Chapter 4

Sparse Bayesian Learning

Sparse Bayesian Learning [6] is a general Bayesian framework within supervised Machine Learning. It can be applied to both regression and classification tasks. The *Relevance Vector Machine*, or *RVM*, is a particular specialisation of the Sparse Bayesian Learning model which has identical functional form to the Support Vector Machine (SVM). However, the RVM comes with a number of key advantages over the SVM. The solution produced by a RVM is typically much sparser than the solution by a comparable SVM. Furthermore, the RVM is a probabilistic model and as such, allows us to estimate error bounds in its predictions.

In this chapter, we will derive the Sparse Bayesian Learning model for regression. We will summarise both the original inference algorithm [6] and also the faster “Sequential Sparse Bayesian Learning Algorithm” [7].

4.1 Model Specification

We are given a data set of N input vectors $\{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$ and their associated *targets* $\{y^{(i)}\}_{i=1}^{N}$. The input vectors live in D-dimensional space, $\boldsymbol{x} \in \mathbb{R}^D$. The targets are real values, $y \in \mathbb{R}$. [1]

We model the data using a linearly-weighted sum of M fixed basis functions $\{\phi_j(\cdot)\}_{j=1}^{M}$ and base our predictions on the function $f(\cdot)$ defined as

$$f(\boldsymbol{x}; \boldsymbol{w}) = \sum_{j=1}^{M} w_j \phi_j(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) \tag{4.1}$$

[1]When using the Sparse Bayesian model for regression, we assume the targets are real-valued. It is also possible to use the model for classification in which cased the targets are assumed to be discrete class labels.

where $\boldsymbol{w} = [w_1, \ldots, w_M]^T$ and $\boldsymbol{\phi}(\cdot) = [\phi_1(\cdot), \ldots, \phi_M(\cdot)]^T$. Using a large number M of non-linear basis functions $\phi_j : \mathbb{R}^D \to \mathbb{R}$ allows for a highly flexible model.

The *Relevance Vector Machine*, or RVM, is a specialisation of the Sparse Bayesian Learning model in which the basis functions take the form of *kernel functions*

$$\phi_j(\cdot) \equiv K\left(\cdot\,, \boldsymbol{x}^{(j)}\right).$$

This defines a basis function for each training data point $\boldsymbol{x}^{(i)}$. Typically, we also include an additional *bias* term $\phi_0(\cdot) \equiv 1$, so that $M = N + 1$. The RVM has identical functional form to the popular Support Vector Machine (SVM), but superior properties. It typically gives sparser solutions than the SVM and has the additional advantage of providing confidence measures for its predictions.

However, in the following derivation, we will stick to the case of general basis functions $\phi_j : \mathbb{R}^D \to \mathbb{R}$. Thus M need not equal $N+1$ and may, in fact, be a lot larger.

To train the model (4.1), i.e. find values for $\boldsymbol{w}$ that are optimal in some sense, we make the standard assumption that our training data are samples from the model with additive noise:

$$\begin{aligned} y^{(i)} &= f(\boldsymbol{x}^{(i)}; \boldsymbol{w}) + \epsilon^{(i)} \\ &= \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}^{(i)}) + \epsilon^{(i)} \qquad i = 1, \ldots, N. \end{aligned} \tag{4.2}$$

The errors $\{\epsilon^{(i)}\}_{i=1}^N$ are assumed to be independent samples from a zero-mean Gaussian distribution with variance σ^2

$$p(\epsilon^{(i)}) = \mathcal{N}(\epsilon^{(i)} \,|\, 0, \sigma^2) \qquad i = 1, \ldots, N. \tag{4.3}$$

Combining equation (4.2) with equation (4.1), we may express the model for the complete data using matrix notation:

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{w} + \boldsymbol{\epsilon} \tag{4.4}$$

where $\boldsymbol{\epsilon} = \left[\epsilon^{(1)}, \cdots, \epsilon^{(N)}\right]^T$. The $N \times M$ matrix $\boldsymbol{\Phi}$ is known as the design matrix. The ith row of $\boldsymbol{\Phi}$ is given by $\boldsymbol{\phi}(\boldsymbol{x}^{(i)})^T$. The jth column of $\boldsymbol{\Phi}$ is given by $\boldsymbol{\phi}_j = \left[\phi_j\left(\boldsymbol{x}^{(1)}\right), \cdots, \phi_j\left(\boldsymbol{x}^{(N)}\right)\right]^T$, which is also referred to as the jth *basis vector*. Thus

