} \Phi_{L+1} \tau_{L+1} t) (\hat{S}_{L+1} \Phi_{L+1} \tau_{L+1} t)^T e_{L+1}. 
$$

(11)

By substituting (9) into (11) and simplifying the resulting expression we finally obtain

$$
\omega_{L+1}^2 = (\hat{\varsigma}_{L+1} \phi_{L+1}^T \tau_{L+1} t - \hat{\varsigma}_{L+1} \phi_{L+1}^T \hat{S} \hat{\Phi}^T \phi_{L+1} \tau_{L+1} t)^2, 
$$

(12)

which is identical to (7) with an exception that (12) uses a new basis function $\phi_{L+1}$, a current design matrix $\Phi$ and a weight covariance matrix $\hat{S}$. Once the test parameters $\omega_{L+1}^2$ and $\varsigma_{L+1}$ are computed, we can test if the basis function $\phi_{L+1}$ should be included in the active dictionary $\Phi$. It should be mentioned that in [7] the authors compute parameters $q_{L+1}$ and $s_{L+1}$ that maximize the marginal likelihood function (see Eq. (19) in [7]). In fact, it is easy to notice that the expressions for $\omega_{L+1}^2$ in (12) and $\varsigma_{L+1} = y_{L+1}^{-1}$ in (10) co-incide respectively with the $q_{L+1}^2 / s_{L+1}^2$ and $s_{L+1}$. Consequently, the corresponding pruning test and the value of the sparsity parameter coincide. In the case of pruning an existing basis function the relationship is not that straightforward; nonetheless, the simulation results indicate that the both adaptive schemes achieve almost identical performance in terms of the mean squared error and the sparsity of the estimated models.

In case when the new basis function is accepted, the weight covariance matrix has to be updated as well. Luckily this can be done quite simply as follows:

$$
\hat{S}_{L+1} = \begin{pmatrix}
X_{L+1}^{-1} & -\hat{S} \hat{\Phi}^T \phi_{L+1} y_{L+1}^{-1} \\
-\hat{S} \hat{\Phi}^T \phi_{L+1} y_{L+1}^{-1} & y_{L+1}^{-1}
\end{pmatrix}, 
$$

(13)

where $X_{L+1} = \hat{S}^{-1} - \hat{\Phi}^T \phi_{L+1} \hat{S} \hat{\Phi}^T \phi_{L+1} \phi_{L+1}^T \phi_{L+1}$. The inverse of a Schur complement $X_{L+1}$ can be computed efficiently using a rank-one update [11].

In Algorithm 2 we summarize the main steps of this procedure.

Observe that the conditions for adding or deleting a basis function depend exclusively on the measurement $t$ and the matrix $\hat{S}$ that essentially determines how well a basis function “aligns” or correlates with the other basis functions in the active dictionary.

Notice that the ratio $\omega_1^2 / q_1$ can be interpreted as an estimate of the component signal-to-noise ratio3 $\text{SNR}_1 = \omega_1^2 / q_1$ [10]. Thus, SBL prunes a component if its estimated SNR is below $10$ dB. This interpretation allows a simple adjustment of the pruning condition as follows:

$$
\omega_1^2 > q_1 \times \text{SNR}', 
$$

(14)

where $\text{SNR}'$ is the adjustment SNR. This modified pruning condition (14) can be used both when adding as well as when pruning components. Such adjustment might be of a practical interest in scenarios for which the true SNR is known and the goal is to delete spurious components introduced by SBL due to the “imperfection” of the Gaussian sparsity prior or when we wish a sparse estimator that guarantees a certain quality of the estimated components in terms of their SNRs.

