CHAPTER

# BAYESIAN LEARNING: APPROXIMATE INFERENCE AND NONPARAMETRIC MODELS 13

## CHAPTER OUTLINE

**Machine Learning. http://dx.doi.org/10.1016/B978-0-12-801522-3.00013-6**

## 13.1 INTRODUCTION

This chapter is the second one dedicated to Bayesian learning. The emphasis here, compared to Chapter 12, is on more advanced topics, dealing with approximate inference methods. Such methods are employed when the involved integrations are no longer computationally tractable. Two paths for approximate inference, known as variational techniques, are discussed. One is based on the mean field approximation and the lower bound interpretation of the EM, and the other on convex duality and variational bounds. Regression and mixture modeling are discussed in this framework. Emphasis is given to sparse Bayesian modeling techniques and hierarchical Bayesian models. The relevance vector machine framework is presented. Expectation propagation is also discussed as an alternative to variational methods for approximate inference. At the end of the chapter, Bayesian learning in the context of nonparametric models is discussed, including Gaussian processes. Finally, a case study concerning hyperspectral imaging is presented.

## 13.2 VARIATIONAL APPROXIMATION IN BAYESIAN LEARNING

Recall that in order to apply the EM algorithm, the functional form of the posterior of the latent variables, given the observations, must be known. However, the analytic computation of the posterior is not always tractable. In such cases, the EM algorithm, in its standard form as discussed in the previous chapter, is not applicable. In this section, we will describe an alternative path that builds upon the EM interpretation given in Section 12.5.2.

Once more, we will adopt a general notation, which can then be adapted to the needs of specific problems. Let $\mathcal{X}$ be the set of observed variables and $\mathcal{X}^l$ the respective set of latent ones, as they were defined in Section 12.5. Furthermore, in this section, we will explicitly bring into the game the set of parameters, $\theta \in \mathbb{R}^K$, which are treated as random variables in the Bayesian context, accompanied by a prior pdf. Note that although we could consider the parameters as latent variables, we do not. Here, we reserve the term "latent" for hidden variables whose number depends on the number of observations, $N$. In contrast, a random parameter vector, $\theta$, though a hidden random vector, has a fixed dimension. The reason we do that is to allow us to employ a prior pdf only for the parameters, in order to serve the needs of our examples. The functional in Eq. (12.63) is now redefined as

$$\mathcal{F}(q,\boldsymbol{\xi}) = \int q(\mathcal{X}^l,\boldsymbol{\theta}) \ln \frac{p(\mathcal{X},\mathcal{X}^l,\boldsymbol{\theta};\boldsymbol{\xi})}{q(\mathcal{X}^l,\boldsymbol{\theta})}\, \mathrm{d}\mathcal{X}^l\, \mathrm{d}\boldsymbol{\theta}, \tag{13.1}$$

where $\boldsymbol{\xi}$ is the set of deterministic (hyper)parameters. Let us, for the time being, get rid of $\boldsymbol{\xi}$ for the sake of notational relaxation. Then the counterpart of Eq. (12.64) becomes

$$\mathcal{F}(q) = \ln p(\mathcal{X}) + \int q(\mathcal{X}^l, \boldsymbol{\theta}) \ln \frac{p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X})}{q(\mathcal{X}^l, \boldsymbol{\theta})} \, \mathrm{d}\mathcal{X}^l \, \mathrm{d}\boldsymbol{\theta}. \tag{13.2}$$

The difference with Eq. (12.64) lies in the fact that $p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X})$ is not known, so maximizing Eq. (13.2) with respect to $q$ by setting to zero the KL divergence $\mathrm{KL}(q||p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X}))$ is no longer possible.

Optimizing a functional with respect to a function is known in mathematics as *calculus of variations*. The simplest example of this problem is to compute the geodesic that connects two points on a surface. Two names whose contributions are considered significant breakthroughs that consolidated this field are the Swiss German mathematician Leonhard Euler (1707-1783) and the Italian-born mathematician and astronomer Joseph-Louis Lagrange (1736-1813). It is interesting to note that Lagrange succeeded Euler as director of mathematics in the Prussian Academy of Sciences in Berlin.

In order to deal with the current problem, we will *constrain* $q(\mathcal{X}^l, \boldsymbol{\theta})$ to lie within a family of functions. Note that in this case, if the unknown $p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X})$ does not belong to this specific family of functions, the KL divergence *cannot* become zero, and the lower bound, $\mathcal{F}(q)$, of the marginal log-likelihood *cannot* be made tight. This is the reason the method is called a variational approximation.

### The mean field approximation

This type of approximation results by constraining $q(\mathcal{X}^l, \boldsymbol{\theta})$ to be factorized, that is,

$$q(\mathcal{X}^l, \boldsymbol{\theta}) = q_{\mathcal{X}^l}(\mathcal{X}^l) q_\theta(\boldsymbol{\theta}). \tag{13.3}$$

This factorization can be, and usually is, extended to

$$\boxed{q(\mathcal{X}^l, \boldsymbol{\theta}) = q_{\boldsymbol{x}_1^l}(\boldsymbol{x}_1^l) \ldots q_{\boldsymbol{x}_N^l}(\boldsymbol{x}_N^l) q_\theta(\boldsymbol{\theta}) : \quad \text{Mean Field Approximation.}} \tag{13.4}$$

Also, the hidden variables can be factorized in groups. Similarly, the parameter factor can be further factorized if the parameters can be grouped in different groups, as is often the case. To simplify our notation, without sacrificing generality, we will work with Eq. (13.3). This type of approximation has been inspired from the field of statistical physics and is known as *mean field approximation* (e.g., [14, 40, 53]).

Having adopted Eq. (13.3) and recalling that $p(\mathcal{X}, \mathcal{X}^l, \boldsymbol{\theta}) = p(\mathcal{X}, \mathcal{X}^l|\boldsymbol{\theta}) p(\boldsymbol{\theta})$, Eq. (13.1) becomes (Problem 13.1),

$$\begin{aligned} \mathcal{F}(q_{\mathcal{X}^l}, q_\theta) = & \int q_{\mathcal{X}^l}(\mathcal{X}^l) \left( \int q_\theta(\boldsymbol{\theta}) \ln p(\mathcal{X}, \mathcal{X}^l, \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} \right) \mathrm{d}\mathcal{X}^l \\ & - \int q_{\mathcal{X}^l}(\mathcal{X}^l) \ln q_{\mathcal{X}^l}(\mathcal{X}^l) \, \mathrm{d}\mathcal{X}^l - \int q_\theta(\boldsymbol{\theta}) \ln q_\theta(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}, \end{aligned} \tag{13.5}$$

or equivalently

$$\begin{aligned} \mathcal{F}(q_{\mathcal{X}^l}, q_{\boldsymbol{\theta}}) = & \int q_\theta(\theta) \left( \int q_{\mathcal{X}^l}(\mathcal{X}^l) \ln \left( p(\mathcal{X}, \mathcal{X}^l|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \right) \mathrm{d}\mathcal{X}^l \right) \mathrm{d}\boldsymbol{\theta} \\ & - \int q_\theta(\boldsymbol{\theta}) \ln q_\theta(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} - \int q_{\mathcal{X}^l}(\mathcal{X}^l) \ln q_{\mathcal{X}^l}(\mathcal{X}^l) \, \mathrm{d}\mathcal{X}^l. \end{aligned} \tag{13.6}$$

Having expressed the lower bound, $\mathcal{F}(q_{\mathcal{X}^l}, q_\theta)$, as in Eqs. (13.5) and (13.6), maximization with respect to $q(\mathcal{X}^l, \boldsymbol{\theta})$ (as required by the E-step of the EM algorithm) will take place by splitting the process in order to maximize first with respect to $q_{\mathcal{X}^l}$ and then with respect to $q_\theta$. Bringing back into the scene the (deterministic) parameter vector, $\boldsymbol{\xi}$, and initializing the algorithm from arbitrary values for $\boldsymbol{\xi}^{(0)}$ as well as for the involved statistics related to $\boldsymbol{q}_\theta$ (this will become clear while dealing with the examples), the $(j+1)$ iteration comprises the following steps:

E-Step 1a: Holding $\boldsymbol{\xi}^{(j)}$ and $q_\theta^{(j)}$ fixed, optimize Eq. (13.5) with respect to $q_{\mathcal{X}^l}$, that is,

$$\begin{aligned} q_{\mathcal{X}^l}^{(j+1)}(\mathcal{X}^l) &= \max\nolimits_{q_{\mathcal{X}^l}} \mathcal{F}(q_{\mathcal{X}^l}(\mathcal{X}^l), q_\theta^{(j)}(\boldsymbol{\theta})) \\ &= \max\nolimits_{q_{\mathcal{X}^l}} \int q_{\mathcal{X}^l}(\mathcal{X}^l) \ln \frac{\tilde{p}(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi}^{(j)})}{q_{\mathcal{X}^l}(\mathcal{X}^l)} \, \mathrm{d}\mathcal{X}^l + \text{constant}, \end{aligned} \tag{13.7}$$

where "constant" contains all the terms that do not depend on $\mathcal{X}^l$ and we have defined

$$\begin{aligned} \int q_\theta^{(j)}(\boldsymbol{\theta}) \ln p(\mathcal{X}, \mathcal{X}^l, \boldsymbol{\theta}; \boldsymbol{\xi}^{(j)}) \, \mathrm{d}\boldsymbol{\theta} &= \mathbb{E}_{q_\theta^{(j)}} \left[ \ln p(\mathcal{X}, \mathcal{X}^l, \theta; \boldsymbol{\xi}^{(j)}) \right] \\ &:= \ln \tilde{p}(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi}^{(j)}). \end{aligned} \tag{13.8}$$

The negative Kullback-Leibler divergence in Eq. (13.7) is maximized if we set

$$q_{\mathcal{X}^l}^{(j+1)}(\mathcal{X}^l) \propto \tilde{p}(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi}^{(j)}). \tag{13.9}$$

Elaborating on Eqs. (13.8) and (13.9) and denoting all quantities that do not depend on $\mathcal{X}^l$ as constants, we get

$$\ln q_{\mathcal{X}^l}^{(j+1)}(\mathcal{X}^l) = \mathbb{E}_{q_\theta^{(j)}} \left[ \ln p(\mathcal{X}, \mathcal{X}^l | \theta; \boldsymbol{\xi}^{(j)}) \right] + \text{constant}. \tag{13.10}$$

Hence, we can now write,

$$\boxed{q_{\mathcal{X}^l}^{(j+1)}(\mathcal{X}^l) = \frac{\exp\left( \mathbb{E}_{q_\theta^{(j)}} \left[ \ln p(\mathcal{X}, \mathcal{X}^l | \boldsymbol{\theta}; \boldsymbol{\xi}^{(j)}) \right] \right)}{\int \exp\left( \mathbb{E}_{q_\theta^{(j)}} \left[ \ln p(\mathcal{X}, \mathcal{X}^l | \theta; \boldsymbol{\xi}^{(j)}) \right] \right) \mathrm{d}\mathcal{X}^l},} \tag{13.11}$$

where the proportionality constant has necessarily been absorbed in the normalizing factor.

E-Step 1b: Freezing $\boldsymbol{\xi}^{(j)}$ and $q_{\mathcal{X}^l}^{(j+1)}$ and following similar steps as before (repeat the steps as an exercise), starting from the formulation in Eq. (13.6) and maximizing with respect to $q_\theta$, we obtain

$$\boxed{q_\theta^{(j+1)}(\boldsymbol{\theta}) = \frac{p(\boldsymbol{\theta}; \boldsymbol{\xi}^{(j)}) \exp\left( \mathbb{E}_{q_{\mathcal{X}^l}^{(j+1)}} \left[ \ln p(\mathcal{X}, \mathcal{X}^l | \boldsymbol{\theta}; \boldsymbol{\xi}^{(j)}) \right] \right)}{\int p(\boldsymbol{\theta}; \boldsymbol{\xi}^{(j)}) \exp\left( \mathbb{E}_{q_{\mathcal{X}^l}^{(j+1)}} \left[ \ln p(\mathcal{X}, \mathcal{X}^l | \boldsymbol{\theta}; \boldsymbol{\xi}^{(j)}) \right] \right) \mathrm{d}\boldsymbol{\theta}}.} \tag{13.12}$$

Steps 1a and 1b comprise the E-step of the variational Bayesian EM.

M-Step 2: Freezing $q_\theta^{(j+1)}$ and $q_{\mathcal{X}^l}^{(j+1)}$, maximize the lower bound with respect to $\boldsymbol{\xi}$, that is,

$$\boxed{\boldsymbol{\xi}^{(j+1)} = \arg\max_{\boldsymbol{\xi}} \mathcal{F}(q_\theta^{(j+1)}, q_{\mathcal{X}^l}^{(j+1)}; \boldsymbol{\xi}).} \tag{13.13}$$

The counterpart of the EM illustration of Figure 12.6 is given in Figure 13.1. There are two observations to be made. Step 1 is now split into two parts, and more important, the KL divergence does *not* (in

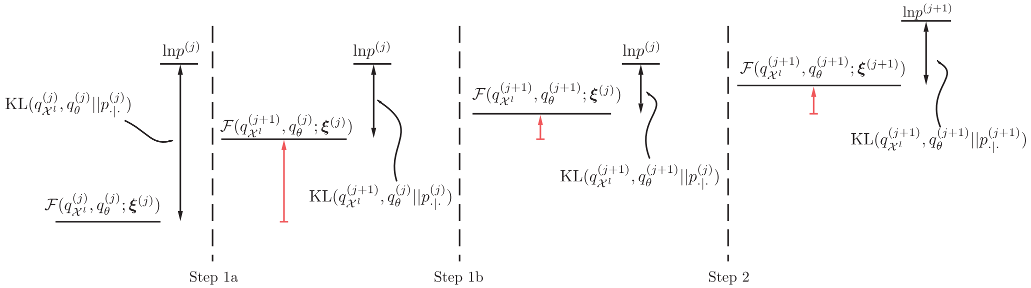

$\ln p^{(j)}$
$\text{KL}(q_{\mathcal{X}^l}^{(j)}, q_\theta^{(j)} || p_{\cdot|\cdot}^{(j)})$
$\mathcal{F}(q_{\mathcal{X}^l}^{(j)}, q_\theta^{(j)}; \boldsymbol{\xi}^{(j)})$
$\mathcal{F}(q_{\mathcal{X}^l}^{(j+1)}, q_\theta^{(j)}; \boldsymbol{\xi}^{(j)})$
$\ln p^{(j)}$
$\text{KL}(q_{\mathcal{X}^l}^{(j+1)}, q_\theta^{(j)} || p_{\cdot|\cdot}^{(j)})$
Step 1a
$\mathcal{F}(q_{\mathcal{X}^l}^{(j+1)}, q_\theta^{(j+1)}; \boldsymbol{\xi}^{(j)})$
$\ln p^{(j)}$
$\text{KL}(q_{\mathcal{X}^l}^{(j+1)}, q_\theta^{(j+1)} || p_{\cdot|\cdot}^{(j)})$
Step 1b
$\mathcal{F}(q_{\mathcal{X}^l}^{(j+1)}, q_\theta^{(j+1)}; \boldsymbol{\xi}^{(j+1)})$
$\ln p^{(j+1)}$
$\text{KL}(q_{\mathcal{X}^l}^{(j+1)}, q_\theta^{(j+1)} || p_{\cdot|\cdot}^{(j+1)})$
Step 2

**FIGURE 13.1**

Illustration of the stepwise increase of $\ln p^{(j)}$ at the $(j+1)$ iteration of the Variational Bayesian EM algorithm. Observe that $\ln p^{(j+1)} > \ln p^{(j)}$, where we have used the notation $p^{(j)} = p(\mathcal{X}; \boldsymbol{\xi}^{(j)})$ and $p_{\cdot|\cdot}^{(j)} := p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X}; \boldsymbol{\xi}^{(j)})$.

general) go to zero; hence the bound does not become tight. This comprises the E-step of the variational Bayesian EM.

If there are more than two factors in $q(\mathcal{X}^l, \boldsymbol{\theta})$, as in Eq. (13.4), then there are more than two substeps in step 1, and each time we estimate one of the factors by averaging $\ln p(\mathcal{X}, \mathcal{X}^l, \boldsymbol{\theta}; \boldsymbol{\xi})$ with respect to the rest. Let $q$ be factorized in $M$ factors

$$q(\mathcal{X}^l) = q_1(\mathcal{X}_1^l) \dots q_M(\mathcal{X}_M^l),$$

where for notational uniformity we have not differentiated between parameters and latent variables and the dependence on $\boldsymbol{\xi}$ has been suppressed. Then the general form of update becomes

$$\boxed{\ln q_m(\mathcal{X}_m^l) = \mathbb{E}\left[\ln p(\mathcal{X}, \mathcal{X}_1^l, \dots, \mathcal{X}_M^l)\right] + \text{constant},} \tag{13.14}$$

where the expectation is with respect to $\prod_{r=1_{r \neq m}}^{M} q_r(\mathcal{X}_r^l)$.

*Remarks 13.1.*

- Note that $q(\mathcal{X}^l, \boldsymbol{\theta})$ is an estimate of the posterior $p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X})$ and each one of the factors is the respective posterior estimate given the observations $\mathcal{X}$, for example, $q_\theta(\boldsymbol{\theta}) \simeq p(\boldsymbol{\theta}|\mathcal{X})$.
- Once $q(\mathcal{X}^l, \boldsymbol{\theta})$ is factorized, *no additional assumptions on the functional form* of $q_{\mathcal{X}^l}$ and $q_\theta$ are made.
- Note that factorization of a pdf implies independence. Thus, if this is not the case for the data at hand, the recovered approximations may not be faithful representations of the underlying data structure. Hence, choosing a specific factorization has to be carried out with care. In practice, one may have to use a number of alternatives and keep the best one. However, computational complexity is the other face of the coin, which one must consider in a trade-off game. In general, the factorized variational approach tends to provide approximations to the posterior pdf that are more compact than the true ones (e.g., [40]).
- Recall our discussion in Section 12.3 related to model selection and Occam's rule. This was a kick-off point for our efforts to maximize the evidence with respect to different models, in order to achieve the complexity-accuracy tradeoff. However, having resorted to approximate solutions (even if we forget convergence to local maxima, which somehow can be bypassed by using different initializations) we do not maximize the evidence but a lower bound of it; the latter is not,

in general, tight. How tight it is depends on the Kullback-Leibler divergence, which, unfortunately, cannot be trivially computed. Hence, if the lower bound is used for model selection, it has to be treated with caution [6].

- The variational approximation to Bayesian inference was first proposed in [21] and later on was used in a number of areas ranging from machine learning to decoding (e.g., [6, 22, 27–29, 31]).
- *Online Versions*: An online version of the variational Bayes algorithm was first proposed in [63]. There, the exponential family has been employed to show that parameter updating via the variational Bayes philosophy is equivalent to a natural gradient descent method (see Section 8.12, for the natural gradient) with step-size equal to one. This equivalence is further discussed in [24], where, similarly, a stochastic approximation algorithm is proposed in order to process chunks of data in parallel. An online variational Bayes algorithm for parameter estimation in the context of sparse linear regression modeling has also been proposed in [73].

### 13.2.1 THE CASE OF THE EXPONENTIAL FAMILY OF PROBABILITY DISTRIBUTIONS

Looking carefully at Eqs. (13.11) and (13.12), it becomes clear that the practical application of the variational Bayesian EM depends on the computational tractability of the expected values of the $\ln p(\mathcal{X}, \mathcal{X}^l|\boldsymbol{\theta};\boldsymbol{\xi})$. Let us now see the form that the iterative steps take when one adopts the pdf models from the exponential family.

Let us assume that the points in the complete data set $(\boldsymbol{x}_n, \boldsymbol{x}_n^l), n = 1, 2, \ldots, N$, are i.i.d. Then,

$$p(\mathcal{X}, \mathcal{X}^l|\boldsymbol{\theta}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n, \boldsymbol{x}_n^l|\boldsymbol{\theta}). \tag{13.15}$$

We further assume $p(\boldsymbol{x}_n, \boldsymbol{x}_n^l|\boldsymbol{\theta})$ to lie within the exponential family (Section 12.4), that is,

$$p(\boldsymbol{x}_n, \boldsymbol{x}_n^l|\boldsymbol{\theta}) = g(\boldsymbol{\theta}) f(\boldsymbol{x}_n, \boldsymbol{x}_n^l) \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta})\boldsymbol{u}(\boldsymbol{x}_n, \boldsymbol{x}_n^l)\right). \tag{13.16}$$

We also adopt a prior for $\boldsymbol{\theta}$ to be of the respective conjugate form, that is,

$$p(\boldsymbol{\theta}|\lambda, \boldsymbol{v}) = h(\lambda, \boldsymbol{v})(g(\boldsymbol{\theta}))^{\lambda} \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta})\boldsymbol{v}\right). \tag{13.17}$$

The parameters $\lambda$, $\boldsymbol{v}$ constitute $\boldsymbol{\xi}$, which will be considered fixed, because our current emphasis is to follow up the specific functional forms that $q_{\mathcal{X}^l}$ and $q_\theta$ get as iterations progress. So we relax the notational dependence on these parameters.

E-Step 1a: We have from Eq. (13.11) that

$$\begin{aligned} q_{\mathcal{X}^l}^{(j+1)}(\mathcal{X}^l) &\propto \exp\left(\mathbb{E}_{q_\theta^{(j)}}\left[\ln p(\mathcal{X}, \mathcal{X}^l|\boldsymbol{\theta})\right]\right) \\ &= \exp\left(\mathbb{E}_{q_\theta^{(j)}}\left[\sum_{n=1}^{N} \ln p(\boldsymbol{x}_n, \boldsymbol{x}_n^l|\boldsymbol{\theta})\right]\right) \\ &= \prod_{n=1}^{N} \exp\left(\mathbb{E}_{q_\theta^{(j)}}\left[\ln p(\boldsymbol{x}_n, \boldsymbol{x}_n^l|\boldsymbol{\theta})\right]\right), \end{aligned}$$

which then suggests that

$$q_{\boldsymbol{x}_n^l}^{(j+1)}(\boldsymbol{x}_n^l) \propto \exp\left(\mathbb{E}_{q_\theta^{(j)}}\left[\ln p(\boldsymbol{x}_n, \boldsymbol{x}_n^l|\boldsymbol{\theta})\right]\right),$$

and combined with Eq. (13.16) results in

$$q^{(j+1)}_{\boldsymbol{x}_n^l}(\boldsymbol{x}_n^l) = \tilde{g} f(\boldsymbol{x}_n, \boldsymbol{x}_n^l) \exp\left(\tilde{\boldsymbol{\phi}}^T \boldsymbol{u}(\boldsymbol{x}_n, \boldsymbol{x}_n^l)\right),$$

where $\tilde{g}$ is the respective normalization constant and

$$\tilde{\boldsymbol{\phi}}^T = \mathbb{E}_{q_\theta^{(j)}}[\boldsymbol{\phi}^T(\theta)]. \tag{13.18}$$

This is very interesting indeed. Although no functional form was assumed for $q_{\mathcal{X}^l}$, it turns out to be a member of the exponential family!