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1 & \cdots & \boldsymbol{\phi}_M \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{x}^{(1)})^T \\ \vdots \\ \boldsymbol{\phi}(\boldsymbol{x}^{(N)})^T \end{bmatrix}$$

Combining equation (4.4) and equation (4.3), we find that the complete data likelihood function is given by

$$\begin{aligned} p\left(\boldsymbol{y} \mid \boldsymbol{w}, \sigma^2\right) &= \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{w}, \sigma^2 \boldsymbol{I}_M\right) \\ &= (2\pi\sigma^2)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}||\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w}||^2\right\} \end{aligned} \tag{4.5}$$

where $\boldsymbol{I}_M$ is the $M \times M$ identity matrix.

So far, we have specified the general linear regression model. To get to the sparse Bayesian formulation, we define a zero-mean Gaussian prior distribution over the parameters $\boldsymbol{w}$

$$p(\boldsymbol{w} \mid \boldsymbol{\alpha}) = \prod_{j=1}^{M} \mathcal{N}\left(w_j \mid \alpha_j^{-1}\right)$$

where $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_M]^T$ is a vector of M hyperparameters. It is important to note that each hyperparameter α_j is solely responsible for controlling the strength of the prior of its associated weight w_j. If α_j is large, the prior over w_j is very strongly peaked at zero. This form of the prior distribution is, more than anything, responsible for the dramatic sparsity in the final model.

To complete the specification, we must define a prior over the noise parameter σ^2 and the a hyperprior over the hyperparameters $\boldsymbol{\alpha}$. Following the derivation in [6], we use the following Gamma [2] priors

$$p(\boldsymbol{\alpha} \mid a, b) = \prod_{j=1}^{M} \text{Gamma}(\alpha_j \mid a, b)$$

$$p(\beta \mid c, d) = \text{Gamma}(\beta \mid c, d)$$

where $\beta \equiv \sigma^{-2}$.

As a side note, consider the prior of $\boldsymbol{w}$ after marginalising out the dependence on the hyperpriors $\boldsymbol{\alpha}$. Since each w_j is normally distributed with an unknown precision parameter α_j and since the (hyper)prior over α_j is the Gamma distribution and

[2] The Gamma distribution is defined by

$$\text{Gamma}(z \mid a, b) = \Gamma(a)^{-1} b^a z^{a-1} \exp(-bz) \qquad z, a, b > 0$$

where $\Gamma(.)$ is the Gamma function defined by

$$\Gamma(z) = \int_0^\infty t^{z-1} \exp(-t) dt.$$

therefore conjugate to $p(w_j \,|\, \alpha_j)$, it follows that the resulting integral can be evaluated analytically

$$\begin{aligned} p(\boldsymbol{w} \,|\, a, b) &= \int p(\boldsymbol{w} \,|\, \boldsymbol{\alpha}) p(\boldsymbol{\alpha} \,|\, a, b) d\boldsymbol{\alpha} \\ &= \prod_{j=1}^{M} \int \mathcal{N}(w_j \,|\, 0, \alpha_j^{-1}) \,\text{Gamma}(\alpha_j | \, a, b) d\alpha_j \\ &= \prod_{j=1}^{M} \frac{b^a \Gamma(a + \frac{1}{2})}{(2\pi)^{\frac{1}{2}} \Gamma(a)} \left(b + \frac{w_j^2}{2} \right)^{-(a+\frac{1}{2})}. \end{aligned}$$

This corresponds to a product of independent Student-t density functions over the weights w_j. The choice $a = b = 0$ implies that $p(\boldsymbol{w} \,|\, a, b) \propto \prod_{j=1}^{M} 1/|w_j|$. As discussed in [6], it is this hierarchical formulation of the weight prior that is ultimately responsible for encouraging sparse solutions.