2.2. Implementation aspects

Variational inference typically requires choosing initial values for the variational parameters of $q(\mathbf{w}, \mathbf{\alpha}, \tau)$. Obviously, the adaptive ability of the algorithm can be exploited to recover the initial factorization by assuming $\Phi = \emptyset$, and $\Phi^C = \mathcal{D}$ and selecting the initial value of the noise precision $\tau$. The algorithm sequentially adds components using Alg. 2 and prunes irrelevant ones using Alg. 1. The pdfs $q(\mathbf{w})$ and $q(\tau)$ can be re-computed from (2) and (4) at any stage of the algorithm. It is important to mention that the order in which basis functions are added from the passive dictionary influences the final sparsity of the algorithm, which, as has been pointed out in [7], is related to the greediness of the fast SBL method. In our implementation of the fast adaptive V-SBL algorithm we rank all the components in $\Phi^C$ by pre-computing their sparsity parameters $\alpha_i$ and begin the inclusion with those basis functions that have the smallest value of $\alpha_i$, i.e., those functions that are best aligned with the measurement $t$. Also notice that updating $q(\tau)$ requires re-computing $\hat{S}$, which requires $O(L^3)$ operations.

3. Simulation results

In this section we compare the performance results of the fast adaptive V-SBL with the standard RVM algorithm [2], the fast marginal likelihood maximization method [7], and fast adaptive variational SBL with an SNR-adjusted threshold, via simulations. The standard RVM scheme is non-adaptive; we thus assume that all the available

3To gain some intuition into why this is so consider (9) and (11) when $L = 0$ and $\Phi = \emptyset$. 

2182
basis functions are included in the active dictionary. Note that the standard RVM algorithm requires one to specify a threshold for the hyperparameters $\alpha_k$, $\forall k$, to “detect” the divergence of the hyperparameter values. Obviously, this affects the performance of the standard RVM algorithm. In order to simplify the analysis of the simulation results we assumed the variance $\hat{\tau}$ of the noise to be known in all simulations. For all compared algorithms the same convergence criterion is used: an algorithm stops when (i) the number of basis functions between two consecutive iterations has stabilized and when (ii) the $\ell^2$-norm of the difference between the values of hyperparameters at two consecutive iterations is less than $10^{-4}$.

To test the algorithms we use basis functions $\phi_k \in \mathbb{R}^N$, $k = 1, \ldots, K$, generated by drawing samples from a multivariate Gaussian distribution with zero mean and covariance matrix $I$, and a sparse vector $w$ with $L = 10$ nonzero elements equal to 1 at random locations. With this setting we aim to test how the algorithm’s pruning mechanism performs when the exact sparsity of the model is known. We set $N = 100$ and $K = 200$. The target vector $t$ is generated according to (1). The performance of the tested algorithms, averaged over 100 independent runs, is summarized in Fig. 1. For adaptive algorithms each iteration corresponds to two steps: 1) adding components from the passive dictionary and 2) removing components from the active dictionary. As we see in Fig. 1(a) and 1(b) the variational SBL with adjusted pruning condition (14) outperforms the other estimation methods in terms of normalized mean-square error (NMSE) as well as in terms of the number of estimated components. In fact, it is able to recover the true model sparsity for $SNR > 10 \text{dB}$. The standard RVM recovers the true model sparsity only for $SNR > 40 \text{dB}$ with a pruning threshold $10^{-1}$; increasing this threshold leads to an over-estimation of the true sparsity. In Fig. 1(c) we plot the convergence rate of the algorithms for the $SNR = 30 \text{dB}$. Here the variational SBL with SNR-adjusted pruning is also a clear winner, reaching the stopping criterion in less than 10 iterations. Note, however, that both the fast variational SBL algorithm without the SNR-adjusted pruning and the fast marginal likelihood maximization algorithm exhibit very fast convergence; nonetheless they tend to overestimate the true model sparsity, as seen from Fig. 1(b).

4. CONCLUSION

In this work a fast adaptive variational Sparse Bayesian Learning (V-SBL) framework has been considered. The fast V-SBL algorithm optimizes a variational free energy with respect to variational parameters of the pdf of a single component. The fixed points of sparsity parameter update expressions as well as conditions that guarantee convergence to these fixed points – pruning conditions – have been obtained in a closed form. This significantly improves the convergence rate of V-SBL. The pruning conditions also reveal the relationship between the performance of SBL in terms of the number of estimated components and a measure of SNR. This relationship enables an empirical adjustment that allows inclusion of the components that guarantee a predefined quality in terms of their individual SNRs. Setting the adjustment parameter to the true SNR allows extraction of the true sparsity in simulated scenarios. Simulation studies demonstrate that this adjustment further accelerates the convergence rate of the algorithm.

5. REFERENCES