E-Step 1b: In a similar way, from (13.12), (13.15) and (13.16), we obtain

$$q_\theta^{(j+1)}(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta}) \exp\Bigg(N \ln g(\boldsymbol{\theta}) + \sum_{n=1}^{N} \mathbb{E}_{q^{(j+1)}_{x_n^l}}\left[\ln\left(f(\boldsymbol{x}_n, \mathbf{x}_n^l)\right)\right]$$
$$+ \boldsymbol{\phi}^T(\boldsymbol{\theta}) \sum_{n=1}^{N} \mathbb{E}_{q^{(j+1)}_{x_n^l}}\left[\boldsymbol{u}(\boldsymbol{x}_n, \mathbf{x}_n^l)\right]\Bigg),$$

which combined with Eq. (13.17) results in

$$q_\theta^{(j+1)}(\boldsymbol{\theta}) \propto (g(\boldsymbol{\theta}))^{\lambda+N} \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta})\left(\boldsymbol{v} + \sum_{n=1}^{N} \mathbb{E}_{q^{(j+1)}_{x_n^l}}\left[\boldsymbol{u}(\boldsymbol{x}_n, \mathbf{x}_n^l)\right]\right)\right). \tag{13.19}$$

Thus, the approximation $q_\theta^{(j+1)}(\boldsymbol{\theta})$ of the posterior $p(\boldsymbol{\theta}|\mathcal{X})$ is of the same form as the conjugate prior with

$$\tilde{\lambda} = \lambda + N, \quad \tilde{v} = v + \sum_{n=1}^{N} \mathbb{E}_{q^{(j+1)}_{x_n^l}}\left[\boldsymbol{u}(\boldsymbol{x}_n, \mathbf{x}_n^l)\right]. \tag{13.20}$$

Note that Eq. (13.20) is of the same form as Eq. (12.45). We only have to average out the hidden variables. This is a very elegant result, because nothing has been assumed about the functional form of $q_\theta$. In other words, once we adopt the functional form for the pdfs of the complete set as well as that of the prior of the parameters to be of the exponential type, then subsequent iterations become a “family business.”

## 13.3 A VARIATIONAL BAYESIAN APPROACH TO LINEAR REGRESSION

Once more, let us consider our familiar regression task

$$\mathbf{y} = \Phi\theta + \boldsymbol{\eta}, \quad \mathbf{y} \in \mathbb{R}^N, \; \theta \in \mathbb{R}^K.$$

In Section 12.6, we treated the case where $\boldsymbol{\eta}$ was Gaussian and the prior $p(\boldsymbol{\theta})$ was also Gaussian. We used the EM in order to optimize the evidence $p(\boldsymbol{y})$ with respect to the parameters that define the two adopted Gaussian pdfs; note that for this case, one could bypass the EM and resort to analytical computations in order to obtain the evidence and subsequently use an optimization technique to estimate the unknown parameters.

In this section, we will adopt assumptions that do not allow for tractable analytic computations of the posterior, $p(\boldsymbol{\theta}|\boldsymbol{y})$, which is a prerequisite both for the standard EM as well as for the analytic computations of the evidence $p(\boldsymbol{y})$. This approach is far from a pedagogic toy and has strong practical flavor. We will develop the task in some detail, and the reader is advised to go through the computations, because they are typical of what will be encountered in practice, once the variational Bayesian approach is chosen for addressing a task.

Assume that

$$p(\mathbf{y}|\boldsymbol{\theta}, \beta) = \mathcal{N}(\Phi\boldsymbol{\theta}, \beta^{-1}I). \tag{13.21}$$

That is, the noise is Gaussian and for simplicity we have considered it to be white, $\Sigma_\eta = \sigma_\eta^2 I$, and $\beta = \frac{1}{\sigma_\eta^2}$. In contrast to what we did in Section 12.6, now we will be more democratic and give the freedom to each one of the parameter components, $\theta_k$, to have a different variance, $\sigma_k^2 := \frac{1}{\alpha_k}, k = 0, 1, \ldots, K-1$. Moreover, we go one step further. The values of $\beta$ and $\alpha_k$, $k = 0, \ldots, K-1$, will not be treated as deterministic variables. We will also treat them as random ones, ($\beta$, $\alpha_k$), which will be assigned prior pdfs; these prior pdfs are in turn controlled by another set of hyperparameters. More specifically, our model, in addition to Eq. (13.21) comprises [7]

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \prod_{k=0}^{K-1} \mathcal{N}(\theta_k|0, \alpha_k^{-1}), \tag{13.22}$$

$$p(\boldsymbol{\alpha}) = \prod_{k=0}^{K-1} \text{Gamma}(\alpha_k|a, b), \tag{13.23}$$

and

$$p(\beta) = \text{Gamma}(\beta|c, d). \tag{13.24}$$

Note that the previous choice of the priors indicates our will to "play" the game within the exponential family terrain. The prior $p(\boldsymbol{\alpha})$ is the conjugate pair of Eq. (13.22) (see Chapter 12). Also, Eq. (13.24) would be the conjugate of Eq. (13.21), if we had considered $\boldsymbol{\theta}$ fixed. Figure 13.2 provides a graphical

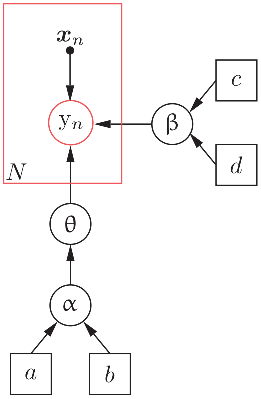


**FIGURE 13.2**

A graphical illustration of the dependencies among the various variables involved in the model of linear regression. The red circle indicates the random variable that is observed, gray circles indicate hidden random variables, and squares correspond to deterministic parameters. The direction of each arrow indicates the direction of the dependence between the connected variables. The red box indicates that the above dependencies hold for all, *N*, time instants.

representation of the dependencies among the various variables involved in our model. Arrows indicate conditional dependencies. Graphical models will be considered in a formal way in Chapter 15. Note that such a model forms various levels of *hierarchy* in the dependency among the involved parameters.

This concept of hierarchy is at the heart of what we call *hierarchical Bayesian modeling*. Each one of the involved pdfs is expressed in terms of certain parameters. Because the values of these parameters are unknown, they are also treated as random variables whose priors are expressed in terms of a new set of hyperparameters. Each one of them is in turn treated as a random variable associated with a new prior, known as *hyperprior*. This rationale can be extended in order to construct different levels of hierarchy. Often, at the higher level of hierarchy, the corresponding (unknown) hyperparameters are assigned values by the user, based on experience; for example, the overall model can be relatively insensitive to their specific values, which makes the corresponding choice a fairly easy job.

Our current task comprises hidden variables in the form of parameters grouped in $\theta$, $\boldsymbol{\alpha}$, and $\beta$ and it involves no other latent variables. The set of observations is now given by $\boldsymbol{y}$. Also, observe that the posterior $p(\boldsymbol{\theta}, \boldsymbol{\alpha}, \beta|\boldsymbol{y})$ is not analytically tractable. We will resort to the variational Bayesian EM to obtain an estimate of the previous posterior pdf.

Using the mean field approximation, we assume that the approximation to the posterior (the dependence on $\boldsymbol{y}$ has been suppressed for notational convenience) factorizes as

$$q(\boldsymbol{\theta}, \boldsymbol{\alpha}, \beta) = q_\theta(\boldsymbol{\theta}) q_\alpha(\boldsymbol{\alpha}) q_\beta(\beta), \tag{13.25}$$

where we have relaxed our notation, for simplicity, from the explicit dependence on $a$, $b$, $c$, and $d$. We will bring them back into the game whenever needed. The variational EM consists of three substeps, one for each factor in Eq. (13.25). Starting from some initial guesses, for $\mathbb{E}[\beta]$, $\mathbb{E}[\alpha_k], k = 0, \ldots, K-1$, (it will become clear soon why we need to start with those[1]) we get:

E-Step 1a: From the general update form of Eq. (13.14) we have,

$$\ln q_\theta^{(j+1)}(\boldsymbol{\theta}) = \mathbb{E}_{q_\alpha^{(j)} q_\beta^{(j)}} \Big[ \ln p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \beta) \Big] + \text{constant}, \tag{13.26}$$

where now

$$\begin{aligned} \ln p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \beta) &= \ln \big( p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{\alpha}, \beta) p(\boldsymbol{\theta}, \boldsymbol{\alpha}, \beta) \big) \\ &= \ln \big( p(\boldsymbol{y}|\boldsymbol{\theta}, \beta) p(\boldsymbol{\theta}|\boldsymbol{\alpha}) p(\boldsymbol{\alpha}) p(\beta) \big), \end{aligned} \tag{13.27}$$

where the independence of $\mathbf{y}$ on $\boldsymbol{\alpha}$, given the values $\boldsymbol{\theta}$, has been taken into account. Using Eqs. (13.21), (13.24) and some trivial algebra we get that

$$\begin{aligned} \ln p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \beta) = \ln \frac{\beta^{N/2}}{(2\pi)^{N/2}} - \frac{\beta}{2} \|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|^2 - \frac{1}{2} \sum_{k=0}^{K-1} a_k \theta_k^2 \\ + \sum_{k=0}^{K-1} \ln \sqrt{\frac{\alpha_k}{2\pi}} + \ln p(\boldsymbol{\alpha}) + \ln p(\beta) \end{aligned}$$

or

$$\ln p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \beta) = -\frac{\beta}{2} \|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|^2 - \frac{1}{2} \sum_{k=0}^{K-1} a_k \theta_k^2 + \text{constant}, \tag{13.28}$$

[1] If $a, b, c, d$ were not fixed, then one would need initialization for these parameters, too.

where constant includes all terms that do not depend on $\boldsymbol{\theta}$, because in this step our goal is to estimate a function of $\boldsymbol{\theta}$. Expanding Eq. (13.28) and taking expectations w.r. to $\beta$ and $\boldsymbol{\alpha}$, considering $q_\beta^{(j)}(\beta)$ and $q_\alpha^{(j)}(\boldsymbol{\alpha})$ known, we get

$$\begin{aligned}\ln q_\theta^{(j+1)}(\boldsymbol{\theta}) &= \mathbb{E}_{q_\beta^{(j)} q_\alpha^{(j)}}\left[\ln p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \beta)\right] + \text{constant} = -\frac{1}{2}\,\mathbb{E}[\beta]\boldsymbol{\theta}^T \Phi^T \Phi \boldsymbol{\theta} \\ &\quad - \frac{1}{2}\,\mathbb{E}[\beta]\boldsymbol{y}^T\boldsymbol{y} + \mathbb{E}[\beta]\boldsymbol{\theta}^T\Phi^T\boldsymbol{y} - \frac{1}{2}\boldsymbol{\theta}^T A\boldsymbol{\theta} + \text{constant}, \qquad (13.29)\end{aligned}$$

where by definition

$$A := \text{diag}\left\{\mathbb{E}[\alpha_0], \ldots, \mathbb{E}[\alpha_{K-1}]\right\},$$

and we have used for notational simplifications

$$\mathbb{E}[\beta] := E_{q_\beta^{(j)}}[\beta] \quad \text{and} \quad \mathbb{E}[\alpha_k] := E_{q_\alpha^{(j)}}[\alpha_k], \quad k = 0, 1, 2, \ldots, K-1. \qquad (13.30)$$

It is readily noticed that the right-hand side of Eq. (13.29) is of a quadratic form with respect to $\boldsymbol{\theta}$, hence $q_\theta^{(j+1)}(\boldsymbol{\theta})$ is Gaussian; in order to completely specify it, it suffices to compute the respective mean and covariance (precision) matrix.

Reshuffling the terms in Eq. (13.29), we get

$$\ln q_\theta^{(j+1)}(\boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{\theta}^T(A + \mathbb{E}[\beta]\Phi^T\Phi)\boldsymbol{\theta} + \mathbb{E}[\beta]\boldsymbol{\theta}^T\Phi^T\boldsymbol{y} + \text{constant},$$

which according to Eqs. (12.111), (12.113) and (12.114) of Section 12.9 of the previous chapter results in

$$q_\theta^{(j+1)}(\boldsymbol{\theta}) = \mathcal{N}(\theta|\mu_\theta^{(j+1)}, \Sigma_\theta^{(j+1)}), \qquad (13.31)$$

$$\Sigma_\theta^{(j+1)} = (A + \mathbb{E}[\beta]\Phi^T\Phi)^{-1}, \qquad (13.32)$$

and

$$\boldsymbol{\mu}_\theta^{(j+1)} = \mathbb{E}[\beta]\Sigma_\theta^{(j+1)}\Phi^T\boldsymbol{y}. \qquad (13.33)$$

During the first iteration step, $\mathbb{E}[\beta]$ and $\mathbb{E}[\alpha_k]$ are provided by their initial values. For the subsequent iterations, they have to be obtained together with $q_\beta^{(j)}$ and $q_\alpha^{(j)}$. Note that the approximation to the posterior $p(\boldsymbol{\theta}|\boldsymbol{y})$ turns out to be Gaussian, although we did not assume it to be so. This is a consequence of the particular form of the adopted pdfs, which spring from the exponential family.

E-Step 1b:

$$\begin{aligned}\ln q_\alpha^{(j+1)}(\boldsymbol{\alpha}) &= \mathbb{E}_{q_\theta^{(j+1)} q_\beta^{(j)}}\left[\ln p(\boldsymbol{y}, \theta, \boldsymbol{\alpha}, \beta)\right] + \text{constant} \qquad (13.34)\\ &= \mathbb{E}_{q_\theta^{(j+1)} q_\beta^{(j)}}\left[\ln p(\theta|\boldsymbol{\alpha}) + \ln p(\boldsymbol{\alpha})\right] + \text{constant}, \qquad (13.35)\end{aligned}$$

where the constant contains all terms that do not depend on $\boldsymbol{\alpha}$. Because no term in the bracket in the right-hand side of Eq. (13.35) depends on $\beta$, we have

$$\ln q_\alpha^{(j+1)}(\boldsymbol{\alpha}) = E_{q_\theta^{(j+1)}}\left[\frac{1}{2}\sum_{k=0}^{K-1}\ln\alpha_k - \frac{1}{2}\sum_{k=0}^{K-1}\alpha_k\theta_k^2\right] + \ln p(\boldsymbol{\alpha}) + \text{constant}. \qquad (13.36)$$

Taking into account Eq. (13.23) and after some algebra (Problem 13.2), we obtain

$$q_\alpha^{(j+1)}(\boldsymbol{\alpha}) = \prod_{k=0}^{K-1} \text{Gamma}(\alpha_k|\tilde{a}, \tilde{b}_k), \tag{13.37}$$

where

$$\tilde{a} = a + \frac{1}{2} \tag{13.38}$$

$$\tilde{b}_k = b + \frac{1}{2}\,\mathbb{E}_{q_\theta^{(j+1)}}[\theta_k^2], \quad k = 0, \ldots, K-1. \tag{13.39}$$

In order to compute $\mathbb{E}[\theta_k^2]$, recall Eqs. (12.73) and (12.74) and apply them into our setting to give

$$\mathbb{E}_{q_\theta^{(j+1)}}[\boldsymbol{\theta}\boldsymbol{\theta}^T] = \Sigma_\theta^{(j+1)} + \boldsymbol{\mu}_\theta^{(j+1)}\boldsymbol{\mu}_\theta^{(j+1)T},$$

or

$$\mathbb{E}[\theta_k^2] = \left[E_{q_\theta^{(j+1)}}[\boldsymbol{\theta}\boldsymbol{\theta}^T]\right]_{kk} = \left[\Sigma_\theta^{(j+1)} + \boldsymbol{\mu}_\theta^{(j+1)}\boldsymbol{\mu}_\theta^{(j+1)T}\right]_{kk}, \quad k = 0, 1, \ldots, K-1, \tag{13.40}$$

where $[A]_{kk}$ denotes the $(k, k)$ element of a matrix, $A$. To complete the computations, we have to compute $\mathbb{E}[\alpha_k], k = 0, 1, \ldots, K-1$, to be used during the next iteration in Eq. (13.32). However, because each $\alpha_k$ follows a gamma distribution, we know that (Section 2.3.2)

$$\mathbb{E}_{q_\alpha^{(j+1)}}[\alpha_k] = \frac{\tilde{a}}{\tilde{b}_k}. \tag{13.41}$$

E-Step 1c:

$$\begin{aligned} \ln q_\beta^{(j+1)}(\beta) &= \mathbb{E}_{q_\theta^{(j+1)} q_\alpha^{(j+1)}}\left[\ln p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \beta)\right] + \text{constant} \\ &= \mathbb{E}_{q_\theta^{(j+1)} q_\alpha^{(j+1)}}\left[\ln p(\boldsymbol{y}|\boldsymbol{\theta}, \beta) + \ln p(\beta)\right] + \text{constant}. \end{aligned}$$

This is of the same form as Eq. (13.35), and following similar steps (Problem 13.3), it can be shown that

$$q_\beta^{(j+1)}(\beta) = \text{Gamma}(\beta|\tilde{c}, \tilde{d}), \tag{13.42}$$

$$\tilde{c} = c + \frac{N}{2}, \tag{13.43}$$

$$\tilde{d} = d + \frac{1}{2}\,\mathbb{E}_{q_\theta^{(j+1)}}[\|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|^2]. \tag{13.44}$$

To compute the expectation in Eq. (13.44), recall Eq. (12.76), which for our needs becomes

$$\mathbb{E}_{q_\theta^{(j+1)}}[\|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|^2] = \|\boldsymbol{y} - \Phi\boldsymbol{\mu}_\theta^{(j+1)}\|^2 + \text{trace}\left\{\Phi\Sigma_\theta^{(j+1)}\Phi^T\right\}. \tag{13.45}$$

Finally, we have that

$$\mathbb{E}_{q_\beta^{(j+1)}}[\beta] = \frac{\tilde{c}}{\tilde{d}}, \tag{13.46}$$

which completes all the computations associated with the E-step of the variational EM. Note that $q_{\boldsymbol{\alpha}}^{(j+1)}(\boldsymbol{\alpha}) \simeq p(\boldsymbol{\alpha}|\boldsymbol{y})$ and $q_\beta^{(j+1)}(\beta) \simeq p(\beta|\boldsymbol{y})$ retain the gamma functional form of the corresponding priors that were originally adopted, without forcing them to.

In principle, one can add an extra M-step in the algorithm to maximize the bound with respect to the unknown parameters $a$, $b$, $c$, and $d$. However, in practice, for computational simplicity these parameters are fixed to very small values, that is, $a = b = c = d = 10^{-6}$, which correspond to *uninformative* gamma prior distributions, in the sense of giving no preference to any specific range of values. Note that for such small values, the gamma distribution falls as $\frac{1}{x}$. Indeed, for $a,\ b \simeq 0$

$$\text{Gamma}(x|a,b) \simeq \frac{1}{x}, \quad x > 0.$$

Because every positive $x$ can be expressed as

$$x = \exp(z), \quad z = \ln x, \quad z \in \mathbb{R},$$

then it can be easily checked out (Problem 13.4) that the pdf that describes $z$ is uniform. This is a typical procedure in practice; that is, one allows enough levels of hierarchy and fixes the hyperparameters in the highest level to define uninformative hyperpriors.

In summary, the variational Bayesian EM steps are given in Algorithm 13.1.

**Algorithm 13.1 (Variational EM for linear regression).**

```
• Initialization
    • Select initial values for $\mathbb{E}[\beta], \mathbb{E}_{q_\alpha}[\alpha_k], \quad k = 0, 1, \ldots, K-1$.
• For, $j = 1, 2, \ldots,$ Do
    • A=diag$\{\mathbb{E}_{q_\alpha}[\alpha_0, ], \mathbb{E}_{q_\alpha}[\alpha_1], \ldots, \mathbb{E}_{q_\alpha}[\alpha_{K-1}]\}$.
    • Compute $\Sigma_\theta$ from Eq. (13.32) and $\boldsymbol{\mu}_\theta$ from Eq. (13.33).
    • Compute $\tilde{a}$ from Eq. (13.38).
    • Compute $\tilde{b}_k,\ k = 0, 1, \ldots, K-1$, from Eqs. (13.40) and (13.39).
    • Compute $\mathbb{E}_{q_\alpha}[\alpha_k],\ k = 0, 1, \ldots, K-1$, from Eq. (13.41).
    • Compute $\tilde{c}$ from Eq. (13.43) and $\tilde{d}$ from Eqs. (13.45) and (13.44).
    • Compute $\mathbb{E}_{q_\beta}[\beta]$ from Eq. (13.46).
    • If convergence criterion is met, Stop.
• End For
```

Once the algorithm has converged, predictions can be made on the basis of the predictive distribution given in Eqs. (12.21)–(12.23), by replacing $\Sigma_{\theta|y}$, $\boldsymbol{\mu}_{\theta|y}$ and $\sigma_\eta^2$ by the converged values of $\Sigma_\theta$, $\boldsymbol{\mu}_\theta$, and $\mathbb{E}[\beta]$, respectively. Note, however, that this is only an approximation, because the Gaussian form for the posterior of the parameters is a result of the mean field approximation and also we have used the mean value, $\mathbb{E}[\beta]$, in place of the noise variance. The latter can be justified that as the number of training samples increases, the distribution of $\beta$ sharply peaks around its mean value, [7].

### Computation of the lower bound

Once the algorithm has converged, the quantities $q_\theta(\boldsymbol{\theta}), q_\alpha(\boldsymbol{\alpha}), q_\beta(\beta)$ are available and the lower bound $\mathcal{F}(q_\theta, q_\alpha, q_\beta)$ can be computed. The computation of this lower bound can also be done at every iteration to check how much it changes from iteration to iteration, and then this can be used as a convergence criterion. Let $\tilde{q}_\theta, \tilde{q}_\alpha, \tilde{q}_\beta$, be the approximate posteriors after convergence, defined by the parameters $\tilde{\Sigma}_\theta, \tilde{\boldsymbol{\mu}}_\theta, \tilde{a}, \tilde{b}_k, k = 0, 1, 2, \ldots, K-1, \tilde{c}$, and $\tilde{d}$. The lower bound is then given as

$$\mathcal{F}(\tilde{q}_\theta, \tilde{q}_\alpha, \tilde{q}_\beta) = \mathbb{E}_{\tilde{q}_\theta \tilde{q}_\alpha \tilde{q}_\beta}\left[\ln p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \beta)\right] - \mathbb{E}_{\tilde{q}_\theta}\left[\ln \tilde{q}_\theta(\boldsymbol{\theta})\right] - \mathbb{E}_{\tilde{q}_\alpha}\left[\ln \tilde{q}_\alpha(\boldsymbol{\alpha})\right] - \mathbb{E}_{\tilde{q}_\beta}\left[\ln \tilde{q}_\beta(\beta)\right], \tag{13.47}$$

Performing the expectations can be a bit tedious, but it is straightforward (Problem 13.5).