4.2 Model Inference

We have specified the likelihood model for the data and a prior distribution over the model parameters. The next step in Bayesian inference is to compute the posterior distribution of the parameters. We begin by setting up Bayes' Rule

$$p(\boldsymbol{w}, \boldsymbol{\alpha}, \sigma^2 \,|\, \boldsymbol{y}) = \frac{p(\boldsymbol{y} \,|\, \boldsymbol{w}, \boldsymbol{\alpha}, \sigma^2) p(\boldsymbol{w}, \boldsymbol{\alpha}, \sigma^2)}{\int p(\boldsymbol{y} \,|\, \boldsymbol{w}, \boldsymbol{\alpha}, \sigma^2) p(\boldsymbol{w}, \boldsymbol{\alpha}, \sigma^2) d\boldsymbol{w} d\boldsymbol{\alpha} d\sigma^2} \tag{4.6}$$

The integral in the denominator of (4.6) is computationally intractable and we must resort to an alternative strategy. First, we decompose the left-hand-side of equation (4.6) as

$$p(\boldsymbol{w}, \boldsymbol{\alpha}, \sigma^2 \,|\, \boldsymbol{y}) = p(\boldsymbol{w} \,|\, \boldsymbol{y}, \boldsymbol{\alpha}, \sigma^2) p(\boldsymbol{\alpha}, \sigma^2 \,|\, \boldsymbol{y}).$$

Next, we use Bayes' Rule to compute the posterior distribution of the weights given $\boldsymbol{\alpha}$ and σ^2

$$p(\boldsymbol{w} \,|\, \boldsymbol{y}, \boldsymbol{\alpha}, \sigma^2) = \frac{p(\boldsymbol{y} \,|\, \boldsymbol{w}, \sigma^2) p(\boldsymbol{w} \,|\, \boldsymbol{\alpha})}{p(\boldsymbol{y} \,|\, \boldsymbol{\alpha}, \sigma^2)} \tag{4.7}$$

The denominator of the right-hand-side is known as the *marginal likelihood* and given by

$$p(\boldsymbol{y} \,|\, \boldsymbol{\alpha}, \sigma^2) = \int p(\boldsymbol{y} \,|\, \boldsymbol{w}, \sigma^2) p(\boldsymbol{w} \,|\, \boldsymbol{\alpha}) d\boldsymbol{w} \tag{4.8}$$

Since $\boldsymbol{\alpha}$ and σ^2 are treated as fixed quantities in equation (4.7), the Gaussian density $p(\boldsymbol{w} \,|\, \boldsymbol{\alpha})$ is the conjugate prior to the Gaussian likelihood function $p(\boldsymbol{y} \,|\, \boldsymbol{w}, \sigma^2)$. Thus, the integral in equation (4.8) is a convolution of two Gaussians and therefore equal to another Gaussian:

$$\begin{aligned} p(\boldsymbol{y} \mid \boldsymbol{\alpha}, \sigma^2) &= \mathcal{N}(\boldsymbol{y} \mid \mathbf{0}, \boldsymbol{C}) \\ &= (2\pi)^{-N/2} |\boldsymbol{C}|^{-1/2} \exp\left\{ -\frac{1}{2} \boldsymbol{y}^T \boldsymbol{C}^{-1} \boldsymbol{y} \right\} \end{aligned}$$

where

$$\boldsymbol{C} = \sigma^2 \boldsymbol{I}_N + \boldsymbol{\Phi} \boldsymbol{A}^{-1} \boldsymbol{\Phi}^T. \tag{4.9}$$

The posterior distribution for $\boldsymbol{w}$ is a also a Gaussian, $p(\boldsymbol{w} \mid \boldsymbol{y}, \boldsymbol{\alpha}, \sigma^2) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Its mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ are given by

$$\boldsymbol{\Sigma} = \left(\sigma^{-2} \boldsymbol{\Phi}^T \boldsymbol{\Phi} + \boldsymbol{A} \right)^{-1} \tag{4.10}$$

$$\boldsymbol{\mu} = \sigma^{-2} \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \boldsymbol{y} \tag{4.11}$$

with $\boldsymbol{A} = \operatorname{diag}(\boldsymbol{\alpha})$.