## 13.4 A VARIATIONAL BAYESIAN APPROACH TO GAUSSIAN MIXTURE MODELING

Dealing with the Gaussian mixture modeling in Section 12.7, it was pointed out in the remarks that the standard EM approach may lead to singularities. One way to bypass this drawback is to enforce priors on the involved parameters and resort to a variational Bayesian philosophy to estimate the quantities of interest. The task was first treated in [3] and later in [11]. We will present the latter approach and comment on the underlying differences between the two later on.

Given a set of observations, $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, the respective pdf model is

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} P_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, Q_k^{-1}), \quad \boldsymbol{x} \in \mathbb{R}^l.$$

The task is to estimate the unknown parameters $(P_k, \boldsymbol{\mu}_k, Q_k), k = 1, 2, \ldots, K$. We already know that this is a typical task with latent variables, and the complete set comprises $(\boldsymbol{x}_n, k_n), n = 1, 2, \ldots, N$, with $k_n$ being the index of the respective mixture, $k_n = 1, 2, \ldots, K$. In section 12.7, the information about each one of the latent variables, $k_n$, entered into the problem via the posterior $P(k_n|\boldsymbol{x}_n)$ for every time instant $n$, the summation over all possible values of $k_n$ was performed, hence one could drop out the time index. However, in the current context, a different path has to be followed and we have to consider the latent variables together with their corresponding time index. To this end, and following [11], an auxiliary latent random vector is introduced, $\mathbf{z}_n \in \mathbb{R}^K$, for each observation, $n = 1, 2, \ldots, N$. Its components take binary values, such as

$$z_{n_k} \in \{0, 1\}, \quad \text{and} \quad \sum_{k=1}^{K} z_{n_k} = 1, \tag{13.48}$$

and they are used as indicators of the respective mixture from which the observation at time $n$, $\boldsymbol{x}_n$, was drawn; that is, if $z_{n_k} = 1$ it indicates that $\boldsymbol{x}_n$ was drawn from the $k$th distribution. Obviously,

$$P(z_{n_k} = 1) = P_k,$$

and for any $\boldsymbol{z}_n \in \mathbb{R}^K$ that satisfies Eq. (13.48),

$$P(\boldsymbol{z}_n) = \prod_{k=1}^{K} P_k^{z_{n_k}}. \tag{13.49}$$

Hence, the probability of occurrence of the set $\mathcal{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$ is

$$P(\mathcal{Z}) = \prod_{n=1}^{N} \prod_{k=1}^{K} P_k^{z_{n_k}}, \tag{13.50}$$

and in this way, we have described the random nature of the $N$ latent variables using a multinomial probability distribution.

In the sequel, we will treat the mean values as well as the precision matrices as random quantities adopting the following prior pdfs,

$$p(\boldsymbol{\mu}_k) = \mathcal{N}\left(\boldsymbol{\mu}_k|0, \beta^{-1}I\right)$$

and

$$p(Q_k) = \mathcal{W}(Q_k|W_0, \nu_0),$$

for fixed $\nu_0$, $W_0$, and $\beta$. That is, the adopted priors are Gaussian for the mean values and Wishart pdfs for the precision matrices, respectively. We will treat $\boldsymbol{P} = [P_1, \ldots, P_K]^T$ as deterministic parameters whose optimized value is obtained in the M-step.

Following the philosophy of the variational Bayesian EM, we adopt

$$q(\mathcal{Z}, \boldsymbol{\mu}_{1:K}, Q_{1:K}) = q_z(\mathcal{Z})q_\mu(\boldsymbol{\mu}_{1:K})q_Q(Q_{1:K}),$$

where $\boldsymbol{\mu}_{1:K}$ and $Q_{1:K}$ indicate the collections $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$ and $\{Q_1, \ldots, Q_K\}$, respectively.

Furthermore, observe that the conditional pdf of the observations can now be written as

$$p(\mathcal{X}|\mathcal{Z}, \boldsymbol{\mu}_{1:K}, Q_{1:K}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\left(\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, Q_k^{-1})\right)^{z_{nk}}.$$

Figure 13.3 shows the corresponding graphical model.

**Computational Steps of The Variational EM for Gaussian Mixture Modeling**

Initialization: (a) $\boldsymbol{P}^{(0)}$, (b) $\mathbb{E}_{q_Q^{(0)}}[\mathrm{Q}_k]$, (c) $\mathbb{E}_{q_Q^{(0)}}[\ln|\mathrm{Q}_k|]$, (d) $\mathbb{E}_{q_\mu^{(0)}}[\boldsymbol{\mu}_k] := \tilde{\boldsymbol{\mu}}_k^{(0)}$, and (e) $\mathbb{E}_{q_\mu^{(0)}}[\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T] := \tilde{\Sigma}_k^{(0)} + \tilde{\boldsymbol{\mu}}_k^{(0)}\tilde{\boldsymbol{\mu}}_k^{(0)T}$, $k = 1, 2, \ldots, K$, where $|\cdot|$ denotes the corresponding determinant.

The $(j+1)$ iteration consists of the following computations (Problem 13.6):

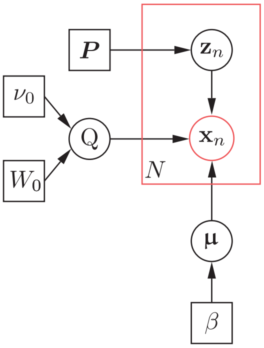


**FIGURE 13.3**

The graphical model associated with the Gaussian mixture modeling of Section 13.4.

E-Step 1a:

$$\begin{aligned}
\pi_{n_k} &= P_k^{(j)} \exp\Big(\frac{1}{2}\,\mathbb{E}_{q_Q^{(j)}}\,[\ln|\mathrm{Q}_k|] - \frac{1}{2}\mathrm{trace}\big\{\,\mathbb{E}_{q_Q^{(j)}}[\mathrm{Q}_k]\big(\boldsymbol{x}_n\boldsymbol{x}_n^T \\
&\qquad -\boldsymbol{x}_n\,\mathbb{E}_{q_\mu^{(j)}}[\boldsymbol{\mu}_k^T] - \mathbb{E}_{q_\mu^{(j)}}[\boldsymbol{\mu}_k]\boldsymbol{x}_n^T + \mathbb{E}_{q_\mu^{(j)}}[\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T]\big)\big\}\Big) \\
\rho_{n_k} &= \frac{\pi_{n_k}}{\sum_{k=1}^K \pi_{n_k}} \\
q_z^{(j+1)}(\mathcal{Z}) &= \prod_{n=1}^N\prod_{k=1}^K \rho_{n_k}^{z_{n_k}}.
\end{aligned}$$

E-Step 1b:

$$\begin{aligned}
\tilde{Q}_k &= \beta I + \mathbb{E}_{q_Q^{(j)}}[\mathrm{Q}_k]\sum_{n=1}^N \rho_{n_k} \\
\tilde{\boldsymbol{\mu}}_k &= \tilde{Q}_k^{-1}\,\mathbb{E}_{q_Q^{(j)}}[\mathrm{Q}_k]\sum_{n=1}^N \rho_{n_k}\boldsymbol{x}_n \\
q_\mu^{(j+1)}(\boldsymbol{\mu}_{1:K}) &= \prod_{k=1}^K \left(\boldsymbol{\mu}_k|\tilde{\boldsymbol{\mu}}_k, \tilde{Q}_k^{-1}\right),
\end{aligned}$$

and adopting Eq. (12.74) to the current needs,

$$\mathbb{E}_{q_\mu^{(j+1)}}[\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T] = \tilde{\Sigma}_k + \tilde{\boldsymbol{\mu}}_k\tilde{\boldsymbol{\mu}}_k^T = \tilde{Q}_k^{-1} + \tilde{\boldsymbol{\mu}}_k\tilde{\boldsymbol{\mu}}_k^T.$$

E-Step 1c:

$$\begin{aligned}
\tilde{\nu}_k &= \nu + \sum_{n=1}^N \rho_{n_k} \\
\tilde{W}_k^{-1} &= \tilde{W}_0^{-1} + \sum_{n=1}^N \rho_{n_k}\left(\boldsymbol{x}_n\boldsymbol{x}_n^T - \tilde{\boldsymbol{\mu}}_k\boldsymbol{x}_n^T - \boldsymbol{x}_n\tilde{\boldsymbol{\mu}}_k^T + \mathbb{E}_{q_\mu^{(j+1)}}[\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T]\right) \\
q_Q^{(j+1)}(Q_{1:K}) &= \prod_{k=1}^K \mathcal{W}(Q_k|\tilde{\nu}_k, \tilde{W}_k) \\
\mathbb{E}_{q_Q^{(j+1)}}[\mathrm{Q}_k] &= \tilde{\nu}_k\tilde{W}_k \\
\mathbb{E}_{q_Q^{(j+1)}}\Big[\ln|\mathrm{Q}_k|\Big] &= \sum_{i=1}^l \psi\left(\frac{\tilde{\nu}_{k+1-i}}{2}\right) + l\ln 2 + \ln|\tilde{W}_k|
\end{aligned}$$

where $\psi(\cdot)$ is the *digamma* function, defined as

$$\psi(a) := \frac{d\ln\Gamma(a)}{da},$$

and the gamma function has been defined in Eq. (2.91).

M-Step 2:

$$P_k^{(j+1)} = \frac{1}{N}\sum_{n=1}^{N} \rho_{n_k}.$$

The previous steps have concluded the algorithm. Observe that the iterations retain the functional form of the pdfs that were adopted for the respective priors; this is a consequence of their exponential family origin. In [11], it is suggested that this procedure can also be used to determine the number of mixtures, instead of adopting a cross-validation technique, as pointed out in the remarks of Section 12.7. By adopting a large enough value for $K$, the probabilities $P_k$ associated with the irrelevant components will be driven to zero during the M-step. Note that such a modeling is possible in the Bayesian framework, because it automatically achieves a trade-off between model complexity and data fitting. In [3], the probabilities $P_k,\ k = 1, 2, \ldots, K$, were considered as random variables and a Dirichlet prior was also imposed on them (Problem 13.7). However, such priors need to be selected with some care, otherwise it may affect the sparsification potential of the algorithm (e.g., [8]).

**Example 13.1.** The purpose of this example is to demonstrate the power of the variational Bayesian method for mixture modeling compared to the more classical EM algorithm, which was discussed in Section 12.7. Five clusters of data were generated using a corresponding number of Gaussians, as shown in Figure 13.4. The parameters used for each one of these Gaussians were

$$\boldsymbol{\mu}_1 = [-2.5, 2.5]^T, \quad \boldsymbol{\mu}_2 = [-4.0, -2.0]^T, \quad \boldsymbol{\mu}_3 = [2.0, -1.0]^T,$$

$$\boldsymbol{\mu}_4 = [0.1, 0.2]^T, \quad \boldsymbol{\mu}_5 = [3.0, 3.0]^T,$$

and

$$\Sigma_1 = \begin{bmatrix} 0.5 & 0.081 \\ 0.081 & 0.7 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 0.4 & 0.02 \\ 0.002 & 0.3 \end{bmatrix}, \quad \Sigma_3 = \begin{bmatrix} 0.6 & 0.531 \\ 0.531 & 0.9 \end{bmatrix},$$
$$\Sigma_4 = \begin{bmatrix} 0.5 & 0.22 \\ 0.22 & 0.8 \end{bmatrix}, \quad \Sigma_5 = \begin{bmatrix} 0.88 & 0.2 \\ 0.2 & 0.22 \end{bmatrix}.$$

Prior to running the algorithms, we assumed that we do not know the exact number of mixtures, so a number of $K = 25$ clusters was used, that is, a much larger number than the true one.

For the EM algorithm, the initial mean values were generated randomly, using a Gaussian $\mathcal{N}(\boldsymbol{\mu} \mid \boldsymbol{0}, I)$ and the respective initial covariance matrices, $\Sigma_k^{(0)}$, $k = 1, 2, \ldots, 25$, with random elements, making sure that it is positive definite. One way to achieve this is to generate a matrix $\Phi$ with random elements from a $\mathcal{N}(0, 1)$ and then form $\Phi^T\Phi$. Another possibility is to start with a diagonal matrix, for example, the identity one $I$.

For the Variational EM algorithm, the following initial values were used: the mean values, $\tilde{\boldsymbol{\mu}}_k^{(0)}$ and the initial covariance matrices, $\tilde{\Sigma}_k^{(0)}$, $k = 1, 2, \ldots, 25$, were generated as before. Also, $\mathbb{E}_{q_Q^{(0)}}[Q_k] = I$, $\mathbb{E}_{q_Q^{(0)}}[\ln |Q_k|] = 1$. In both cases, the initial probabilities were set to be equal.

Observe that the variational EM identifies the five clusters associated with the data; the rest of the mixtures correspond to zero probability weights. In contrast, the EM algorithm tries to identify all 25 mixtures and the result is not satisfactory.

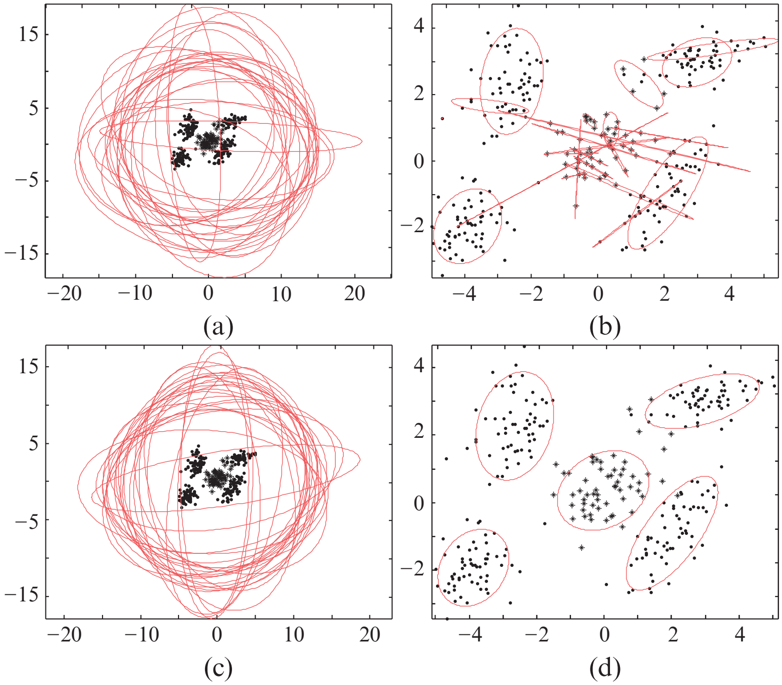


**FIGURE 13.4**

Figure for Example 13.1: (a) The initial (25) Gaussians for the EM algorithm. (b) The final clusters obtained after convergence by the EM algorithm. (c) The initial (25) Gaussians for the variational EM. (d) The final Gaussians obtained by the variational EM, after convergence. All the curves correspond to the 80% probability regions. Observe that the variational EM identifies the five clusters associated with the data; the rest of the mixtures correspond to zero probability weights.

## 13.5 WHEN BAYESIAN INFERENCE MEETS SPARSITY

The Bayesian approach to sparsity-aware learning will soon become our major concern. However, we will use this subsection to "warm" us up. The close relationship between the use of a prior pdf and the regularization of a cost function has already been discussed in Section 12.2.2. There, the adoption of a Gaussian prior together with a Gaussian noise for the regression task led to the equivalence of MAP with the ridge regression. It will not take a minute to show that the use of a Gaussian model for the noise together with a Laplacian prior for each one of the weights, that is,

$$p(\theta_k) = \frac{\lambda}{2} \exp\big(-\lambda|\theta_k|\big),$$

renders MAP equivalent to the $\ell_1$ norm regularization of the LS cost. For a Bayesian, however, who is not interested in cost functions, the secret that lies within the Laplacian prior is hidden in the heavy tails of this distribution. This is in contrast to a Gaussian pdf, which has very light tails. In other words, the probability that an observation of a Gaussian random variable can take values far from its mean decreases very fast. For example, the probability of observing variables that deviate from the mean by more than $2\sigma$, $3\sigma$, $4\sigma$, and $5\sigma$ are 0.046, 0.003, $6 \times 10^{-5}$, and $6 \times 10^{-7}$, respectively. That is, if we provide a Gaussian prior, we basically inform the learning process to look for values "around" the mean; values away from the mean are heavily penalized. However, in sparsity-aware learning this would be the wrong information to pass over to our learning mechanism. Assuming the mean of the prior to be zero, although we expect most of the components of our parameters to be zero, still we want a few of them to be large. Hence, our prior information should be selected so as to assign small (but not too small) probabilities to large values. Hence, to a Bayesian, sparsity-aware learning becomes synonymous with imposing heavy-tail priors. Let us now turn back to our current task, and see how this brief introduction is related to our model. Our prior pdf, $p(\boldsymbol{\theta})$, according to the model Eqs. (13.22)–(13.23) is obtained by marginalizing out the hyperparameters $\boldsymbol{\alpha}$ (Problem 13.10), that is,

$$\begin{aligned} p(\boldsymbol{\theta}; a, b) &= \int p(\boldsymbol{\theta}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})\, d\boldsymbol{\alpha} \\ &= \int \prod_{k=0}^{K-1} \mathcal{N}(\theta_k|0, \alpha_k^{-1})\text{Gamma}(\alpha_k|a, b)\, d\boldsymbol{\alpha} \\ &= \prod_{k=0}^{K-1} \text{st}\left(\theta_k|0, \frac{a}{b}, 2a\right), \end{aligned} \tag{13.51}$$

where $\text{st}(x|\mu, \lambda, \nu)$ is the student's-t pdf, defined by

$$\text{st}(x|\mu, \lambda, \nu) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})} \left(\frac{\lambda}{\pi\nu}\right)^{1/2} \frac{1}{\left(1 + \frac{\lambda(x-\mu)^2}{\nu}\right)^{\frac{\nu+1}{2}}}. \tag{13.52}$$

The parameter $\nu$ is known as the number of degrees of freedom. Figure 13.5 shows the graph of student's-t pdfs for different values of $\nu$. For $\nu \longrightarrow \infty$, the student's-t distribution tends to a Gaussian of the same mean and precision $\lambda$. Observe the heavy-tail feature of student's-t pdf, especially for low

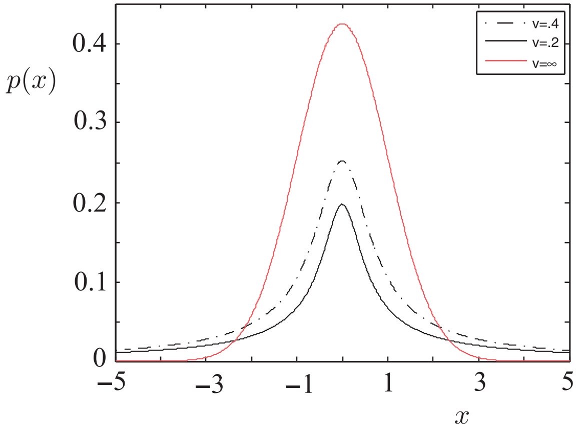


**FIGURE 13.5**

Observe that for low values of the degrees of freedom, $\nu$, student's-t pdf has very high tails. In contrast, the Gaussian pdf is a low-tailed pdf.

values of $\nu$. Recall that in our case, where we have used uninformative hyperpriors, the hyperparameter, $a$, was given a small value. Thus, our treatment in this section favors sparse solutions for the regression model. It will push as many of the coefficients, $\theta_k$, as possible toward zero. That is, it prunes the less relevant basis functions, $\phi_k(\boldsymbol{x})$, by setting the corresponding coefficients to zero. This is also the reason for using different hyperparameters, $\alpha_k$, for each one of the parameters, $\theta_k, k = 0, 2, \dots, K-1$, which provide more freedom to the learning procedure to adjust each one of the parameters individually. In the earlier days, this approach was coined *Automatic Relevance Determination* (ARD) [41, 46, 48]. An interesting discussion relating adaptive regularization and pruning is provided in [20].

Figure 13.6a provides a clear demonstration of the sparsity-imposing properties of the student's-t distribution. In the two-dimensional space, and as we move away from zero, probability mass is skewed toward the coordinate axes; that is, the pdf peaks around sparse solutions and *sparsity is now enforced probabilistically*. In contrast, the Gaussian does not give much chance to large values; see Figure 13.6b.

## 13.6 SPARSE BAYESIAN LEARNING (SBL)

In Section 13.3, the prior for each one of the unknown parameters, $\theta_k,\ k = 0, 1, \dots, K-1$, were given the liberty to have their own variances, $\sigma_k^2 := \frac{1}{\alpha_k}$. In turn, these variances were treated as hidden random variables and a prior was assigned to each of them in terms of a number of hyperparameters.

In [75, 81], the model was slightly modified. The concept of using different variances for the priors was retained, but the variances were treated as deterministic parameters and not as random ones.[2] In this context, the task becomes a generalization of the one treated in Section 12.6, and it is built upon the following assumptions:

$$p(\boldsymbol{y}|\boldsymbol{\theta};\beta) = \mathcal{N}(\boldsymbol{y}|\Phi\boldsymbol{\theta}, \beta^{-1}I), \tag{13.53}$$

$$p(\boldsymbol{\theta};\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{0}, A^{-1}), \tag{13.54}$$

where

$$A := \operatorname{diag}\{\alpha_0, \dots, \alpha_{K-1}\}. \tag{13.55}$$

The precision $\beta$ is also treated as an unknown deterministic parameter. Our goal is (a) to obtain estimates for $\beta$ and $\alpha_k,\ k = 0, 1, \dots, K-1$, and (b) to compute the predictive distribution, $p(y|\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{y}$ is the vector of observations.[3] To this end, one could adopt the EM algorithm and follow similar steps as in Section 12.6; the only difference is that there, a common variance was shared by all the involved prior pdfs. The method is usually referred to as *sparse Bayesian learning* (SBL) and complies with the ARD rationale discussed in Section 13.5.

In this section, we will adopt a different path, exploiting the Gaussian nature of the involved pdfs. A Type II maximum likelihood method will be employed, which was introduced in Remarks 12.2. Type II likelihood is defined as the marginal one, after integrating out the parameters, $\boldsymbol{\theta}$. Following the discussion in Section 12.2 and for our current needs, Eq. (12.15) is written as

$$p(\boldsymbol{y};\boldsymbol{\alpha},\beta) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{0}, \beta^{-1}I + \Phi A^{-1}\Phi^T). \tag{13.56}$$

[2] A slightly different yet equivalent view, employing uniform priors and using the respective modes instead of marginalizing out the variances, is followed in [75].

[3] Notational dependence on the input training data, $\mathcal{X}$, has been suppressed.

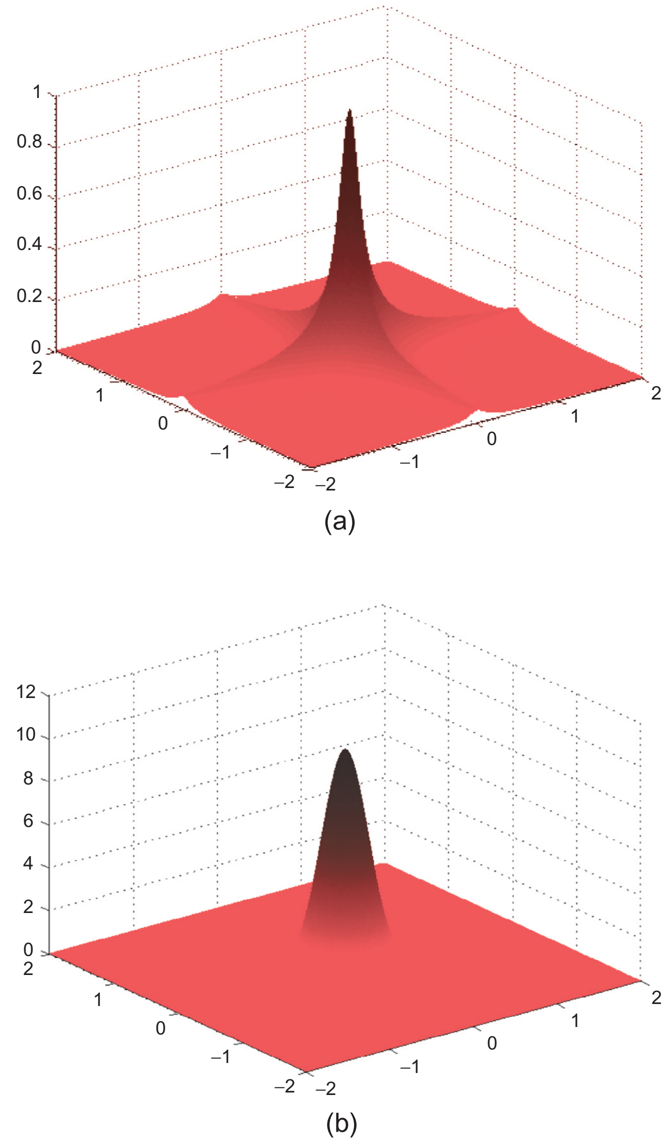


**FIGURE 13.6**

(a) The student's-t peaks sharply around zero and falls slowly along the axes; hence, sparse solutions are favored. (b) The Gaussian peaks around zero and decays very fast, along all directions.

Also, for the sake of completeness, Eqs. (12.16), (12.17) and (12.10) take the form

$$p(\boldsymbol{\theta}|\boldsymbol{y};\boldsymbol{\alpha},\beta) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu},\Sigma;\boldsymbol{\alpha},\beta), \tag{13.57}$$

with

$$\boldsymbol{\mu} = \beta\Sigma\Phi^T\boldsymbol{y}, \quad \Sigma = \left(A + \beta\Phi^T\Phi\right)^{-1}. \tag{13.58}$$

The objective now becomes to maximize with respect to $\alpha_k$, $k = 0, \ldots, K-1$, and $\beta$ the cost function,

$$\begin{aligned} L(\boldsymbol{\alpha},\beta) :&= \ln p(\boldsymbol{y};\boldsymbol{\alpha},\beta) \\ &= -\frac{N}{2}\ln(2\pi) - \frac{1}{2}|\beta^{-1}I + \Phi A^{-1}\Phi^T| \\ &\quad - \frac{1}{2}\boldsymbol{y}^T\left(\beta^{-1}I + \Phi A^{-1}\Phi^T\right)^{-1}\boldsymbol{y}. \end{aligned} \tag{13.59}$$

Maximizing the above cost cannot be carried out analytically, and the following iterative scheme is derived (Problem 13.11, the proof is a bit tedious):

$$\gamma_k = 1 - \alpha_k^{(\text{old})}\Sigma_{kk}^{(\text{old})}, \tag{13.60}$$

$$\alpha_k^{(\text{new})} = \frac{\gamma_k}{(\mu_k^{(\text{old})})^2}, \quad k = 0, 1, \ldots, K-1, \tag{13.61}$$

$$\beta^{(\text{new})} = \frac{N - \sum_{k=0}^{K-1}\gamma_k}{||\boldsymbol{y} - \Phi\boldsymbol{\mu}^{(\text{new})}||^2}. \tag{13.62}$$

The iterative scheme is initialized by an arbitrary set of values and it is repeated until a convergence criterion is met. $\Sigma_{kk}$ is the respective diagonal element of the matrix $\Sigma$. Note that both $\Sigma$ and $\boldsymbol{\mu}$ depend on the values of $\beta$ and $\alpha_k$. The main complexity per iteration step is due to the matrix inversion involved in the respective definition in (13.58), which amounts to $O(K^3)$ operations. Moreover, because a matrix inversion is involved, one must take care of near singularities, due to numerical errors. This can be the case in practice, because some of the values of $\alpha_k$ may become very large. Thus, care must be taken so that once such values occur, one removes the corresponding columns in $\Phi$ and sets the respective values of $\theta_k$ to zero. As a matter of fact, this is how sparsity is enforced by the method. Parameters with mean value equal to zero and a variance that becomes very small (precision very large) are set to zero. This behavior has empirically been observed in practice.

The alternative path to deal with the method is via the EM algorithm. This leads to an equivalent set of recursions [75], but practical experience has shown that the previously given set of updates converge faster.

Extensions of the SBL framework in the context of block sparsity and for the case of multiple measurement vectors (MMV), when elements in each nonzero row of the solution matrix are temporally correlated, are reported in [85, 86]. Moreover, in the latter one, a theoretical analysis is provided, which shows that the SBL cost function has the very desirable property that its global minimum coincides with the sparsest solution to the MMV problem.

**Example 13.2.** The goal of this example is to demonstrate the comparative performance, via a simulation example, of (a) the variational Bayesian method, (b) the maximum likelihood/LS (Eq. (12.6)), and (c) the EM algorithm of Section 12.6 in the context of linear regression and in particular in the sparse modeling framework. The SBL method gave results very similar to the variational approach, and it is not discussed any further. To this end, we generated the training data according to the following scenario.

The interval in the real axis $[-10, \ 10]$ was sampled at $N = 100$ equidistant points, $x_n$, $n = 1, 2, \ldots, 100$. The training data comprise the pairs $(y_n, x_n)$, $n = 1, 2, \ldots, N$, where

$$y_n = \exp\left(-\frac{1}{2}\frac{(x_n + 5.8)^2}{0.1}\right) + \exp\left(-\frac{1}{2}\frac{(x_n - 2.6)^2}{0.1}\right) + \eta_n,$$

where $\eta_n$ are i.i.d. zero mean Gaussian noise samples, of variance $\sigma_\eta^2 = 0.015$. To fit the data, the following model was adopted,

$$y = \sum_{k=1}^{N} \theta_k \exp\left(-\frac{1}{2}\frac{(x - x_k)^2}{0.1}\right).$$

Thus, the matrix $\Phi$ has the following elements:

$$[\Phi]_{nk} = \exp\left(-\frac{1}{2}\frac{(x_n - x_k)^2}{0.1}\right), \quad n = 1, 2, \ldots, N, \ k = 1, 2, \ldots, N.$$

Note that we have as many parameters as the number of training points. This is in line with the relevance vector machine rationale, which will be discussed in Section 13.7. Figure 13.7 illustrates the results. The red full-line curve corresponds to the true function that generates the data. The gray full-curve corresponds to the model, having plugged in as estimated values $\hat{\theta}_k$ the respective posterior mean values from Eq. (13.33). The dotted red curve corresponds to the ML solution and the dotted gray curve to the EM, where the estimates correspond to the mean of the respective posterior (Eq. (12.71)). The performance advantages of the variational approach are obvious, which almost coincide with the true one. Observe how the variational Bayesian approach managed to cope with the overfitting and pushed most of the parameters to zero values.

### 13.6.1 THE SPIKE AND SLAB METHOD

This is an old technique for imposing sparsity [37, 44]. Let us consider our familiar regression model,

$$\mathrm{y} = \theta^T \boldsymbol{\phi}(\boldsymbol{x}) + \eta = \sum_{k=0}^{K-1} \theta_k \phi_k(\boldsymbol{x}) + \eta. \tag{13.63}$$

A new set of auxiliary binary *indicator* variables are introduced, $\mathrm{s}_k \in \{0, 1\}$, $k = 0, 1, \ldots, K-1$. Let also the prior imposed on $\theta$ be a Gaussian, $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \sigma^2 I)$. As the name suggests, the indicator variables control the presence or not of a parameter in the summation in Eq. (13.63). For example, if $s_k = 1$ the corresponding parameter, $\theta_k$, is present and if $s_k = 0$ then $\theta_k$ is removed; this is the way sparsity is imposed onto the model. To this end, a joint Bernoulli prior distribution (Chapter 2) is adopted for the indicator variables, to push as many of them as possible to zero, that is,

$$P(\boldsymbol{s}) = \prod_{k=0}^{K-1} p^{s_k}(1-p)^{1-s_k}, \tag{13.64}$$

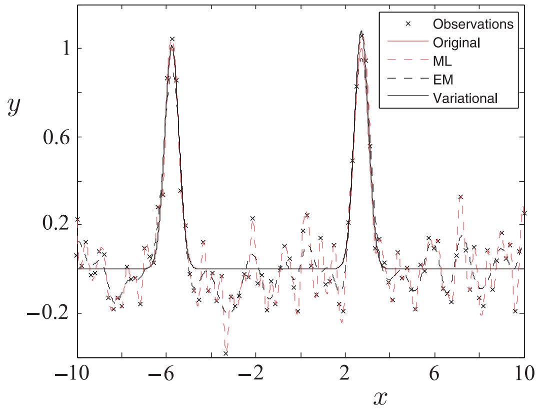


**FIGURE 13.7**

The figure corresponds to the setup of Example 13.2. Observe that the fitting curve obtained via the variational method is almost identical to the true one.

where the parameter $0 \leq p \leq 1$ specifies a prior level of sparsity. This turns out to be equivalent with adopting the following prior on the parameters:

$$\boxed{p(\boldsymbol{\theta}) = \prod_{k=0}^{K-1} \Big( s_k \mathcal{N}(\theta_k | 0, \sigma^2) + (1 - s_k)\delta(\theta_k) \Big) : \quad \text{Spike and Slab Prior.}} \tag{13.65}$$

The latter is known as the *spike and slab* prior. The name comes from the fact that if $s_k = 0$ then a "spike" is impose at the zero and the values $s_k = 1$ imposes a "slab," because a Gaussian is a broad one (for large enough $\sigma^2$). The corresponding posterior is not Gaussian and its computation can be done by mobilizing approximate inference techniques, such as variational or Monte Carlo (see e.g., [25] and the references therein).

Variants of the basic spike and slab scheme do also exist (see e.g., [70]). In the latter reference, it is shown that one can obtain the classical $\ell_0$-based sparsity-enforcing constraint on the LS criterion (Chapter 9) as a limiting case of one of these variants. Such a path provides another connection between probabilistic and optimization-based techniques for sparsity. Another connection will be discussed in Section 13.10.

## 13.7 THE RELEVANCE VECTOR MACHINE FRAMEWORK

An important aspect of the work in [75] was the introduction of *Relevance Vector Machines* for regression as well as for classification. Inspired by the support vector regression (SVR) which was discussed in Chapter 11, a specific regression model was considered, that is,

$$\mathrm{y}(\mathbf{x}) = \theta_0 + \sum_{k=1}^{N} \theta_k \kappa(\mathbf{x}, \boldsymbol{x_k}) + \eta. \tag{13.66}$$

In other words, the general regression model of Eq. (12.1) is considered for $K = N + 1$, where $N$ is the number of observations and

$$\phi_k(\mathbf{x}) = \kappa(\mathbf{x}, \boldsymbol{x}_k),$$

where $\kappa(\cdot,\cdot)$ is a kernel function, as defined in Chapter 11, centered at the input observation points, $\boldsymbol{x}_k, k = 1, 2, \ldots, N$. Thus, the number of parameters becomes equal (plus one) to the number of training points.

The task can be treated either via the SBL philosophy or via the employment of the variational approximation rationale, in order to impose sparsity. In [75], it is pointed out that the variational Bayesian approach is computationally more intensive and in practice it results to mean values for the hyperparameters, which are identical to the values obtained by using the sparse Bayesian learning (SBL) approach.

Inspired by the definition of the support vectors in the support vector regression (SVR), the surviving data points that contribute to Eq. (13.66) are called *relevance vectors*. Also, the kernels to be used in the RVM framework need not be symmetric positive definite functions, because the modeling is not necessarily associated to a reproducing kernel Hilbert space (RKHS).

### 13.7.1 ADOPTING THE LOGISTIC REGRESSION MODEL FOR CLASSIFICATION

Besides the relevance vector regression, the relevance vector classification was also introduced in [75]. Recall that in the support vector machine (SVM) classification, a linear (in an RKH space) classifier was designed. The same model is also adopted for the RVM. Given the value of a measured feature vector, $\boldsymbol{x}$, classification is performed according to the sign of the discriminant function, namely

$$f(\boldsymbol{x}) := \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}) := \theta_0 + \sum_{k=1}^{N} \theta_k \phi_k(\boldsymbol{x}).$$

The goal is to obtain an estimate of the parameters $\boldsymbol{\theta}$ in the Bayesian framework; thus, somehow, we have to "embed" $\boldsymbol{\theta}$ into a pdf that relates the input-output data. In this vein, a well-known and widely used technique is the *logistic regression* model, which was introduced in Section 7.6.

According to this model and for a two-class $(\omega_1, \omega_2)$ classification task, the posterior probabilities, as required by the Bayesian classifier, are modeled as

$$\boxed{P(\omega_1|\boldsymbol{x}) = \frac{1}{1 + \exp\left(-\boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x})\right)} : \quad \text{Logistic Regression Model,}} \tag{13.67}$$

and

$$P(\omega_2|\boldsymbol{x}) = 1 - P(\omega_1|\boldsymbol{x}). \tag{13.68}$$

There is more than one reason that justifies such a choice (see, e.g., [40] and Problem 13.12). Multiclass generalizations are also possible (e.g., [74], and Chapter 7).

For the sake of the less familiar reader, let us look at Eq. (13.67) more closely. The graph of the function

$$\sigma(t) = \frac{1}{1 + \exp(-t)}, \tag{13.69}$$

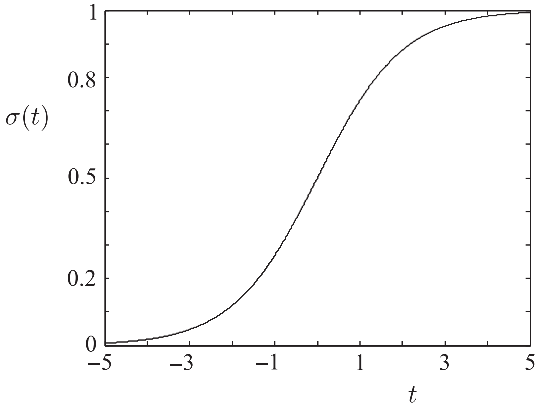


**FIGURE 13.8**

The logistic sigmoid function.

known as the *logistic sigmoid* function, is shown in Figure 13.8. For $t > 0$ $(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}) > 0)$, $P(\omega_1|\boldsymbol{x}) > \frac{1}{2}$ and the decision is in favor of $\omega_1$. The opposite holds true for $t < 0$, $(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x})) < 0)$. Considering the training set $(y_n, \boldsymbol{x}_n), \boldsymbol{x}_n \in \mathbb{R}^l$, and $y_n \in \{0, 1\}$, and adopting a Bernoulli distribution for $P(y|\boldsymbol{x})$, the respective likelihood function can be defined as

$$P(\boldsymbol{y}|\boldsymbol{\theta}) = \prod_{n=1}^{N} \left(\sigma\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)\right)^{y_n}\left(1 - \sigma\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)\right)^{1-y_n}, \tag{13.70}$$

which is the counterpart of Eq. (13.21) for the regression case. We also adopt a Gaussian prior for θ, as in Eqs. (13.54) and (13.55). As in the SBL approach, our goal is to maximize the Type II log-likelihood with respect to the unknown parameters, $\boldsymbol{\alpha}$. However, $p(\boldsymbol{y}|\boldsymbol{\theta})$ is no longer Gaussian, and marginalizing out θ cannot be carried out analytically. In [75], the Laplacian approximation is employed and the following stepwise procedure is adopted:

1. Assuming $\boldsymbol{\alpha}$ to be currently available, maximize with respect to $\boldsymbol{\theta}$ the posterior, which by simple arguments is easily shown to be

$$p(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{\alpha}) = \frac{P(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})}{P(\boldsymbol{y}|\boldsymbol{\alpha})},$$

or equivalently,

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \ln\left(P(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})\right)$$

$$= \arg\max_{\boldsymbol{\theta}} \left\{ \sum_{n=1}^{N} \left[y_n \ln \sigma\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right) + \right.\right. \tag{13.71}$$

$$\left. (1 - y_n) \ln\left(1 - \sigma\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)\right)\right] -$$

$$\left. \frac{1}{2}\boldsymbol{\theta}^T A\boldsymbol{\theta} + \text{constant}\right\}, \tag{13.72}$$

where $A := \text{diag}\{\alpha_0, \alpha_2, \dots, \alpha_N\}$. Maximizing Eq. (13.72) with respect to $\boldsymbol{\theta}$ results in (Problem 13.13),

$$\boxed{\hat{\boldsymbol{\theta}}_{\text{MAP}} = A^{-1}\Phi^T(\boldsymbol{y} - \boldsymbol{s}),} \tag{13.73}$$

where $\boldsymbol{s} := [s_1, \dots, s_N]^T$ and $s_n := \sigma\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right),\ n = 1, 2, \dots, N$.

2. Use $\hat{\boldsymbol{\theta}}_{\text{MAP}}$ and the Laplace approximation method (Section 12.3) to approximate $p(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{\alpha})$ by a Gaussian centered at $\hat{\boldsymbol{\theta}}_{\text{MAP}}$ [39]. Recall from Section 12.3 that the covariance matrix of the approximate Gaussian is given by

$$\Sigma^{-1} = -\frac{\partial^2 \ln\left(P(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\alpha)\right)}{\partial\boldsymbol{\theta}^2}\bigg|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{\text{MAP}}},$$

or (Problem 13.14)

$$\boxed{\Sigma^{-1} = (\Phi^T T \Phi + A),} \tag{13.74}$$

where $T = \text{diag}\{t_1, t_2, \dots, t_N\}$ and

$$t_n = \sigma\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)\left(1 - \sigma\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)\right)\bigg|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{\text{MAP}}}$$

3. Having obtained $\hat{\boldsymbol{\theta}}_{\text{MAP}}$ and computed $\Sigma$, then adapting Eq. (12.37) to our current notation we obtain

$$P(\boldsymbol{y}|\boldsymbol{\alpha}) = P(\boldsymbol{y}|\hat{\boldsymbol{\theta}}_{\text{MAP}})p(\hat{\boldsymbol{\theta}}_{\text{MAP}}|\boldsymbol{\alpha})(2\pi)^{\frac{N}{2}}|\Sigma|^{1/2}. \tag{13.75}$$

Next, maximization of Eq. (13.75) with respect to $\boldsymbol{\alpha}$ provides the updated iteration estimate. Note that the first term of the product on the right-hand side is independent of $\boldsymbol{\alpha}$. Taking the logarithm and maximizing easily results in (Problem 13.15)

$$-\frac{1}{2}\theta^2_{\text{MAP},k} + \frac{1}{2\alpha_k} - \frac{1}{2}\Sigma_{kk} = 0. \tag{13.76}$$

Because $\Sigma_{kk}$ as well as $\theta_{\text{MAP},k}$ depend on $\boldsymbol{\alpha}$, the equation is solved iteratively and results in exactly the same scheme as in Eq. (13.61), that is,

$$\alpha_k^{(\text{new})} = \frac{1 - \alpha_k^{(\text{old})}\Sigma_{kk}^{(\text{old})}}{\left(\theta^{(\text{old})}_{\text{MAP},k}\right)^2}.$$

The procedure continues until a convergence criterion is met [38, 75].

As pointed out in [75], although in general the Laplacian local approximation to a Gaussian may not be a good one, in the case of the current classification task, due to the specific nature of the adopted models, the approximation is expected to provide good accuracy.

Figure 13.9 shows the decision curve that results from the RVM method[4] and classifies the points of the red/gray classes. The data set is the same as the one used in Example 11.4 of Chapter 11

[4] The software used was that in http://www.miketipping.com/sparsebayes.htm#software

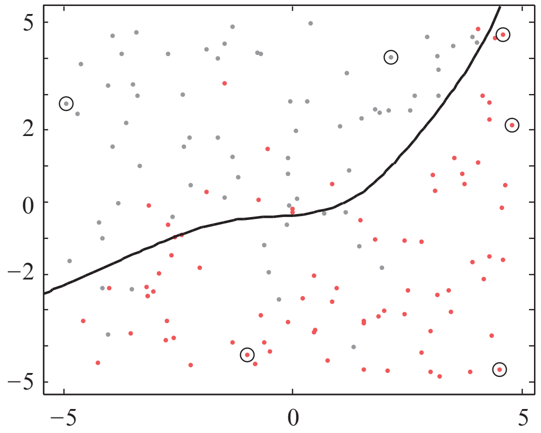


**FIGURE 13.9**

The decision curve that separates the two classes (red vs. gray), which is obtained by the RVM classifier, corresponds to posterior probability values $P(\omega_1|\boldsymbol{x}) = 0.5$. The Gaussian kernel was used with $\sigma^2 = 3$. Only six relevance vectors survive—the ones that have been circled.

when dealing with the SVM classifier. Six points, which have been circled, are the surviving relevance vectors. The Gaussian kernel was used with $\sigma^2 = 3$, which was found to give the best results. Observe that the number of support vectors surviving is significantly less compared to the case of SVM of Chapter 11.