Finally, we need to find the posterior of the hyperparameters, $p(\boldsymbol{\alpha}, \sigma^2 \mid \boldsymbol{y})$. This part is computationally intractable, so instead we approximate the posterior by a delta-function at its mode. Hence, the problem reduces to finding the values of $\boldsymbol{\alpha}$ and σ^2 that maximise $p(\boldsymbol{\alpha}, \sigma^2 \mid \boldsymbol{y}) \propto p(\boldsymbol{y} \mid \boldsymbol{\alpha}, \sigma^2) p(\boldsymbol{\alpha}) p(\sigma^2)$.

Here, we make the simplifying assumption that $a = b = c = d = 0$, giving us uniform (but improper) hyperpriors (see [6] for the general case). Maximising $p(\boldsymbol{\alpha}, \sigma^2 \mid \boldsymbol{y})$ is then equivalent to maximising the marginal likelihood, or equivalently, its logarithm

$$\begin{aligned} \mathcal{L}(\boldsymbol{\alpha}, \sigma^2) &= \log p(\boldsymbol{y} \mid \boldsymbol{\alpha}, \sigma^2) = \log \mathcal{N}(\boldsymbol{y} \mid \mathbf{0}, \boldsymbol{C}) \\ &= -\frac{1}{2} \left[N \log 2\pi + \log |\boldsymbol{C}| + \boldsymbol{y}^T \boldsymbol{C}^{-1} \boldsymbol{y} \right] \end{aligned} \tag{4.12}$$

The procedure of finding $\boldsymbol{\alpha}$ and σ^2 that maximise the (log) marginal likelihood (4.12) is also known as *type-II Maximum likelihood* and *evidence approximation.*

4.2.1 Original Training Algorithm

The original training algorithm in [6] is derived by setting the derivatives of (4.12) to zero. We obtain the following update equations for $\boldsymbol{\alpha}$ and σ^2:

$$\alpha_j^{\text{new}} = \frac{\gamma_j}{\mu_j^2} \tag{4.13}$$

Algorithm 4.1 Sparse Bayesian Learning: Original Training Algorithm

```
1: Choose some initial positive values for $\sigma^2$ and $\alpha_j$ for $j = 1, \cdots, M$
2: repeat
3:     $\boldsymbol{A} = \text{diag}(\boldsymbol{\alpha})$
4:     $\boldsymbol{\Sigma} = \left(\sigma^{-2}\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \boldsymbol{A}\right)^{-1}$
5:     $\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\boldsymbol{y}$

6:     for $j = 1, \cdots, M$ do
7:         $\gamma_j = 1 - \alpha_j\Sigma_{jj}$
8:         $\alpha_j = \gamma_j/\mu_j^2$
9:     end for
10:    $\sigma^2 = ||\boldsymbol{y} - \boldsymbol{\Phi\mu}||^2/(N - \sum_j \gamma_j)$
11: until Convergence
```

$$(\sigma^2)^{\text{new}} = \frac{||\boldsymbol{y} - \boldsymbol{\Phi\mu}||^2}{N - \sum_j \gamma_j} \tag{4.14}$$

where μ_j is the jth component of the posterior mean $\boldsymbol{\mu}$ (4.11). The quantities γ_j are defined by

$$\gamma_j = 1 - \alpha_j\Sigma_{jj}$$

where Σ_{jj} is the jth diagonal element of the posterior covariance $\boldsymbol{\Sigma}$ (4.10).

To train the model, we can start by giving $\boldsymbol{\alpha}$ and σ^2 some initial values and evaluate the mean and covariance of the weights posterior using equations (4.11) and (4.10), respectively. Next we alternate between re-estimating the hyperparamaters $\boldsymbol{\alpha}$ and σ^2 using (4.13) and (4.14) and updating the posterior mean and covariance parameters using (4.11) and (4.10). We continue until a relevant convergence criterion is met. For example, we may choose to stop if the change in the marginal likelihood - or, alternatively, the change in the parameter values - between two iterations is below a certain pre-defined threshold.

This procedure is summarised in Algorithm 4.1.