*Remarks 13.2.*

- Compared to SVM (SVR), the RVM machinery presents advantages and disadvantages. The SVM approach has the mathematically elegant property of, theoretically, giving a single minimum due to the convexity of the associated cost functions. This is not the case for the RVM framework, where the involved optimization steps refer to a nonconvex cost. It must be kept in mind that solving a nonconvex task, one may have to run the optimization algorithm a number of times, starting each time from different initial conditions, because a nonconvex problem can be trapped in a local minimum.

  Concerning complexity, the algorithmic steps for the RVM involve the inversion of the Hessian matrix, which amounts to $O(N^3)$ complexity. As discussed in Section 11.11, the complexity range of the efficient schemes for solving the SVM scales from linear to (approximately) quadratic. Also, the memory for the RVM exhibits an $O(N^2)$ on the size of the training set as opposed to a linear dependence to the SVM case. Besides complexity, inverting (big) matrices must be done with care in order to avoid numerical instabilities due to possible (near) singularity. Also, in general, RVMs need longer training times to converge, compared to SVMs for similar error rates.

  A fast RVM algorithm has been developed in [76] by analyzing the properties of the marginal likelihood. This enables a sequential addition and deletion of candidate basis functions (columns of $\Phi$) to monotonically maximize the marginal likelihood. This iterative algorithm operates in a constructive manner, until all relevant basis functions (for which the associated weights are nonzero) have been included. If $M$ denotes the number of relevant terms, the complexity amounts to $O(M^3)$, which for $M << N$ is more efficient than the original RVM.

The main advantage of RVMs is that, in general, they result in sparser solutions compared to SVMs for similar levels of generalization errors. This makes the prediction step, after the training has been completed, more efficient compared to the prediction model resulting from SVM. Moreover, SVMs suffer from their dependence on the user-dependent hyperparameter, $C$ ($\epsilon$ for regression), and they are generally found by cross-validation, which involves multiple training for different values.

- In [7], a different algorithmic approach has been adopted based on the *variational bound approximation* method, to be described next.

## 13.8 CONVEX DUALITY AND VARIATIONAL BOUNDS

In the previous chapter, the Laplacian technique for the approximation of a general pdf by a Gaussian was introduced. The driving force behind such an approximation was to benefit from the computationally friendly nature of the Gaussian pdf. In this section, we will approach this task from a different perspective, involving maximization of a lower bound of the pdf at hand with respect to an extra parameter, which is introduced into the problem and on which the lower bound depends. Our theoretical framework is that of convex duality, a well-known and powerful tool in convex analysis.

Let a function $f : \mathbb{R}^l \longmapsto \mathbb{R}$. The function

$$f^* : \mathbb{R}^l \longmapsto \mathbb{R},$$

defined as[5]

$$f^*(\boldsymbol{\xi}) = \max_{\boldsymbol{x}} \left\{ \boldsymbol{\xi}^T \boldsymbol{x} - f(\boldsymbol{x}) \right\}, \tag{13.77}$$

is called the *conjugate* of $f$. The domain of the conjugate function consists of all $\boldsymbol{\xi} \in \mathbb{R}^l$ for which the maximum is finite. A notable property of the conjugate function is its *convexity*; this is true whether or not $f$ is convex. The convexity is the outcome of the point-wise maximization of a family of (convex with respect to $\boldsymbol{\xi}$) affine functions [10].

Maximizing Eq. (13.77) with respect to $\boldsymbol{x}$ results in a value of $\boldsymbol{x}_*$, such that

$$\boldsymbol{x}_* : \nabla f(\boldsymbol{x}_*) = \boldsymbol{\xi}, \tag{13.78}$$

which leads to the value

$$f^*(\boldsymbol{\xi}) = \boldsymbol{\xi}^T \boldsymbol{x}_* - f(\boldsymbol{x}_*). \tag{13.79}$$

Equations (13.78) and (13.79) provide the geometric interpretation of the conjugate function. The graph of the linear function $\boldsymbol{\xi}^T \boldsymbol{x}$ defines a hyperplane whose direction is controlled by $\boldsymbol{\xi}$; the latter is now equal to $\nabla f(\boldsymbol{x}_*)$, which defines the direction of the tangent hyperplane of the graph of $f(\boldsymbol{x})$ at $\boldsymbol{x}_*$. This tangent hyperplane is described by

$$g(\boldsymbol{x}) = f(\boldsymbol{x}_*) + (\boldsymbol{x} - \boldsymbol{x}_*)^T \nabla f(\boldsymbol{x}_*),$$

[5] Strictly speaking, one should use the sup instead of max and inf instead of min, throughout.

or using Eq. (13.79),

$$g(\boldsymbol{x}) = \boldsymbol{\xi}^T \boldsymbol{x} - f^*(\boldsymbol{\xi}). \tag{13.80}$$

For $\boldsymbol{x} = 0$, Eq. (13.80) becomes $g(\mathbf{0}) = -f^*(\boldsymbol{\xi})$. This is illustrated in Figure 13.10. Thus, $f^*(\boldsymbol{\xi})$ corresponds to the displacement that the graph of $\boldsymbol{\xi}^T\boldsymbol{x}$ has to undergo in order to "touch" that of $f(\boldsymbol{x})$. A byproduct of all these turns out to be very useful for us. It can be shown (Problem 13.16) that if $f$ is a convex function, then $(f^*)^* = f$ and in this case we can write

$$f(\boldsymbol{x}) = \max_{\boldsymbol{\xi}} \left\{ \boldsymbol{x}^T\boldsymbol{\xi} - f^*(\boldsymbol{\xi}) \right\}. \tag{13.81}$$

Thus, once $f^*$ is computed, a lower bound for $f$ becomes readily available, that is,

$$f(\boldsymbol{x}) \geq \boldsymbol{x}^T\boldsymbol{\xi} - f^*(\boldsymbol{\xi}), \tag{13.82}$$

where now $\boldsymbol{\xi}$ is interpreted as a parameter. To investigate this bound a bit further, plug Eqs. (13.78) and (13.79) into Eq. (13.82) to obtain[6]

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}_*) + (\boldsymbol{x} - \boldsymbol{x}_*)^T \nabla f(\boldsymbol{x}_*),$$

where the right-hand side is the linear function $g(\boldsymbol{x})$ describing the hyperplane tangent to $f(\boldsymbol{x})$ at $\boldsymbol{x}_*$. The bound becomes tight at $\boldsymbol{x} = \boldsymbol{x}_*$; see Figure 13.10. We will soon see how we can make this linear function bound a nonlinear one; it suffices to transform the argument of the function.

All that has been said for convex functions applies to concave ones if we replace the max operation in Eqs. (13.77) and (13.81) with min operations. Note that following this definition, the conjugate function is *concave*, being the result of point-wise minimization of a set of concave functions (an affine function can be considered either convex or concave). Furthermore, if the involved function is neither convex nor concave, one can search for *invertible* transformations that render it convex or concave.

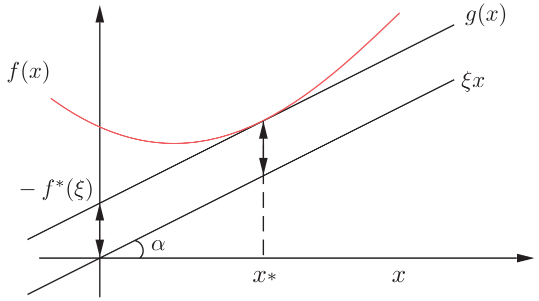


**FIGURE 13.10**

The direction of the line $y = \xi x$ that crosses the origin is controlled by $\xi$ ($\xi = \tan\alpha$). The (negative) value of the conjugate function $f^*$ at $\xi$ defines the point where the line $y = g(x)$ cuts the vertical axis; $g(x)$ is formed by translating $\xi x$ until it becomes tangent to $f$ at $x_*$.

[6] Note that this is a necessary and sufficient condition for convexity.

In our context, the purpose of resorting to the notion of the conjugate function is our expectation that such a function, which bounds a (pdf) function as in Eq. (13.82), may lead to a functional form that lends itself to tractable computations of the involved integrations.

**Example 13.3.** Compute the conjugate of the logarithmic function $f(x) = \ln x, x > 0$.

The logarithmic function is known to be concave. Hence

$$f^*(\xi) = \min_{x>0} \{\xi x - \ln x\},$$

or

$$x_* : \frac{1}{x} = \xi \Rightarrow x_* = \frac{1}{\xi}.$$

Hence,

$$f^*(\xi) = 1 + \ln \xi.$$

Therefore,

$$\ln x = \min_{\xi>0} \{\xi x - 1 - \ln \xi\}.$$

Figure 13.11 shows the respective graphs of the logarithmic function as well as the resulting *linear* function bound for different values of $\xi$.

**Example 13.4.** Consider the univariate Laplacian pdf, which we have already seen in Section 13.5,

$$p(\theta) = \frac{\lambda}{2} \exp(-\lambda|\theta|), \quad \theta \in \mathbb{R}. \tag{13.83}$$

Our goal is to derive a lower bound in terms of its conjugate function. From Eq. (13.83), we get

$$\ln p(\theta) = \ln \frac{\lambda}{2} - \lambda|\theta|.$$

Define

$$f(x) = \ln \frac{\lambda}{2} - \lambda\sqrt{x}, \quad x > 0. \tag{13.84}$$

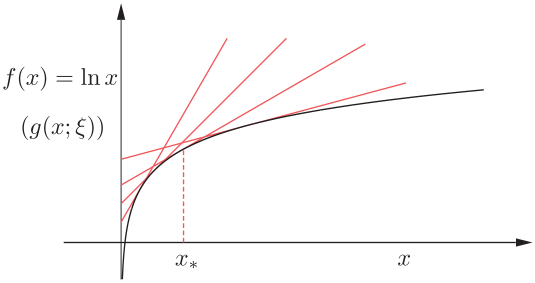


**FIGURE 13.11**

The linear functions $g(x;\xi) = \xi x - 1 - \ln \xi$ provide upper bounds to $f(x) = \ln x$. Each one of the lines is tangent to $f(x) = \ln x$ at the point $x_* = \frac{1}{\xi}$.

Then

$$\ln p(\theta) = f(\theta^2). \tag{13.85}$$

Note that $f(x)$ is a convex function with respect to $x$ (Problem 13.16). The conjugate of $f(x)$ is obtained as[7]

$$f^*(\xi) = \max_x \left\{ -\frac{\xi}{2}x - f(x) \right\}, \; \xi > 0.^{8} \tag{13.86}$$

Recalling Eq. (13.84), maximization leads to

$$x_* : \lambda x^{-\frac{1}{2}} = \xi \Rightarrow x_* = \lambda^2 \xi^{-2}. \tag{13.87}$$

Combining Eqs. (13.86) and (13.87) gives

$$f^*(\xi) = \frac{\lambda^2}{2}\xi^{-1} - \ln\frac{\lambda}{2}. \tag{13.88}$$

Hence, we obtain the bound

$$f(x) \geq -\frac{\xi}{2}x - \frac{\lambda^2}{2}\xi^{-1} + \ln\frac{\lambda}{2},$$

or

$$\ln p(\theta) \geq -\frac{\xi}{2}\theta^2 - \frac{\lambda^2}{2}\xi^{-1} + \ln\frac{\lambda}{2}, \quad \xi > 0.$$

Because this is true $\forall \xi > 0$, we can replace $\xi$ with $\xi^{-1}$, for notational convenience, which results in

$$p(\theta) \geq \frac{\lambda}{2}\exp\left(-\frac{\xi^{-1}}{2}\theta^2\right)\exp\left(-\frac{\lambda^2}{2}\xi\right), \tag{13.89}$$

which, after mobilizing the Gaussian notation and its integration property, can be rewritten as

$$p(\theta) \geq \mathcal{N}(\theta|0,\xi)\phi(\xi), \quad \xi > 0 \tag{13.90}$$

with

$$\phi(\xi) = \frac{\lambda}{2}\sqrt{2\pi\xi}\exp\left(-\frac{\lambda^2}{2}\xi\right), \quad \xi > 0.$$

This is very interesting indeed. The obtained lower bound has a functional dependence on $\theta$, which is of a Gaussian nature; the Gaussian term is centered at zero with variance $\xi$. Maximizing with respect to $\xi$, we will obtain the required approximation. Figure 13.12 shows the obtained approximation for different values of $\xi$. Observe that introducing transformations in the involved variables can render the function in the obtained bound a *nonlinear* one.

[7] Because maximization takes place for all $\xi$, we use $-\frac{\xi}{2}$. This is only for notational convenience in order to obtain the result in a convenient form.

[8] $\xi$ is constrained to positive values, because for $\xi \leq 0$ the maximum with respect to $x$ becomes infinite, which violates the definition of the conjugate functions.

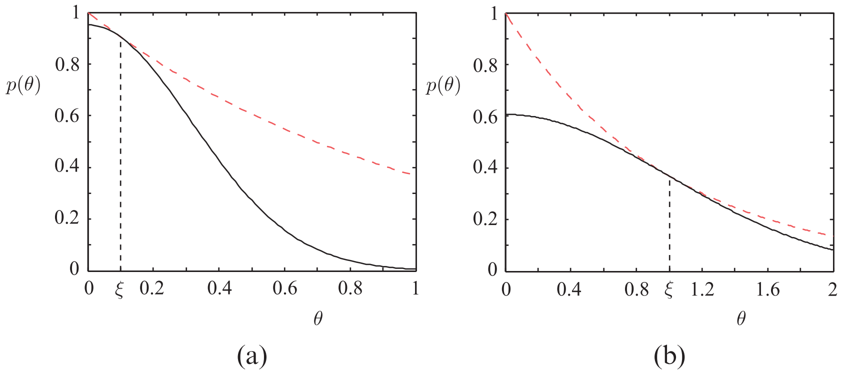


**FIGURE 13.12**

The Laplacian (red curves) and the approximating Gaussians for two different values of $\xi$.

For a multivariate Laplacian and assuming a parameter vector with independent components, it can be trivially shown that

$$p(\boldsymbol{\theta}) = \prod_{k=0}^{K-1} p(\theta_k) \geq \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \Xi) \prod_{k=0}^{K-1} \phi(\xi_k) := \hat{p}(\boldsymbol{\theta}; \boldsymbol{\xi}), \tag{13.91}$$

where $\boldsymbol{\xi} = [\xi_0, \ldots, \xi_{K-1}]^T$ and

$$\Xi := \text{diag}\{\xi_0, \xi_1, \ldots, \xi_{K-1}\}.$$

*Remarks 13.3.*

- The method of representing a convex function via the optimization of lower bound in terms of its conjugate (dual form) is known as the *variational method*, and the associated parameters $\xi_k$ as *variational* parameters [59]. Its use in the context of machine learning was first reported in [27] (see also [31]), and its use subsequently proliferated and was adopted in different scenarios.
- The method has been used to obtain variational approximations for a number of pdfs that are suitable for sparsity-aware learning, for example, Jeffreys', student's-t, generalized Gaussians (see, e.g., [52]), and for the logistic regression model [29] (Problem 13.18). Compared to the Laplacian approximation method, the variational approach provides the extra flexibility of optimizing with respect to the corresponding variational parameters (see [29] for a related discussion). The reader, however, has to keep in mind that both approximations need not always be good ones. This is readily observed from Figure 13.12. One may obtain a good approximation locally, but not everywhere. However, it turns out that in practice, in the context of Bayesian learning, a poor approximation of the prior (for which such approximations are used) does not necessarily lead to a poor approximation of the posterior. There is no guarantee of it, and it is only the performance in practice that has the final verdict. This can be considered a drawback of the Bayesian technique compared to the deterministic methods based on optimization criteria. The latter ones, by adopting usually convex cost functions, can lead to solutions that are well characterized. In contrast,

Bayesian inference techniques suffer from their nonconvex nature and the fact that very often the imposed approximations may not necessarily be good ones. However, this is always the case in life. There is no free lunch. At the time this book was written, both of these paths to machine learning still comprised viable and powerful techniques, with their pros and cons.

## 13.9 SPARSITY-AWARE REGRESSION: A VARIATIONAL BOUND BAYESIAN PATH

The goal of this section is to demonstrate the use of convex duality and the respective variational bounds in order to approximate the computation of the evidence function, in cases where the corresponding integral is intractable. We have chosen to describe the method in the framework of sparsity-aware learning; this can also help us establish bridges with Chapter 9, and some of the results will be used in Section 13.10 for this purpose.

To comply with the assumptions made in Chapter 9, and without loss of generality, let us assume that the involved data have zero mean values. If not, the training data can be centered by subtracting their respective sample means; thus, we set $\theta_0 = 0$ and assume that the number of parameters is $K$. Then, our regression model becomes

$$\mathbf{y} = \Phi\boldsymbol{\theta} + \boldsymbol{\eta}, \quad \boldsymbol{\eta}, \mathbf{y} \in \mathbb{R}^N, \ \boldsymbol{\theta} \in \mathbb{R}^K, \ K > N,$$

where we are informed about the "secret" that most of the components of $\boldsymbol{\theta}$ are (almost) zero. In Section 13.5, we commented on the inadequacy of a Gaussian prior to provide a reasonable statistical description of a sparse random vector. A heavy-tailed distribution that enjoys popularity for sparse modeling is the Laplacian one. After all, adopting a Laplacian prior to $\boldsymbol{\theta}$, that is,

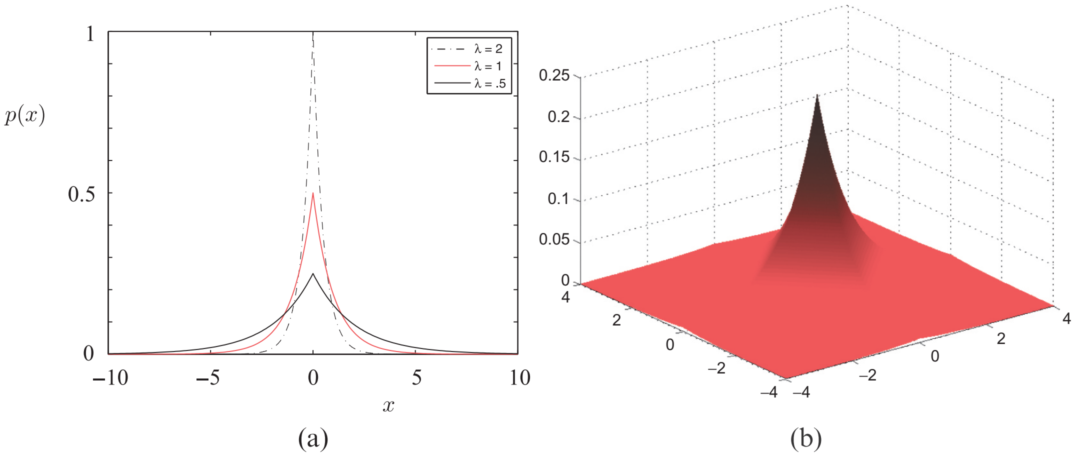
$$p(\boldsymbol{\theta}) = \prod_{k=1}^{K} p(\theta_k) = \prod_{k=1}^{K} \frac{\lambda}{2} \exp(-\lambda|\theta_k|),$$

and a Gaussian conditional pdf, $p(\boldsymbol{y}|\boldsymbol{\theta})$, for the observations $\boldsymbol{y}$ as in Eq. (13.21), makes the MAP estimation identical to our familiar LASSO task, discussed in Chapter 9. In this vein, we will build this section around the Laplacian pdf. Figure 13.13a shows the Laplacian for different values of $\lambda$. In 13.13b the two-dimensional plot is provided, from which the respect that this pdf shows to sparse solutions is readily observed.

The problem with the Laplacian pdf is that its presence in Eq. (12.14) makes the computation of the integral computationally intractable. Also, recall from Section 12.4 that the Laplacian pdf does not belong to the computationally attractive exponential family. To facilitate its treatment, we will employ the variational bound approximation method in order to approximate the Laplacian by a Gaussian, following Eqs. (13.89) and (13.91). The variational parameters will be determined by maximizing the respective evidence via the EM algorithm. This method in recovering sparse solutions was first introduced in [18] in the context of dictionary learning.

For simplicity, let the noise sequence in our regression model be white with variance $\sigma_\eta^2 := \frac{1}{\beta}$. Then

$$p(\boldsymbol{y}|\boldsymbol{\theta};\beta) = \mathcal{N}(\boldsymbol{y}|\Phi\boldsymbol{\theta}, \beta^{-1}I),$$



(a)

(b)

**FIGURE 13.13**

(a) The one-dimensional Laplacian for different values of λ. (b) A plot of a two-dimensional Laplacian pdf.

and using Eq. (13.91), we can write

$$\begin{aligned} p(\mathbf{y};\beta) &= \int \mathcal{N}(\mathbf{y}|\Phi\boldsymbol{\theta},\beta^{-1}I)p(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta} \\ &\geq \left(\int \mathcal{N}(\mathbf{y}|\Phi\boldsymbol{\theta},\beta^{-1}I)\mathcal{N}(\boldsymbol{\theta}|\mathbf{0},\Xi)\,\mathrm{d}\boldsymbol{\theta}\right)\prod_{k=1}^{K}\phi(\xi_k). \end{aligned}$$

We know by now that the integral results in a new Gaussian function (recall Eq. (12.15)), hence

$$p(\mathbf{y};\beta) \geq \mathcal{N}(\mathbf{y}|\mathbf{0},\beta^{-1}I+\Phi\Xi\Phi^T)\prod_{k=1}^{K}\phi(\xi_k) := \hat{p}(\mathbf{y};\beta,\Xi). \tag{13.92}$$

The unknown values of $\beta$ and $\Xi$ could be obtained by direct maximization of the previous bound. However, here we will adopt an EM algorithm approach, in a similar way as in Section 12.6; the difference here lies in the existence of the multiplicative terms $\phi(\xi_k)$ that differentiates the M-step. In order to employ the EM, we need to know the posterior $p(\boldsymbol{\theta}|\mathbf{y};\beta)$. We will accept the following,

$$p(\boldsymbol{\theta}|\mathbf{y};\beta) \simeq \hat{p}(\boldsymbol{\theta}|\mathbf{y};\beta,\Xi) := \frac{\mathcal{N}(\mathbf{y}|\Phi\boldsymbol{\theta},\beta^{-1}I)\mathcal{N}(\boldsymbol{\theta}|\mathbf{0},\Xi)}{\int\mathcal{N}(\mathbf{y}|\Phi\boldsymbol{\theta},\beta^{-1}I)\mathcal{N}(\boldsymbol{\theta}|\mathbf{0},\Xi)\,\mathrm{d}\boldsymbol{\theta}}, \tag{13.93}$$

where in place of $p(\boldsymbol{\theta})$ we have used its respective bound, $\hat{p}(\boldsymbol{\theta};\boldsymbol{\xi})$, from Eq. (13.91). Note that irrespective of which method one adopts to optimize with respect to the unknown parameters, the approximate posterior given in Eq. (13.93) is the quantity of interest in regression; this is used either to predict $\boldsymbol{\theta}$ or to perform predictions of the output value (Eq. (12.18)).

It must be stressed, however, that Eq. (13.93) is not a bound of $p(\boldsymbol{\theta}|\mathbf{y};\beta)$ anymore, because normalization has taken place and division does not necessarily respect bounds. Recalling what we said in Section 12.2.2 (Eqs. (12.27) and (12.28)) we obtain

$$\hat{p}(\boldsymbol{\theta}|\mathbf{y};\beta,\Xi) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_{\theta|y},\Sigma_{\theta|y}), \tag{13.94}$$

where

$$\boxed{\begin{aligned}\boldsymbol{\mu}_{\theta|y} &= \Xi\Phi^T\left(\frac{1}{\beta}I+\Phi\Xi\Phi^T\right)^{-1}\boldsymbol{y}, \qquad (13.95)\\ \Sigma_{\theta|y} &= \Xi-\Xi\Phi^T\left(\frac{1}{\beta}I+\Phi\Xi\Phi^T\right)^{-1}\Phi\Xi. \qquad (13.96)\end{aligned}}$$

We are now ready to give the algorithmic steps. Recall that in EM, the goal is to maximize the expected value of the complete log-likelihood with respect to the unknown set of deterministic parameters. In our case, our goal will be to maximize the corresponding bound with respect to $\beta$ and $\boldsymbol{\xi}$,

$$\mathbb{E}\Big[\ln p(\boldsymbol{y},\boldsymbol{\theta};\beta)\Big]=\mathbb{E}\Big[\ln\Big(p(\boldsymbol{y}|\boldsymbol{\theta};\beta)p(\boldsymbol{\theta})\Big)\Big]\geq\mathbb{E}\Big[\ln\Big(p(\boldsymbol{y}|\boldsymbol{\theta};\beta)\hat{p}(\boldsymbol{\theta};\boldsymbol{\xi})\Big)\Big]$$

Assuming $\boldsymbol{\xi}^{(0)},\beta^{(0)}$ are known, the $(j+1)$ iteration comprises the following computations:

- E-step: From Eq. (13.95), Eq. (13.96) compute

$$\boldsymbol{\mu}_{\theta|y}^{(j)} \quad \text{and} \quad \Sigma_{\theta|y}^{(j)}.$$

Following similar steps as in Section 12.6, we readily obtain that

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\xi},\beta;\boldsymbol{\xi}^{(j)},\beta^{(j)}) = {} & \frac{N}{2}\ln\beta-\frac{N}{2}\ln(2\pi)-\frac{\beta}{2}\mathbb{E}_{\theta|y}\left[\|\boldsymbol{y}-\Phi\boldsymbol{\theta}\|^2\right]\\ &+\sum_{k=1}^{K}\ln\phi(\xi_k)-\frac{K}{2}\ln(2\pi)-\frac{1}{2}\ln|\Xi|\\ &-\frac{1}{2}\sum_{k=1}^{K}\frac{\mathbb{E}_{\theta|y}\left[\theta_k^2\right]}{\xi_k},\end{aligned} \qquad (13.97)$$

where (recall Eq. (12.76))

$$\mathbb{E}_{\theta|y}\left[\|\boldsymbol{y}-\Phi\boldsymbol{\theta}\|^2\right]=\|\boldsymbol{y}-\Phi\boldsymbol{\mu}_{\theta|y}^{(j)}\|^2+\operatorname{trace}\left\{\Phi\Sigma_{\theta|y}^{(j)}\Phi^T\right\},$$

and (recall Eq. (12.74))

$$\mathbb{E}_{\theta|y}\left[\theta_k^2\right]=\left[\boldsymbol{\mu}_{\theta|y}^{(j)}\boldsymbol{\mu}_{\theta|y}^{(j)T}+\Sigma_{\theta|y}^{(j)}\right]_{kk}.$$

- M-Step: Taking the derivative of $\mathcal{Q}(\boldsymbol{\xi},\beta;\boldsymbol{\xi}^{(j)},\beta^{(j)})$ with respect to $\beta$ and equating to 0 we get

$$\boxed{\beta^{(j+1)}=\frac{N}{\|\boldsymbol{y}-\Phi\boldsymbol{\mu}_{\theta|y}^{(j)}\|^2+\operatorname{trace}\left\{\Phi\Sigma_{\theta|y}^{(j)}\Phi^T\right\}}.} \qquad (13.98)$$

The derivation with respect to $\xi_k, k=1,2,\ldots,K$, results in (Problem 13.19)

$$\boxed{\xi_k^{(j+1)}=\sqrt{\frac{\mathbb{E}_{\theta|y}\left[\theta_k^2\right]}{\lambda^2}},} \qquad (13.99)$$

which completes the loop. Iterations continue until a termination criterion is met.

An alternative viewpoint that justifies the maximization of the bound of the evidence with respect to the variational parameters is the following (see also [84] for a related discussion): At each iteration step, the EM algorithm maximizes $\mathbb{E}\left[p(\boldsymbol{y}|\boldsymbol{\theta};\beta)\hat{p}(\boldsymbol{\theta};\boldsymbol{\xi})\right]$ due to the monotonicity of the logarithmic function. Equivalently, this can be seen as the following minimization task,

$$\boldsymbol{\xi} = \arg\min_{\boldsymbol{\xi}} \mathbb{E}\left[p(\boldsymbol{y}|\boldsymbol{\theta};\beta)|p(\boldsymbol{\theta}) - \hat{p}(\boldsymbol{\theta};\boldsymbol{\xi})|\right], \tag{13.100}$$

where the lower bound property (Eq. (13.91)) has been used in order to involve the absolute value. Looking at Eq. (13.100), one may think of a reason that justifies what in practice is commonly observed; that is, the method results in good performance although the overall approximation of the prior may not be a good one. The important issue is to have a good approximation in values of $\boldsymbol{\theta}$ that correspond to relatively large values of $p(\boldsymbol{y}|\boldsymbol{\theta})$. The approximation in ranges of $\boldsymbol{\theta}$ where $p(\boldsymbol{y}|\boldsymbol{\theta}) \approx 0$ does not affect the main goal of the task. Moreover, Eq. (13.100) could also provide a justification of the relative advantage of the variational approximation method compared to the Laplacian method; in the latter, there is no room left to leverage any extra parameters in order to improve the final goal.

*Remarks 13.4.*

- As we have already commented in Chapter 9, sparsity-aware learning has been a field of intense research. No doubt this is also the case for the Bayesian approach to sparsity-promoting models. So far, we presented a hierarchical approach in Section 13.5, where sparsity was indirectly imposed by associating a gamma pdf prior on each one of the precision variables individually; this led to an equivalent high-tail student's-t pdf description of the involved parameters $\boldsymbol{\theta}$. In the current section, a Laplacian prior was imposed on $\boldsymbol{\theta}$ in order to promote sparseness. These are not the only possibilities. We focused on them in order to demonstrate two possible paths to treat the evidence maximization whenever the resulting integral is computationally "awkward."
- In [15], sparsity in the Bayesian framework is attacked by imposing a Gaussian prior on the parameters, treating variance as latent variables with an exponential prior. Such modeling is equivalent to a Laplacian pdf, once variances are integrated out. The EM procedure is then used to compute the required estimates. Also in this paper, the use of Jeffreys' prior ($p(x) \simeq \frac{1}{x}$) is proposed as an alternative to the Laplacian one.
- In [4], sparsity is imposed in a similar way as before, but in the hierarchical model, the parameter controlling the exponential prior of the precisions is also treated as a latent variable with a Jeffreys' prior. In [5], sparsity on the unknown parameters was imposed via a generalized Gaussian pdf, that is,

$$p(\boldsymbol{\theta}|\alpha) \propto \exp\left(-\lambda \sum_{k=1}^{K} |\theta_k|^p\right). \tag{13.101}$$

  Combining this prior with a Gaussian pdf for the conditional, $p(\boldsymbol{y}|\boldsymbol{\theta})$ in Eq. (13.21), would result in a MAP that corresponds to the LS regularized by a nonconvex $\ell_p$, $p < 1$ norm; we know that such norms are more aggressive, compared to the $\ell_1$ norm, in recovering sparse solutions. For $p = 1$, Eq. (13.101) becomes the Laplacian prior. In [5], gamma priors are used in association with the hyperparameter $\alpha$ and the noise variance, and the variational Bayesian approach is used to obtain the solution.

- In [30], the RVM framework is exploited to obtain sparse solutions and the information related to the variance of the obtained estimates is used to determine the number of measurements, which is sufficient for recovering the solution in the framework of compressed sensing.

## 13.10 SPARSITY-AWARE LEARNING: SOME CONCLUDING REMARKS

The goal of this section is to establish bridges among the different aspects of sparsity-aware learning that have been addressed in various parts of this book. In Chapter 9, it was treated as an optimization of a regularized cost function. In Section 13.3, the automatic relevance determination (ARD) concept in Bayesian learning was discussed, and in Section 13.9, the variational bound technique was exploited to overcome the computational obstacle associated with the Laplacian prior.

Let us recall our basic regression model (assuming centered data) and $K$ unknown parameters,

$$\mathbf{y} = \Phi\boldsymbol{\theta} + \boldsymbol{\eta}, \quad \mathbf{y}, \boldsymbol{\eta} \in \mathbb{R}^N, \; \boldsymbol{\theta} \in \mathbb{R}^K, \tag{13.102}$$

where the interest in the theory for sparse modeling is for $K > N$, and $\boldsymbol{\theta}$ is assumed to have most of its components (approximately) equal to zero, and $\Phi \in \mathbb{R}^{N \times K}$ is a fixed predetermined matrix ($\boldsymbol{\theta}$ is replaced by $\theta$ if it is assumed to be treated as a random vector). Viewed differently, this can be seen as a task of expanding a signal vector in terms of the columns of an overcomplete dictionary in the presence of noise (Chapter 9).

There are two questions that are naturally posed. The first one concerns the relationship between the Bayesian and the regularized cost function optimization approaches. How different are they? Are there any paths that establish connections between them? The second question addresses theoretical issues associated with the performance of the Bayesian techniques; recall that in our Bayesian treatment of the sparse modeling task, no such theorems were mentioned. In contrast, Chapter 9 was full of theorems concerning conditions, properties, and performance bounds related to sparse solutions. Needless to say that all these theorems were proved under assumptions on the nature of the data matrix $\Phi$. A first systematic attempt to address both questions was made in [80, 82, 83]. Our goal is to highlight some of the findings in this innovative reported research.

The cost function for the regularized regression in a more general setting is written as

$$\boxed{L_I(\boldsymbol{\theta}, \lambda) = ||\mathbf{y} - \Phi\boldsymbol{\theta}||^2 + \lambda \sum_{k=1}^{K} g(\theta_k),} \tag{13.103}$$

and the resulting estimate is given by

$$\hat{\boldsymbol{\theta}}_I = \arg\min_{\boldsymbol{\theta}} L_I(\boldsymbol{\theta}, \lambda). \tag{13.104}$$

This formulation covers all the regularizers discussed in Chapter 9. For example, if $g(\theta_k) = |\theta_k|$, the $\ell_1$ norm results. Also, we have already pointed out that Eq. (13.104) can be seen as the MAP estimator, assuming Gaussian noise and a prior of the form

$$p(\boldsymbol{\theta}) \propto \exp\left(-\frac{1}{2}\sum_{k=1}^{K} g(\theta_k)\right). \tag{13.105}$$

Hence, from now on, we will refer to the regularized regression approach as Type I estimator (see Remarks 12.1). We know that in order to promote sparsity, the function $g$ must be of a certain form; for example, it has been established in Chapter 9 that $g(x) = x^2$ is not a good choice. Functions that promote sparsity can be obtained by using

$$g(x) = h(x^2),$$

where $h(x)$ is *concave* and *nondecreasing* in $[0, \infty)$. As a matter of fact, the more concave $h$ is, the more aggressive toward sparsity the optimization becomes (e.g., [52]). A number of densities of the form of Eq. (13.105) satisfy this criterion, such as generalized Gaussians ($\exp(-|x|^p)$, $0 < p \leq 2$), student's-t ($(1 + x^2/\nu)^{-\frac{\nu+1}{2}}$), logistic ($1/\cosh^2(\frac{x}{2})$), and Laplace. An example is $h(x) = \sqrt{x}$ (which is concave) and results in our familiar $g(x) = |x|$. In the context of the regularized cost functions in Chapter 9, we have also discussed other forms, for example, $g(x) = \ln(|x| + \epsilon)$.

Let us now return to the Bayesian alternative path, which we will refer to as Type II (Remarks 12.2). For notational compliance with Eq. (13.103), assume

$$p(\boldsymbol{y}|\boldsymbol{\theta}) \propto \exp\left(-\frac{1}{2\lambda}||\boldsymbol{y} - \Phi\boldsymbol{\theta}||^2\right)$$

and $p(\boldsymbol{\theta})$ given as in Eq. (13.105). Following similar arguments as in Example 13.4 and setting $h(x)$ in place of $\sqrt{x}$, we can readily obtain that

$$p(\theta_k) \geq \mathcal{N}(\theta_k|0, \xi_k)\phi(\xi_k), \quad \xi_k > 0, \tag{13.106}$$

where now $\phi(\xi_k)$ depends on the dual $f^*(\xi)$ of

$$f(x) = C - \frac{1}{2}h(x),$$

and $C$ is a proportionality constant. Therefore, as in Section 13.9, the starting point involves the approximation of the unnormalized prior

$$\hat{p}(\boldsymbol{\theta}; \Xi) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{0}, \Xi)\prod_{k=1}^{K}\phi(\xi_k),$$

as well as the normalized posterior

$$\hat{p}(\boldsymbol{\theta}|\boldsymbol{y}; \lambda, \Xi) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_{\theta|y}, \Sigma_{\theta|y}).$$

Using Eq. (13.95) we get

$$\boldsymbol{\mu}_{\theta|y} = \Xi\Phi^T(\lambda I + \Phi\Xi\Phi^T)^{-1}\boldsymbol{y}, \tag{13.107}$$

where $\lambda$ has replaced $\beta^{-1}$, and as in Eq. (13.96),

$$\Sigma_{\theta|y} = \Xi - \Xi\Phi^T\left(\lambda I + \Phi\Xi\Phi^T\right)^{-1}\Phi\Xi.$$

From Eq. (13.107), it is readily seen that if $\boldsymbol{\xi}$ is sparse, then $\boldsymbol{\mu}_{\theta|y}$ is also sparse. Also, recall from Section 13.9 that the bound on the evidence, after marginalizing out $\boldsymbol{\theta}$, is given by (13.92),

$$\hat{p}(\boldsymbol{y};\lambda,\Xi)=\mathcal{N}(\boldsymbol{y}|\boldsymbol{0},\lambda I+\Phi\Xi\Phi^T)\prod_{k=1}^{K}\phi(\xi_k). \tag{13.108}$$

Note that if one assumes $\phi(\xi_k)=1,\ k=1,2,\ldots,K$, then the task becomes identical with the ARD approach, which was discussed in Section 13.3; in other words, it is the evidence if a Gaussian prior is used for each one of the parameters, $\theta_k$, with a different variance parameter $(\xi_k)$ assigned to each one of them.

The variational parameters $\boldsymbol{\xi}$ (as well as $\lambda$) are obtained by maximizing the bound in Eq. (13.108). In Section 13.9, the EM algorithm was adopted. Here, we will act directly on Eq. (13.108) and our goal will be to minimize the negative logarithm of the bound; that is, the cost function will be

$$L_{II}(\boldsymbol{\xi};\lambda)=-2\ln\mathcal{N}(\boldsymbol{y}|\boldsymbol{0},\lambda I+\Phi\Xi\Phi^T)-2\sum_{k=1}^{K}\ln\phi(\xi_k)$$

or

$$\boxed{L_{II}(\boldsymbol{\xi};\lambda)=\boldsymbol{y}^T\Sigma_y^{-1}\boldsymbol{y}+\ln|\Sigma_y|+\sum_{k=1}^{K}\psi(\xi_k),} \tag{13.109}$$

where

$$\psi(\xi_k):=-2\ln\phi(\xi_k),$$

and

$$\Sigma_y:=\lambda I+\Phi\Xi\Phi^T,$$

and the constants have been neglected. In words, the heart of the Type I method beats around Eq. (13.103) and minimization takes place in the $\boldsymbol{\theta}$-parameters space, while that of Type II method beats around Eq. (13.109) and the optimization is performed in the $\boldsymbol{\xi}$-parameters (variational) space.

However, the two approaches are not as different as they may appear at first glance. The Type I optimization task, based on the loss function in Eq. (13.103), can also be expressed in the $\boldsymbol{\xi}$-parameters space. To this end, let us apply a variational bound on $g$ in Eq. (13.103) (after all, it is the specific nature of $g$ that led to the bound in Eq. (13.106)); then it is shown ([83], Problem 13.20) that the Type I estimator can also result by minimizing a rigorous upper bound of $L_I(\boldsymbol{\theta},\lambda)$, that is,

$$L_I^{\xi}(\boldsymbol{\xi},\lambda):=\boldsymbol{y}^T\Sigma_y^{-1}\boldsymbol{y}+\Sigma_{k=1}^{K}f_I(\xi_k), \tag{13.110}$$

where

$$f_I(\xi_k):=\ln\xi_k+\psi(\xi_k).$$

Once $\hat{\boldsymbol{\xi}}_I$ is obtained, that is,

$$\hat{\boldsymbol{\xi}}_I=\arg\min_{\boldsymbol{\xi}}\mathcal{L}_I^{\xi}(\boldsymbol{\xi},\lambda),\quad \xi_k>0,\ k=1,2,\ldots,K,$$

then the optimizer in Eq. (13.104) is obtained as

$$\hat{\boldsymbol{\theta}}_I=\Xi_I\Phi^T(\lambda I+\Phi\Xi_I\Phi^T)^{-1}\boldsymbol{y}, \tag{13.111}$$

with

$$\Xi_I := \text{diag}\{\hat{\xi}_{I1}, \ldots, \hat{\xi}_{IK}\}.$$

In [83], it is pointed out that the correspondence between the two formulations carries on to the local minima as well. That is, $\hat{\boldsymbol{\theta}}_I$ is a local minimum of Eq. (13.103), iff $\hat{\boldsymbol{\xi}}_I$ is a local minimum of Eq. (13.110).

Following similar arguments, one can show that optimizing Eq. (13.109), with respect to $\boldsymbol{\xi}$, can equivalently be expressed in the $\boldsymbol{\theta}$-parameters space (Problem 13.21) via the cost function

$$L_{II}^{\theta}(\boldsymbol{\theta}, \lambda) = ||\boldsymbol{y} - \Phi\boldsymbol{\theta}||^2 + \lambda g_{II}(\boldsymbol{\theta}), \tag{13.112}$$

where

$$g_{II}(\boldsymbol{\theta}, \lambda) = \arg\min_{\boldsymbol{\xi}} \left( \sum_{k=1}^{K} \frac{\theta_k^2}{\xi_k} + \ln|\Sigma_y| + \sum_{k=1}^{K} \psi(\xi_k) \right), \quad \xi_k > 0, \ k = 1, 2, \ldots, K, \tag{13.113}$$

and $\hat{\boldsymbol{\theta}}_{II}$ is given by Eq. (13.107) using the optimal $\boldsymbol{\xi}$ parameters. Moreover, as pointed out in [83], this correspondence between the two formulations extends to the local minima, under certain assumptions.

Comparing Eq. (13.110) with Eq. (13.109), and Eq. (13.112) with Eq. (13.103), one observes the essential difference between Type I and Type II approaches, that is, the presence of the $\ln|\Sigma_y|$ term in the latter case. This is the result of the marginalization step in the Bayesian methods. Integration over $\boldsymbol{\theta}$ introduces dependencies among variables, giving rise to this coupling term. Viewing it as a penalizing term in the least-squares cost function, this term does not possess the separable structure of $\sum_k g(\theta_k)$. This is the power and at the same time the drawback of the Bayesian approach. It provides more information, which can be used to the benefit of the performance, but at the same time increases the computational load. In [65], a technique for optimizing Eq. (13.112) is proposed as an alternative to the slow, more standard techniques, such as the EM. The method relies on the introduction of a variational bound on $\ln|\Sigma_y|$, which leads to a majorization-minimization scheme (see also [66]).

Finally, taking advantage of the $\boldsymbol{\theta}$-parameters space formulation of the Type II estimation path, the following performance-related results have been shown in [83]:

- If $\psi(\xi_k) = 0, \ k = 1, 2, \ldots, K$,[9] $g_{II}(\boldsymbol{\theta}, \lambda)$ in Eq. (13.113) is nondecreasing and concave and every local minimum of Eq. (13.112) has *at most* $N$ nonzero elements, regardless of $\lambda$.
- For the noiseless case task, $\boldsymbol{y} = \Phi\boldsymbol{\theta}$, if $\Phi$ is of full spark and there is at least one feasible solution, such that $||\boldsymbol{\theta}||_0 < N$, and $\psi(\xi_k) = 0, \ \forall k$, the set of global minimizers of the Type II task equals the set of global $\ell_0$ norm minimizers.

### *Parameter identifiability and sparse Bayesian modeling*

The task of parameter identifiability refers to the conditions under which the set of parameters that define a model can be learned, if this is possible. In a more general setting, given a parametric model, $f(\boldsymbol{y}; \boldsymbol{\theta})$, associated with a set of variables, $\boldsymbol{y}$, we say that the parameters, $\boldsymbol{\theta}$, are *identifiable* if $f(\boldsymbol{y}; \boldsymbol{\theta}_1) \neq f(\boldsymbol{y}; \boldsymbol{\theta}_2), \ \forall \ \boldsymbol{\theta}_1 \neq \boldsymbol{\theta}_2$. From such a perspective, some of the sparsity conditions discussed in Chapter 9 are identifiability conditions. The task of parameter identifiability in the context of sparse Bayesian learning has been treated in [56].

[9] This condition can be relaxed to concave and nondecreasing functions.