During training, it is typically observed that many of the hyperparameters α_j tend to infinity. Equations (4.11) and (4.10) imply that the weights w_j corresponding to these hyperparameters have a posterior distribution where the mean and the variance are both zero, meaning their posterior is infinitely peaked at zero. As a consequence, the corresponding basis functions $\phi_j(\cdot)$ are effectively removed from the model and we achieve sparsity.

In the case of the RVM, where $\phi_j(\cdot) \equiv K(\cdot, \boldsymbol{x}^{(j)})$, the input vectors $\boldsymbol{x}^{(j)}$ corresponding to the remaining non-zero weights are known as the *relevance vectors* of the model.

4.2.2 Sequential Sparse Bayesian Learning Algorithm

A central drawback of the training algorithm discussed in the previous section is its speed. The computational complexity scales with the cube of the number of basis functions. During training, as basis functions are pruned from the model, the algorithm accelerates. Nevertheless, if M is very large, the procedure can be very expensive to run.

An alternative strategy of maximising the marginal likelihood (4.12) was developed by [7], resulting in a highly accelerated training algorithm: the *Sequential Sparse Bayesian Learning Algorithm.* It starts with a single basis function and maximises the marginal likelihood by sequentially adding and deleting candidate basis functions.

To derive the algorithm, we follow the analysis in [5] and consider the dependence of the marginal likelihood $\mathcal{L}(\boldsymbol{\alpha}, \sigma^2)$ on a single hyperparameter α_j. First, we decompose the matrix $\boldsymbol{C}$, defined in (4.9), as follows:

$$\begin{aligned}\boldsymbol{C} &= \sigma^2 \boldsymbol{I}_N + \sum_{m \neq j} \alpha_m^{-1} \boldsymbol{\phi}_m \boldsymbol{\phi}_m^T + \alpha_j^{-1} \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \\ &= \boldsymbol{C}_{-j} + \alpha_j^{-1} \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T\end{aligned}$$

where $\boldsymbol{C}_{-j} \equiv \sigma^2 \boldsymbol{I}_N + \sum_{m \neq j} \alpha_m^{-1} \boldsymbol{\phi}_m \boldsymbol{\phi}_m^T$ is $\boldsymbol{C}$ without the contribution of the jth basis vector $\boldsymbol{\phi}_j$. Making use of standard identities [WHICH ONES?] for matrix inverses and determinants, we can express $|\boldsymbol{C}|$ and $\boldsymbol{C}^{-1}$ as

$$\boldsymbol{C}^{-1} = \boldsymbol{C}_{-j}^{-1} - \frac{\boldsymbol{C}_{-j}^{-1} \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \boldsymbol{C}_{-j}^{-1}}{\alpha_j + \boldsymbol{\phi}_j^T \boldsymbol{C}_{-j}^{-1} \boldsymbol{\phi}_j}$$

$$|\boldsymbol{C}| = |\boldsymbol{C}_{-j}| \left| 1 + \alpha_j^{-1} \boldsymbol{\phi}_j^T \boldsymbol{C}_{-j}^{-1} \boldsymbol{\phi}_j \right|$$

This allows us to decompose the marginal likelihood:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha}, \sigma^2) &= \mathcal{L}(\boldsymbol{\alpha}_{-j}, \sigma^2) + \frac{1}{2}\left[\log \alpha_j - \log(\alpha_j + s_j) + \frac{q_j^2}{\alpha_j + s_j}\right] \\ &\equiv \mathcal{L}(\boldsymbol{\alpha}_{-j}, \sigma^2) + \ell(\alpha_j, \sigma^2)\end{aligned} \tag{4.15}$$