The starting point is the model in Eq. (13.102), where a zero mean Gaussian prior is chosen for $\theta$, with a *diagonal* covariance matrix, $\Sigma_\theta$. Moreover, it is assumed that only $k$ out of the $K > N$ elements of this matrix are nonzero. Then, we know that the marginal of the observations, after marginalizing out $\theta$, (evidence) is also Gaussian with a covariance matrix given by Eq. (12.15). The goal of the task reported in [56] is to study whether it is possible, and, if yes, under which conditions Type II maximum likelihood can recover the elements of $\Sigma_\theta$. It turns out that this is indeed possible, provided that $N$ and $K$ follow an implicit relation via the Khatri-Rao product of the matrix $\Phi$. It turns out that model identifiability can still be guaranteed even if $k > N$. Moreover, this is possible without the enforcement of an extra constraint; note that this is not possible by minimizing the LS cost function, which is associated with the Type I maximum likelihood. Furthermore, the method recovers the true values of the elements of $\Sigma_\theta$ and asymptotically attains the Cramer-Rao bound. In contrast, if the conditions are violated, then a regularizing constraint, such as the $\ell_1$ norm is required to guarantee that the Fisher information matrix is nonsingular.

## 13.11 EXPECTATION PROPAGATION

*Expectation propagation* is an alternative to the variational techniques for approximating posterior pdfs. The task of interest is the same as the one treated in the beginning of this chapter, in Section 13.2. Assume that we are given a set of observations $\mathcal{X}$, which are distributed according to $p(\mathcal{X}|\boldsymbol{\theta})$, and a prior, $p(\boldsymbol{\theta})$, corresponding to the set of the unknown parameters[10] $\theta$. The goal is to obtain an estimate of the posterior, $p(\boldsymbol{\theta}|\mathcal{X})$, assuming that its computation is intractable.

Let us denote by $q(\boldsymbol{\theta})$ the estimate of the posterior. The starting point is to compute $q$ by minimizing the Kullback-Leibler divergence,

$$\mathrm{KL}(p||q) = \int p(\boldsymbol{\theta}|\mathcal{X}) \ln \frac{p(\boldsymbol{\theta}|\mathcal{X})}{q(\boldsymbol{\theta})} \, d\boldsymbol{\theta}. \qquad (13.114)$$

Note that $\mathrm{KL}(p||q)$ is different than the $\mathrm{KL}(q||p)$ divergence, which is involved in the bound in Eq. (13.2). Because the KL divergence is not symmetric, the two methods minimize a different cost. Before proceeding any further, it is important to highlight some implications associated with the two forms of the KL divergence.

- *I-Projection*: The $\mathrm{KL}(q||p)$ divergence is given by

$$\mathrm{KL}(q||p) = \int q(\boldsymbol{\theta}) \ln \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathcal{X})} \, d\boldsymbol{\theta}. \qquad (13.115)$$

  This is sometimes known as *I-projection* or *information projection*. Looking carefully at it, note that in regions of the parameter space where $p(\boldsymbol{\theta}|\mathcal{X})$ assumes small values, $\mathrm{KL}(q||p)$ gets large values and minimization pushes $q(\boldsymbol{\theta})$ to small values as well. Consider now the case that $p(\boldsymbol{\theta}|\mathcal{X})$ is bimodal, while $q(\boldsymbol{\theta})$ is constrained to be unimodal. Then, minimizing $\mathrm{KL}(q||p)$ will force $q$ to be placed close to either of the two peaks of $p$ in order to get small values in the regions where $p$ takes small values, too.

[10] If other hidden variables are also involved, we consider them as part of $\theta$.

- *M-Projection*: We now turn our focus to KL$(p||q)$ divergence, defined in Eq. (13.114). This is also known as *M-projection* or *moment projection*. For the case discussed before, in the regions where $p$ assumes large values, then KL$(p||q)$ gets large values and minimization estimates $q$ in order to have large values in these regions, too. Thus, the estimate $q$ is placed in order for its mode to lie somewhere between the two modes of $p$, as a compromise between the two. Obviously, this is not a good result, because the estimate puts high-probability mass in regions where $p$ assumes small values. This discussion points out some limitations on the performance that the expectation propagation method is expected to exhibit in practice, because it is based on KL$(p||q)$ minimization.

We now assume that $p(\mathcal{X}, \boldsymbol{\theta})$ can be factorized, that is,

$$p(\mathcal{X}, \boldsymbol{\theta}) = \prod_j f_j(\boldsymbol{\theta}). \tag{13.116}$$

For example, such a product can cover the case where

$$p(\mathcal{X}, \boldsymbol{\theta}) = \prod_n p(\boldsymbol{x}_n|\boldsymbol{\theta})p(\boldsymbol{\theta}),$$

where $p(\boldsymbol{\theta})$ is the corresponding prior. The more general formulation of the factorization, used in Eq. (13.116), can serve the needs for more general tasks, as for example graphical models to be treated in Chapter 15. Thus, we can now write that

$$\boxed{p(\boldsymbol{\theta}|\mathcal{X}) = \frac{1}{p(\mathcal{X})} \prod_j f_j(\boldsymbol{\theta}),} \tag{13.117}$$

where $p(\mathcal{X})$ is the evidence of the model. The estimate $q$ will be chosen to be given in a factorized form, as in the variational approach in Section 13.2, that is,

$$\boxed{q(\boldsymbol{\theta}) = \frac{1}{Z} \prod_j \hat{f}_j(\boldsymbol{\theta}),} \tag{13.118}$$

where $\hat{f}_j(\boldsymbol{\theta})$ corresponds to $f_j(\boldsymbol{\theta})$ and $Z$ is the normalizing constant. The next assumption is that the $q(\boldsymbol{\theta})$ is constrained to lie within the exponential family of pdfs (Section 12.4) and

$$q(\boldsymbol{\theta}) := g(\boldsymbol{\eta})h(\boldsymbol{\theta}) \exp\left(\boldsymbol{\eta}^T \boldsymbol{u}(\boldsymbol{\theta})\right). \tag{13.119}$$

### Minimizing the KL divergence

Plugging into Eq. (13.114) the definition in Eq. (13.119) and collecting all terms that are independent of $\boldsymbol{\eta}$ in a constant, we readily obtain

$$\text{KL}(p||q) = -\ln g(\boldsymbol{\eta}) - \int p(\boldsymbol{\theta}|\mathcal{X}) \left(\boldsymbol{\eta}^T \boldsymbol{u}(\boldsymbol{\theta})\right) d\boldsymbol{\theta} + \text{constants}. \tag{13.120}$$

Taking the gradient with respect to $\boldsymbol{\eta}$ and equating to zero we get

$$-\frac{1}{g(\boldsymbol{\eta})} \nabla g(\boldsymbol{\eta}) = \mathbb{E}_p \left[\boldsymbol{u}(\Theta)\right]. \tag{13.121}$$

However, from Eq. (13.119) we have that

$$g(\boldsymbol{\eta})\int h(\boldsymbol{\theta})\exp\left(\boldsymbol{\eta}^T\boldsymbol{u}(\boldsymbol{\theta})\right)\,\mathrm{d}\boldsymbol{\theta}=1,$$

and taking the gradient with respect to $\boldsymbol{\eta}$ results in

$$\begin{aligned}\mathbf{0}=\nabla g(\boldsymbol{\eta})&\int h(\boldsymbol{\theta})\exp\left(\boldsymbol{\eta}^T\boldsymbol{u}(\boldsymbol{\theta})\right)\,\mathrm{d}\boldsymbol{\theta}\\&+g(\boldsymbol{\eta})\int h(\boldsymbol{\theta})\exp\left(\boldsymbol{\eta}^T\boldsymbol{u}(\boldsymbol{\theta})\right)\boldsymbol{u}(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}\end{aligned}$$

or

$$-\frac{1}{g(\boldsymbol{\eta})}\nabla g(\boldsymbol{\eta})=\mathbb{E}_q\left[\boldsymbol{u}(\Theta)\right],$$

which combined with Eq. (13.121) finally results in

$$\boxed{\mathbb{E}_q\left[\boldsymbol{u}(\Theta)\right]=\mathbb{E}_p\left[\boldsymbol{u}(\Theta)\right]:\quad\text{Moment Matching.}}\tag{13.122}$$

The latter is an elegant equation known as *moment matching*. It basically states that at the optimum, $q(\boldsymbol{\theta})$, the expectations of its sufficient statistics are equal to the expectations associated with the pdf to be learned. For example, if $q$ is chosen to be a Gaussian, the sufficient statistics involves the mean and the covariance matrix. Thus, all one has to do is compute the mean and covariance with respect to $p(\boldsymbol{\theta})$ (assuming that they can be obtained) and use them to define the respective Gaussian.

### The expectation propagation algorithm

We will now make use of the moment matching result to obtain the factors, $\hat{f}_j(\boldsymbol{\theta})$, one at a time. The algorithm starts from some initial estimates, $\hat{f}_j^{(0)}$. Let us assume that we are currently seeking to update factor $\hat{f}_k(\boldsymbol{\theta})$. Let $q^{(i)}(\boldsymbol{\theta})$ be the currently available estimate of $q(\boldsymbol{\theta})$, at the $i$th iteration.

***Step 1*:** Remove $\hat{f}_k^{(i)}(\boldsymbol{\theta})$ from $q^{(i)}(\boldsymbol{\theta})$, and define

$$q_{/k}^{(i)}(\boldsymbol{\theta}):=\frac{q^{(i)}(\boldsymbol{\theta})}{\hat{f}_k^{(i)}(\boldsymbol{\theta})}.\tag{13.123}$$

***Step 2*:** Define the pdf

$$\frac{1}{Z_k}f_k(\boldsymbol{\theta})q_{/k}^{(i)}(\boldsymbol{\theta}).\tag{13.124}$$

In other words, in the current estimate $q^{(i)}(\boldsymbol{\theta})$, $\hat{f}_k^{(i)}(\boldsymbol{\theta})$ is replaced by $f_k(\boldsymbol{\theta})$, and $Z_k$ is the corresponding normalizing constant.

***Step 3*:** Compute the normalizing constant,

$$Z_k=\int f_k(\boldsymbol{\theta})q_{/k}^{(i)}(\boldsymbol{\theta})d\boldsymbol{\theta}.\tag{13.125}$$

***Step 4*:** In this step, the optimization is performed by minimizing the KL divergence,

$$\mathrm{KL}\left(\frac{1}{Z_k}f_k(\boldsymbol{\theta})q_{/k}^{(i)}(\boldsymbol{\theta})||q^{(i+1)}(\boldsymbol{\theta})\right).$$

This is achieved by moment matching, and the new $q^{(i+1)}$ is defined so that the expectations of the respective sufficient statistics are matched to those of $\frac{1}{Z_k}f_k(\boldsymbol{\theta})q^{(i)}_{/k}(\boldsymbol{\theta})$, and this operation is assumed to be computationally tractable.

***Step 5***: Compute $\hat{f}_k^{(i+1)}$ such that

$$\hat{f}_k^{(i+1)}(\boldsymbol{\theta}) := K\frac{q^{(i+1)}(\boldsymbol{\theta})}{q^{(i)}_{/k}(\boldsymbol{\theta})}, \tag{13.126}$$

where the proportionality constant is computed so that

$$\int \hat{f}_k^{(i+1)}(\boldsymbol{\theta})q^{(i)}_{/k}(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta} = \int f_k(\boldsymbol{\theta})q^{(i)}_{/k}(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}, \tag{13.127}$$

which results in $K = Z_k$.

The procedure is then applied for the estimation of $\hat{f}_{k+1}^{(i+1)}$. For convergence, more than one passes have to be performed. The evidence can be approximated as

$$p(\mathcal{X}) \approx \int \prod_j \hat{f}_j(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}. \tag{13.128}$$

A detailed application of the algorithm in the context of a simple-to-follow example is given in [42].

*Remarks 13.5.*

- In general, there is no guarantee that the algorithm will converge, which is a major disadvantage of the method. However, it can be shown that if the iterations do converge, the solution is a stationary point of a particular energy function [42]. Recall that in the variational Bayes approach, there is guarantee of convergence to a local optimum point. Of course, one could optimize the KL divergence for the expectation propagation method directly, which guarantees convergence, but in this case the algorithm is more complex and slow.
- Taking into account our discussion concerning the two forms of KL divergence, the expectation propagation method results in poor performance when the true posterior is multimodal. However, for other scenarios, such as logistic-type models, the expectation propagation method can offer competitive and sometimes better performance compared to the variational methods or methods built around the Laplacian approximation (see, e.g., [34, 42]).
- The expectation propagation algorithm was first proposed in [42], and it is a modification of what was known before as *assumed density filtering* (ADF) or *moment matching* (e.g., [51] and the references therein).
- Looking at the factors, $f_j(\boldsymbol{\theta})$, in a more general view, it turns out that the expectation propagation offers the vehicle for obtaining a range of message passing algorithms in the context of probabilistic graphical models (Chapter 15) [43].
- *$\alpha$-Divergence*: Having spent time discussing in some detail the two forms of KL divergence, it is interesting to point out that both formulations can be obtained as special cases of a more general family known as the $\alpha$ family of divergences, defined as

$$\boxed{D_\alpha(p||q) := \frac{4}{1-\alpha^2}\left(1 - \int p(x)^{(1+\alpha)/2}q(x)^{(1-\alpha)/2}\,\mathrm{d}x\right) : \ \alpha - \text{Divergence},} \tag{13.129}$$

where $\alpha \in \mathbb{R}$ is a parameter. Note that $\mathrm{KL}(p||q)$ is obtained at the limit $\alpha \to 1$ and $\mathrm{KL}(q||p)$ is obtained for $\alpha \to -1$. $D_\alpha(p||q)$ is nonnegative and it becomes zero if $p = q$ (see, e.g., [2]).

## 13.12 NONPARAMETRIC BAYESIAN MODELING

The Bayesian approach to parametric modeling has been the focus of our attention in the current and previous chapters. The underlying assumption was that the number of the unknown parameters was fixed and finite. We now turn our attention to a more general task. We will assume that the hidden structure of our model is not fixed but is allowed to grow with the data. In other words, its complexity is not specified a priori but is left to be determined from the data. This is the reason that such models are called *nonparametric*; recall from Chapter 3 that a model is called parametric if the number of free parameters is fixed and independent of the size of the data set.

We will avoid treating nonparametric Bayesian models in a mathematically rigorous sense. Such a path would take us a bit far from the purpose of this book and also from the mathematical skills of the average reader. Thus, we will be content with presenting the main concepts in a mathematically "humble" way. Once the basics have been grasped, the keen reader can delve deeper into the topic by referring to more specialized literature (see, e.g., [23]).

The idea behind nonparametric Bayesian models will be demonstrated via our familiar mixture modeling task, treated in Section 13.4. There, $K$ mixtures (clusters) were assumed; each mixture was modeled via a Gaussian pdf with unknown mean and precision matrix, $(\boldsymbol{\mu}_k, Q_k), k = 1, 2, \ldots, K$. These, in turn, were considered as random entities and were dressed up with a prior—a Gaussian pdf for the mean and a Wishart one for the precision matrix. The probabilities, $P_k,\ k = 1, 2, \ldots, K$, for each mixture were treated either as constants, which were optimized during the M-step, or they were considered random variables and a Dirichlet pdf prior was associated with them (Problem 13.7). The goal was to obtain an estimate of the *posterior probabilities* of the labels associated with each observation point, $P(k_n|\mathcal{X})$. There, we resorted to variational techniques and the mean field approximation and the posterior was approximated by the function $q_z(\mathcal{Z}) = q_z(z_1, \ldots, z_N)$, where $z_n,\ n = 1, 2, \ldots, N$, were $0-1$ coding vectors, with a one placed at the location corresponding to a specific mixture ($k$) and the rest of the elements being zero. In this way, an equivalent clustering of the observation points was achieved in $K$ Gaussian-distributed clusters.

The nonparametric counterpart of the previous task is expressed in almost the same way with a single important difference. The number of mixtures, $K$, is not fixed to a finite value; as a matter of fact, the number of mixtures is allowed to be *countably infinite*. There are two questions that now pop up: (a) how can one deal with an infinite number of clusters, and (b) how can one deal with a prior related to infinite many probability values?

To give an indication of how to deal with infinity, recall that in nonparametric modeling the number of data points is still finite and equal to $N$. Thus, whatever model one adopts, there is no way of having more than $N$ mixtures (clusters); the latter corresponds to the worst case scenario, where each one of the points belongs to a different cluster. Hence, although in theory one can have infinite many clusters, only a finite subset of them is *nonempty*. Thus, all we need to do is obtain an explicit representation of the nonempty mixture components.

Concerning the second question, it can be shown that a prior distribution over an infinite number of groupings, $P(\mathcal{Z})$, that favors assigning data to a small number of groups, is the *Chinese restaurant* *process* (CRP); this is a distribution over infinite partitions of the integers, [1, 19, 55]. An easy way to follow the steps leading to CRP as a limiting case ($K \to \infty$) of the finite mixture modeling is given in [64].

### 13.12.1 THE CHINESE RESTAURANT PROCESS

The name draws from the seemingly infinite number of tables in some very large Chinese restaurants in California. Each table is associated with one cluster/mixture and each customer with one observation. The first customer sits at the first table. The second customer sits at the first table with probability $\frac{1}{1+\alpha}$ and at a new table with probability $\frac{\alpha}{1+\alpha}$. The $n$th customer sits at one of the previously occupied tables with a probability proportional to the people who already sit at it, and he/she sits at a new table with a probability proportional to $\alpha$. The parameter $\alpha$ is known as the *concentration parameter*. The larger its value, the more tables are occupied and the fewer the customers who sit at a single table. In a more formal way, let $k_n$ denote the table for the $n$th customer. Then, we can write

$$P(k_n = k|k_{1:n-1}) = \begin{cases} \dfrac{n_k}{n-1+\alpha}, & \text{if } k \leq K_{n-1}, \\ \dfrac{a}{n-1+\alpha}, & \text{otherwise,} \end{cases} \tag{13.130}$$

where $K_{n-1}$ is the number of tables occupied by the previously $n-1$ arrived customers and $n_k$ the number of customers already sitting at table $k$. It can be shown that the expected number of occupied tables grows as $\alpha \ln n$; that is, the expected number of clusters grows with the number of data (e.g., [16]). The rule in Eq. (13.130) provides the sampling philosophy for assigning data to clusters as new data arrive sequentially. It can be shown (see, e.g., [16]) that the resulting probability $P(k_1, \ldots, k_n)$ is independent (up to label changes) of the sequence in which data arrive; this is an important invariance property.

### 13.12.2 INFERENCE

Having defined an appropriate prior over the partitions (clusters), the next goal is to perform inference and compute the corresponding posterior, $P(k_1, \ldots, k_N|\mathcal{X})$; this step basically reveals the clustering structure by telling us which latent structure is more likely to have generated the data. This is not a computationally tractable task, and one has to resort either to variational approximation methods, for example, [9, 33], or to Monte Carlo techniques, to be discussed in Chapter 14. For the latter case, the CPR model is particularly convenient for Gibbs sampling (Section 14.9); see, for example, [49]. For variational inference, a different representation of Dirichlet processes seems more appropriate (Section 13.12.4).

### 13.12.3 DIRICHLET PROCESSES

This section provides a brief discussion of the more general mathematical framework in which CRPs belong. This section can be bypassed in a first reading.

The notion of stochastic processes was introduced in Chapter 2, Section 2.4. The *Dirichlet process* (DP), $G$, is a distribution of distributions and it is defined in terms of (a) the *concentration parameter*, $\alpha$; and (b) the *base distribution*, $G_0$, over a space $\Theta$, and we write $G \sim \text{DP}(\alpha, G_0)$. We say that

$G \sim$DP$(\alpha, G_0)$ is a Dirichlet process, if for *any* partition[11] $T_i,\ i = 1, 2, \ldots, K$ of $\Theta$, that is, $\Theta = \cup_{i=1}^{K} T_i$, the following holds true:

$$(G(T_1), \ldots, G(T_K)) \sim \text{Dir}\left(\alpha G_0(T_1), \ldots, \alpha G_0(T_K)\right), \tag{13.131}$$

where $G_0(T_i)$ is the probability mass corresponding to $T_i$. In other words, $G(T_i),\ i = 1, 2, \ldots, K$, are jointly distributed according to a Dirichlet distribution (Chapter 2). Dirichlet processes were defined by Ferguson [13]. Two important theorems were proved there:

- Random distributions drawn from a DP, $G \sim$DP$(\alpha, G_0)$, are *discrete*. That is, they place their probability mass on a *countably* infinite number of points, $\boldsymbol{\theta}_k,\ k \in \mathbb{Z}$. Thus, we can write

$$\boxed{G = \sum_{k=1}^{\infty} P_k \delta_{\boldsymbol{\theta}_k}(\boldsymbol{\theta}).} \tag{13.132}$$

  The points $\boldsymbol{\theta}_k$ are called the *atoms* and are i.i.d. drawn from the base distribution, $\boldsymbol{\theta}_k \sim G_0$, and $P_k$ are the corresponding probabilities.
- Due to the discrete nature of the atoms, if we keep sampling we are going to get more and more repetitions of the previously drawn samples. This leads to a clustering structure, which is associated with DPs. Moreover, the structure of the shared values defines a partition of the integer set whose distribution is a Chinese restaurant process (CRP).

## 13.12.4 THE STICK-BREAKING CONSTRUCTION OF A DP

Besides CRP, the *stick-breaking representation* of a DP was developed in [67]. This representation is the one that is usually used in the variational inference approach to nonparametric mixture modeling (e.g., [9]).

Consider a stick of unit length, Figure 13.14. The stick will be divided into a sequence of infinite many segments, of length $P_k,\ k = 1, 2, \ldots$, according to the following algorithm. First, choose a beta (Chapter 2) distributed variable, $\beta_1 \sim \text{Beta}(\beta|1, \alpha)$, and break off a segment of the stick of length equal to $\beta_1$. Then, choose another variable, $\beta_2 \sim \text{Beta}(\beta|1, \alpha)$, and break off another segment of length $\beta_2$. Then, the $k$th step of the algorithms can be written as

$$\beta_k \sim \text{Beta}(\beta|1, \alpha),$$
$$P_k = \beta_k \prod_{i=1}^{k-1}(1 - \beta_i), \quad k = 1, 2, 3, \ldots$$

Using the resulting from the algorithm sequence of $P_k$, a random distribution is formed according to Eq. (13.132), with $\boldsymbol{\theta}_k$ drawn i.i.d. from $G_0$. It can be shown that the distribution of this random distribution is a DP, $G \sim$DP$(\alpha, G_0)$.