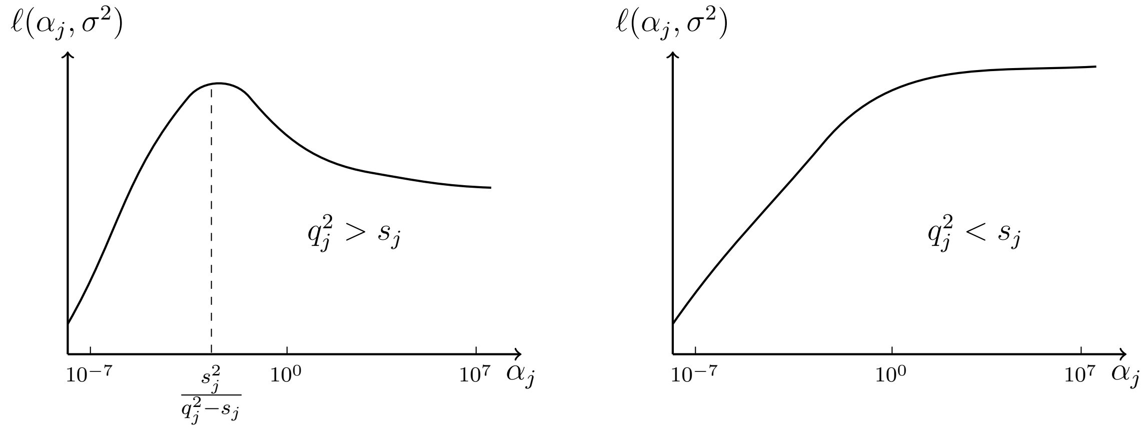

Fig. 4.1 Example plots of $\ell(\alpha_j, \sigma^2)$ against α_j illustrating the stationary points when $q_j^2 > s_j$ (left) and $q_j^2 < s_j$ (based on [5]).

This conveniently seperates terms in α_j in $\ell(\alpha_j, \sigma^2)$ from the remaining terms in $\mathcal{L}(\boldsymbol{\alpha}_{-j}, \sigma^2)$, which is the (log) marginal likelihood with the basis vector $\boldsymbol{\phi}_j$ excluded.

The quantity s_j is the *sparsity factor*, defined as

$$s_j = \boldsymbol{\phi}_j^T \boldsymbol{C}_{-j}^{-1} \boldsymbol{\phi}_j.$$

It serves as a measure of how much the marginal likelihood would decrease if we added $\boldsymbol{\phi}_j$ to the model. The quantity q_j, on the other hand, is known as the *quality factor*. It is defined as

$$q_j = \boldsymbol{\phi}_j^T \boldsymbol{C}_{-j}^{-1} \boldsymbol{y}$$

and measures the extent to which ϕ_j increases $\mathcal{L}(\boldsymbol{\alpha}, \sigma^2)$ by helping to explain the data $\boldsymbol{y}$. Thus, a particular basis vector $\boldsymbol{\phi}_j$ should not be included in the model if its sparsity factor s_j is large, unless it is offset by a large quality factor q_j.

We can see this more explicitly if we consider the first derivative of $\ell(\alpha_j, \sigma^2)$ with respect to α_j [5]

$$\frac{\partial \ell(\alpha_j, \sigma^2)}{\partial \alpha_j} = \frac{\alpha_j^{-1} s_j^2 - (q_j^2 - s_j)}{2(\alpha_j + s_j)^2}$$

Equating it to zero (and noting that α_j is an inverse-variance and therefore positive), we obtain the following solution for α_j:

$$\alpha_j = \begin{cases} s_j^2/(q_j^2 - s_j) & \text{if } q_j^2 > s_j \\ +\infty & \text{otherwise} \end{cases} . \tag{4.16}$$

The solution (4.16) is illustrated in Figure 4.1.

Algorithm 4.2 Sequential Sparse Bayesian Learning Algorithm [7]

```
1: Initialise $\sigma^2$.
2: Add basis function $\boldsymbol{\phi}_j$ to the model, where $j = \arg\max_m \left\{||\boldsymbol{\phi}_m^T \boldsymbol{y}||^2/||\boldsymbol{\phi}_m||^2\right\}$.
   Set $\alpha_j = \frac{||\boldsymbol{\phi}_j||^2}{||\boldsymbol{\phi}_j^T \boldsymbol{y}||^2/||\boldsymbol{\phi}_j||^2 - \sigma^2}$. Set $\alpha_m = \infty$ for $m \neq j$.
3: Compute $\boldsymbol{\Sigma} = \left(\sigma^{-2}\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \boldsymbol{A}\right)^{-1}$ and $\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\boldsymbol{y}$ which are scalars initially.
   Compute $S_m$, $Q_m$, $s_m$ and $q_m$ for $m = 1, \cdots, M$ using (4.17) − (4.20).
4: **repeat**
5:     Select some candidate basis vector $\boldsymbol{\phi}_j$.
6:     **if** $q_j^2 > s_j$ and $\alpha_j = \infty$ **then add** $\boldsymbol{\phi}_j$ to the model and update $\alpha_j$.
7:     **if** $q_j^2 > s_j$ and $\alpha_j < \infty$ **then re-estimate** $\alpha_j$.
8:     **if** $q_j^2 < s_j$ and $\alpha_j < \infty$ **then delete** $\boldsymbol{\phi}_j$ from the model and set $\alpha_j = \infty$.
9:     Update $\boldsymbol{\Sigma}$, $\boldsymbol{\mu}$ and $S_m$, $Q_m$, $s_m$, $q_m$ for $m = 1, \cdots, M$.
10: **until** Convergence
```

It follows that, if, during training, the basis vector $\boldsymbol{\phi}_j$ is currently in the model (meaning $\alpha_j < \infty$) even though $q_j^2 \leq s_j$, then α_j should be set to ∞ and $\boldsymbol{\phi}_j$ should be pruned from the model. On the other hand, if $\boldsymbol{\phi}_j$ is currently excluded from the model (i.e. $\alpha_j = \infty$), but $q_j^2 > s_j$, then α_j should be set to $s_j^2/(q_j^2 - s_j)$ and $\boldsymbol{\phi}_j$ should be added to the model.

During the algorithm, we must maintain and update values of the quality factors and sparsity factors for all basis functions, as well as the posterior mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ of the weights $\boldsymbol{w}$. In practice, it easier to keep track of the quantities $Q_m = \boldsymbol{\phi}_m^T \boldsymbol{C}^{-1}\boldsymbol{\phi}_m$ and $S_m = \boldsymbol{\phi}_m^T \boldsymbol{C}^{-1}\boldsymbol{y}$ which can also be written as

$$S_m = \sigma^{-2}\boldsymbol{\phi}_m^T\boldsymbol{\phi}_m - \sigma^{-4}\boldsymbol{\phi}_m^T\boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\boldsymbol{\phi}_m \tag{4.17}$$

$$Q_m = \sigma^{-2}\boldsymbol{\phi}_m^T\boldsymbol{y} - \sigma^{-4}\boldsymbol{\phi}_m^T\boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\boldsymbol{y} \tag{4.18}$$

where $\boldsymbol{\Sigma}$ and $\boldsymbol{\Phi}$ contain only the basis functions that are currently included in the model.

The factors s_m and q_m can by obtained from S_m and Q_m as follows:

$$s_m = \frac{\alpha_m S_m}{\alpha_m - S_m} \tag{4.19}$$

$$q_m = \frac{\alpha_m Q_m}{\alpha_m - S_m} \tag{4.20}$$

Note that if $\alpha_m = \infty$, then $q_m = Q_m$ and $s_m = S_m$.

4.3 Making Predictions

sparse mu predictive distribution

For details on the RVM and its implementation see [3, 6, 7].

References

[1] Candès, E. J. and Wakin, M. B. (2008). An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30.

[2] DeVore, R. A., Jawerth, B., and Lucier, B. J. (1992). Image compression through wavelet transform coding. *Information Theory, IEEE Transactions on*, 38(2):719–746.

[3] Pilikos, G. (2014). Signal reconstruction using compressive sensing. MPhil thesis, University of Cambridge.

[4] Stollnitz, E. J., DeRose, T. D., and Salesin, D. H. (1995). Wavelets for computer graphics: a primer. 1. *Computer Graphics and Applications, IEEE*, 15(3):76–84.

[5] Tipping, A. and Faul, A. (2002). Analysis of sparse bayesian learning. *Advances in neural information processing systems*, 14:383–389.

[6] Tipping, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *The journal of machine learning research*, 1:211–244.

[7] Tipping, M. E., Faul, A. C., et al. (2003). Fast marginal likelihood maximisation for sparse bayesian models. In *AISTATS*.