We will return to the DPs in Chapter 19, in the context of factor analysis for latent variables modeling. A concise tutorial concerning DPs can also be found in [16]. There, a number of sites with publicly available software tools are also provided.

[11] Strictly speaking, we should say measurable partition; a partition is measurable if it is closed under complementation and countable union.

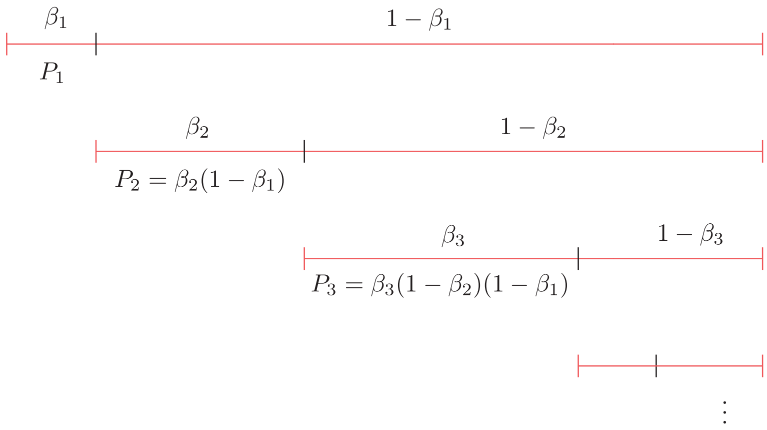


**FIGURE 13.14**

The stick-breaking construction of a DP.

**Example 13.5.** This example illustrates the computational evolution of the variational inference method in [9] for a two-dimensional Gaussian mixture model based on the Chinese restaurant process. The data are generated according to five separate Gaussian distributions, with parameters

$$\boldsymbol{\mu}_1=[-12.5,2.5]^T,\boldsymbol{\mu}_2=[-4,-0.1]^T,\boldsymbol{\mu}_3=[2,-3.5]^T,$$

$$\boldsymbol{\mu}_4=[10,8]^T,\boldsymbol{\mu}_5=[3,3]^T,$$

and

$$\Sigma_1=\begin{bmatrix}1.4 & 0.81\\ 0.81 & 1.3\end{bmatrix},\quad \Sigma_2=\begin{bmatrix}1.5 & 0.2\\ 0.2 & 2.1\end{bmatrix},\quad \Sigma_3=\begin{bmatrix}1.6 & 1\\ 1 & 2.9\end{bmatrix},$$

$$\Sigma_4=\begin{bmatrix}0.5 & 0.22\\ 0.22 & 0.8\end{bmatrix},\quad \Sigma_5=\begin{bmatrix}1.5 & 1.4\\ 1.4 & 2.4\end{bmatrix},$$

for the means and covariance matrices, respectively. One hundred data points were generated from the Gaussian mixture, where each Gaussian was assigned an arbitrary number of points. Figure 13.15 depicts the data points as red circles. Variational inference on the model was performed based on the Matlab implementation of the method[12] in [9]. A data set comprising equidistant, closely spaced test points in the area $[-20,15]\times[-8,12]$ was used to compute the approximate predictive distribution estimated by the variational inference method. The contours of the predictive distribution computed during the first, second, and fifth iterations of the algorithm are plotted in Figure 13.15. The algorithm has clearly identified the clusters of the data.

[12] http://sites.google.com/site/kenichikurihara/academic-software

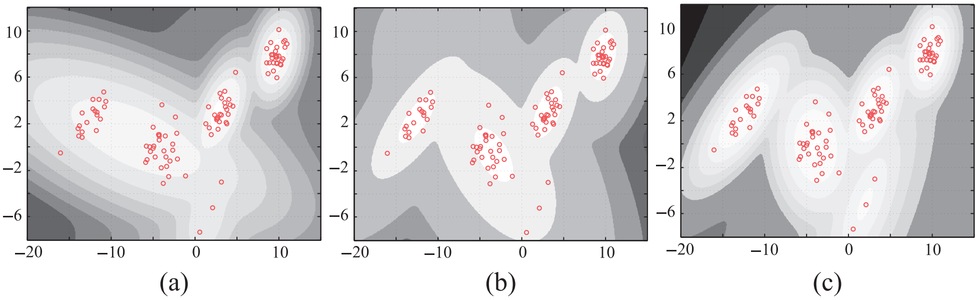


FIGURE 13.15

Contours of predictive distribution for Example 13.5, after (a) the first, (b) the second, and (c) the fifth iteration.

## 13.13 GAUSSIAN PROCESSES

In Section 13.12, the way to impose priors onto the model was similar in spirit with that used for parametric modeling techniques; that is, priors were imposed on the set of unknown parameters. In this section, a different rationale will be adopted. The prior will be placed directly over the space of nonlinear functions, rather than specifying a parametric family of nonlinear functions and placing priors over their parameters.

Let us recall the nonlinear regression task given in Eq. (12.1), that is,

$$\mathrm{y} = \theta_0 + \sum_{k=1}^{K-1} \theta_k \phi_k(\boldsymbol{x}) + \eta = \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}) + \eta, \tag{13.133}$$

where the parameters, $\boldsymbol{\theta}$, are treated as a random vector. Let us define

$$\mathrm{f}(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}),$$

where $\mathrm{f}(\boldsymbol{x})$ is a *random process*. From Chapter 2, we know that a random process is a random entity whose realization (the outcome of an experiment) is a function, $f(\boldsymbol{x})$, instead of a single value. The idea that spans this section is to work directly on $\mathrm{f}(\boldsymbol{x})$ instead of the indirect approach of modeling it via the set of parameters, $\boldsymbol{\theta}$. This is not the first time we have adopted such a path. We silently did it in Chapter 11 while searching for functions in RKH spaces. As a matter of fact, this section can be considered a bridge between the current chapter and Chapter 11.

Recall from Chapter 11 that instead of expanding an unknown function in parameterized form in terms of a number of *preselected* basis functions as in Eq. (13.133), we preferred to search directly for functions that reside in an RKHS; the optimization was carried out with respect to the function itself (not with respect to a set of parameters). In the context of the LS cost function, the optimization was cast as

$$\min_{f \in \mathbb{H}} \sum_{n=1}^{N} \big(y_n - f(\boldsymbol{x}_n)\big)^2 + C\|f\|^2,$$

where $\|\cdot\|$ denotes the norm in $\mathbb{H}$. The goal in this section is to state the "Bayesian counterpart" to this approach. To this end, we will focus on a specific family of processes, known as Gaussian processes, proposed in [50].

**Definition 13.1.** A random process, $\mathrm{f}(\boldsymbol{x})$, is called a *Gaussian process* (GP) iff for *any* finite number of points, $\boldsymbol{x}_{(1)},\ldots,\boldsymbol{x}_{(N)}$, the respective joint probability density function, $p\big(f(\boldsymbol{x}_{(1)}),\ldots,f(\boldsymbol{x}_{(N)})\big)$, is Gaussian.

We know that a set of jointly Gaussian distributed random variables is fully described by the respective mean value and the covariance matrix. In a similar spirit, a Gaussian process is fully determined by its mean value and its *covariance function*, that is,

$$\boldsymbol{\mu}_x = \mathbb{E}\left[\mathrm{f}(\boldsymbol{x})\right], \quad \mathrm{cov}_f(\boldsymbol{x},\boldsymbol{x}') = \mathbb{E}\left[(\mathrm{f}(\boldsymbol{x})-\boldsymbol{\mu}_x)(\mathrm{f}(\boldsymbol{x}')-\boldsymbol{\mu}_{x'})\right].$$

A Gaussian process is said to be *stationary* if $\boldsymbol{\mu}_x = \boldsymbol{\mu}$ and its covariance function is of the form (see also Chapter 2)

$$\mathrm{cov}_f(\boldsymbol{x},\boldsymbol{x}') = \mathrm{cov}_f(\boldsymbol{x}-\boldsymbol{x}').$$

In addition, if $\mathrm{cov}_f(\cdot,\cdot)$ depends on the *magnitude* of the distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$ (i.e., $\|\boldsymbol{x}-\boldsymbol{x}'\|$), the Gaussian process is called *homogeneous*. From now on, we will assume $\boldsymbol{\mu}_x = \boldsymbol{0}$. Before we proceed further, let us establish another connection with Chapter 11.

### 13.13.1 COVARIANCE FUNCTIONS AND KERNELS

For any $N$ and *any* collection of $N$ points, $\boldsymbol{x}_{(1)},\ldots,\boldsymbol{x}_{(N)}$, the respective covariance matrix is defined by

$$\Sigma = \mathbb{E}[\mathbf{f}\mathbf{f}^T],$$

where

$$\mathbf{f} := [\mathrm{f}(\boldsymbol{x}_{(1)}),\ldots,\mathrm{f}(\boldsymbol{x}_{(N)})]^T, \tag{13.134}$$

with elements given by

$$[\Sigma]_{ij} = \mathrm{cov}_f(\boldsymbol{x}_{(i)},\boldsymbol{x}_{(j)}), \quad i,j=1,2,\ldots,N.$$

Because $\Sigma$ is a positive semidefinite matrix, this guarantees that the covariance function is a *kernel* function (Section 11.5.1). To stress this, from now on we will use the notation

$$\mathrm{cov}_f(\boldsymbol{x},\boldsymbol{x}') = \kappa(\boldsymbol{x},\boldsymbol{x}'),$$

and the covariance matrix becomes the corresponding *kernel matrix* denoted as $\mathcal{K}$ (Chapter 11). This change of notation will make the connections with RKH spaces readily spotted. Some typical examples of kernel functions used for Gaussian processes are

- *Linear Kernel*:

  $$\kappa(\boldsymbol{x},\boldsymbol{x}') = \boldsymbol{x}^T\boldsymbol{x}'.$$

  Note that this kernel does not correspond to a stationary process.
- *Squared Exponential or Gaussian Kernel*:

  $$\kappa(\boldsymbol{x},\boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2h^2}\right),$$

where $h$ is a parameter determining the *length scale* of the process. The smaller the value of $h$, the larger the "statistical" similarity (stronger correlation) of two points having a distance $d = ||\boldsymbol{x} - \boldsymbol{x}'||$ apart.

- *Ornstein - Uhlenbeck Kernel*:

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|}{h}\right).$$

- *Rational Quadratic Kernel*:

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = \left(1 + \|\boldsymbol{x} - \boldsymbol{x}'\|^2\right)^{-\alpha}, \quad \alpha \geq 0.$$

Recall from Chapter 2, where random processes were first presented, that a stationary covariance function/kernel has as its Fourier transform the power spectrum of the respective random process; by definition, the power spectrum of a process is a nonnegative function in the frequency domain. This suggests a way of constructing kernels for random processes; that is, take the inverse Fourier transform of a positive function in the frequency domain. Moreover, in principle, all the rules for constructing kernels, which are discussed in Section 11.5.2, can also be applied to construct covariance functions. For example, a popular choice of a kernel for a Gaussian process is

$$\kappa(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}) = \theta_1 \exp\left(-\sum_{m=1}^{M} \frac{(x_i - x_i')^2}{2h_i^2}\right) + \theta_2,$$

where $\theta_1, \theta_2$ are hyperparameters, which define the process.

Figure 13.16a shows examples of different realizations of a stationary Gaussian process with $h = 2$, and Figure 13.16b for $h = 0.2$.

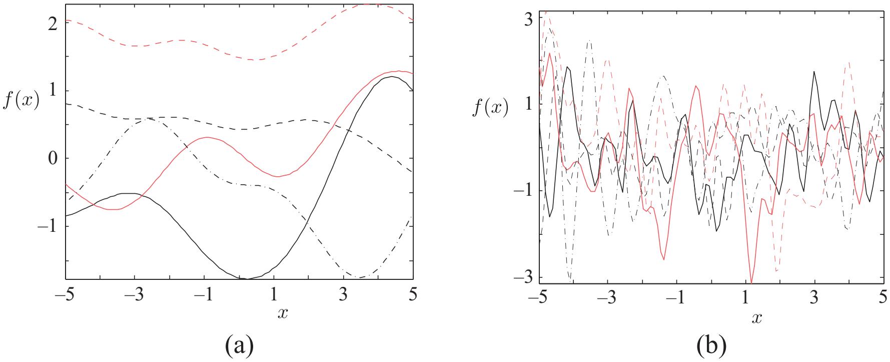


**FIGURE 13.16**

Different realizations of a Gaussian process. (a) Gaussian covariance kernel $h = 2$, (b) $h = 0.2$. Note that when the correlation function fades away fast, the graph of the respective realizations shows a fast variation as a function of the free variable ($x$).

### 13.13.2 REGRESSION

Let us assume that we are given a set $\mathcal{X}$ of input observations, $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$. Recall from Section 12.2 that the main goal in a Bayesian regression task is to obtain the two pdfs,

$$p(\boldsymbol{y}|\mathcal{X}) \quad \text{and} \quad p(y|\boldsymbol{x}, \boldsymbol{y}, \mathcal{X}),$$

where

$$\mathbf{y} = \mathbf{f} + \boldsymbol{\eta}, \quad \mathbf{y} := [\mathrm{y}_1, \dots, \mathrm{y}_N]^T, \tag{13.135}$$

and

$$\mathrm{y} = \mathrm{f}(\boldsymbol{x}) + \eta,$$

and $\mathbf{f}$ is defined in Eq. (13.134). The first of the two pdfs is the joint probability density of the output variables, which are generated by input points in $\mathcal{X}$; the associated randomness is due to $\mathbf{f}$ as well as to the noise $\eta$. The second pdf refers to the prediction of the value of y, given the value of $\boldsymbol{x}$ and the training data $(y_n, \boldsymbol{x}_n), n = 1, 2, \dots, N$. We will drop out $\mathcal{X}$ to unclutter notation, as we did in Section 12.2.

Assuming $\mathrm{f}(\cdot)$ to be a zero-mean Gaussian process, then $\mathbf{f}$ is jointly Gaussian with zero mean and covariance matrix $\mathcal{K}$, dictated by the covariance function/kernel $\kappa(\cdot, \cdot)$, that is,

$$p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}|\mathbf{0}, \mathcal{K}).$$

Also, let $\boldsymbol{\eta}$ be of zero mean with covariance matrix $\Sigma_\eta$ and independent of $\mathrm{f}(\cdot)$; without harming generality, let $\Sigma_\eta = \sigma_\eta^2 I$. Thus,

$$p(\boldsymbol{y}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{f}, \sigma_\eta^2 I).$$

Then, following exactly the same arguments as in Section 12.2, we obtain

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|\mathbf{0}, \mathcal{K} + \sigma_\eta^2 I). \tag{13.136}$$

This is also obvious from the fact that the sum of two independent Gaussian variables is also Gaussian and the mean and covariance matrix can directly be obtained from Eq. (13.135).

To obtain $p(y|\boldsymbol{x}, \boldsymbol{y})$ we can use (13.136) and apply it recursively. It will also be useful here to bring into the notation the number of available observations, $N$, explicitly and write

$$\mathrm{y}_{N+1} = \begin{bmatrix} \mathrm{y} \\ \mathbf{y}_N \end{bmatrix}, \quad \mathbf{y}_N := [\mathrm{y}_1, \dots, \mathrm{y}_N]^T.$$

From Eq. (13.136), $\mathbf{y}_{N+1}$ follows a Gaussian distribution

$$p(\boldsymbol{y}_{N+1}|\mathbf{0}, \Sigma_{N+1}),$$

with

$$\Sigma_{N+1} := \mathcal{K}_{N+1} + \sigma_\eta^2 I_{N+1}.$$

Then from Bayes theorem, we have

$$p(y|\boldsymbol{y}_N) = \frac{p(\boldsymbol{y}_{N+1})}{p(\boldsymbol{y}_N)}. \tag{13.137}$$

However, because the joint pdf is Gaussian, the conditional in Eq. (13.137) is also Gaussian. The respective mean and variance are computed by partitioning the matrix $\Sigma_{N+1}$ (see Section 12.9, Eqs. (12.130) and (12.131)),

$$\Sigma_{N+1} = \begin{bmatrix} \kappa(\boldsymbol{x},\boldsymbol{x}) + \sigma_\eta^2, & \boldsymbol{\kappa}^T(\boldsymbol{x}) \\ \boldsymbol{\kappa}(\boldsymbol{x}), & \Sigma_N \end{bmatrix}, \quad \boldsymbol{\kappa}(\boldsymbol{x}) := [\kappa(\boldsymbol{x},\boldsymbol{x}_1), \ldots, \kappa(\boldsymbol{x},\boldsymbol{x}_N)]^T$$

and

$$\boxed{\begin{aligned} &\mu_y(\boldsymbol{x}) = \boldsymbol{\kappa}^T(\boldsymbol{x})\Sigma_N^{-1}\boldsymbol{y}, \\ &\sigma_y^2(\boldsymbol{x}) = \sigma_\eta^2 + \kappa(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{\kappa}^T(\boldsymbol{x})\Sigma_N^{-1}\boldsymbol{\kappa}(\boldsymbol{x}). \end{aligned}} \tag{13.138}$$

Compare Eq. (13.138) with Eq. (11.27). Taking into account that $\Sigma_N = \mathcal{K}_N + \sigma_\eta^2 I$, $\mu_y(\boldsymbol{x})$ is identical to $\hat{y}$ obtained by the kernel ridge regression, for appropriate choices of $C$ and $\sigma_\eta^2$. However, now we have also obtained information concerning the respective variance of the resulting estimate.

At this point, it is interesting to look back at the Bayesian regression task for parametric modeling in Section 12.2.3, and to remember that the obtained mean value in Eq. (12.20) was the same (for a zero mean prior $p(\boldsymbol{\theta})$) as that provided by the ridge regression, for an appropriate choice of $\lambda$.

*Remarks 13.6.*

- From the previous discussion it is apparent that solving the regression task by resorting to Gaussian processes is the Bayesian answer to solving a regression task in an RKHS. Both approaches share a common advantage. Although the underlying mapping to an RKHS (implied by the adopted kernel) may live in a high-dimensional space, the complexity for solving the task depends on the number of training points, $N$. The source of complexity associated with the Gaussian processes is the inversion of the matrix, which amounts to $\mathcal{O}(N^3)$ operations.
- Both equations in Eq. (13.138) can be obtained from the corresponding equation derived for the linear case of Bayesian learning, covered in Section 12.2.3. Indeed, setting $\boldsymbol{\theta}_0 = \boldsymbol{0}$ in Eq. (12.27) and combining it with Eq. (12.22), we obtain

$$\mu_y(\boldsymbol{x}) = \sigma_\theta^2 \boldsymbol{x}^T X^T \left(\sigma_\eta^2 I + \sigma_\theta^2 X X^T\right)^{-1} \boldsymbol{y}, \tag{13.139}$$

  where $X$ has replaced $\Phi$, because the linear case is treated. Applying now the kernel trick, as discussed in Chapter 11, to replace $\sigma_\theta^2 \boldsymbol{x}_i^T \boldsymbol{x}_j$ with a kernel $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ operation, one readily obtains the corresponding equation in Eq. (13.138).

  In a similar way, one can obtain $\sigma_y^2(\boldsymbol{x})$ in Eq. (13.138) from Eq. (12.23) by using Woodbury's formula for matrix inversion from Appendix A.1 to reformulate Eq. (12.23) according to Eq. (12.28) (try it).

### Dealing with hyperparameters

As we have already stated, the kernel function can be given in terms of some parameters, say, $\boldsymbol{\theta}$, which in turn have to be estimated from the data. There are various ways to deal with this task. The first that comes to mind is to optimize the resulting parameterized log-likelihood, $\ln p(\boldsymbol{y};\boldsymbol{\theta})$, with respect to $\boldsymbol{\theta}$, by taking the gradient and equating to zero. Another way is to assume a prior on the parameters and use Bayesian arguments to integrate them out. The integration is usually intractable and approximate techniques must be used, for example, Monte Carlo methods, Chapter 14. Needless to say that both techniques have their drawbacks. Optimizing the log-likelihood is a nonconvex task that cannot guarantee, in general, a global maximum. On the other hand, Monte Carlo techniques tend to be computationally intensive, requiring many iterations to converge. More on these issues can be found in [58].

#### Computational considerations

In order to reduce the $\mathcal{O}(N^3)$ computational load associated with the inversion of $\Sigma_N$, a number of approximate techniques have been proposed. A possible path is the *sparse GPs*; in these methods, the full Gaussian process model is approximated by using an expansion in terms of a finite set of basis functions. For example, it is common to use as bases the set $\kappa(\boldsymbol{x}, \boldsymbol{u}_m)$, where $\boldsymbol{u}_m, m = 1, 2, \ldots, M \ll N$, is a subset of the input samples known as *active set*. Such techniques can lead to a reduced cost of the order of $\mathcal{O}(M^2 N)$ (e.g., [57]). Other alternatives that do not require the active set to be a subset of the training samples have also been proposed (e.g., [35, 69]). In [77], a variational sparse method is proposed that attempts to alleviate problems encountered when one increases the size of the active set.

A variation of the Gaussian processes approach is to equip it with the ability to forget past samples for time-varying environments; this method has been proposed in [54, 78] as an alternative to the kernel RLS algorithm discussed in Chapter 11. Other variants use transformations of the output variables to make Gaussian models applicable to a wider range of problems, [36, 68].

In [62], the connection between Gaussian processes and Kalman filtering is exploited and the solution is obtained via the involvement of stochastic differential equations, which makes the dependence of the complexity on time to be linear.

### 13.13.3 CLASSIFICATION

In contrast to the regression task, under the Gaussian assumptions for the noise and the involved random process, the classification task gets more involved. In Section 13.7, the logistic regression in its parametric form, given in Eq. (13.67), was employed. In the context of the Gaussian processes, the model becomes

$$P(\omega_1|\boldsymbol{x}) = \frac{1}{1 + \exp\big(-f(\boldsymbol{x})\big)} = \sigma\big(f(\boldsymbol{x})\big),$$

where now $f(\boldsymbol{x})$ will be treated in terms of a Gaussian random process, associated with a kernel function $\kappa(\cdot,\cdot)$. Given a set of training samples, $(y_n, \boldsymbol{x}_n), n = 1, 2, \ldots, N, y_n \in \{0, 1\}$, and following the same arguments as in Section 12.3, we can now write that

$$P(\boldsymbol{y}|\boldsymbol{f}) = \prod_{n=1}^{N} \sigma(f_n)^{y_n} (1 - \sigma(f_n))^{1-y_n},$$

where $f_n := f(x_n)$, and

$$p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{0}, \mathcal{K}).$$

Note that $P(\boldsymbol{y}|\boldsymbol{f})$ is no longer Gaussian and the involved integrations needed to obtain $P(\boldsymbol{y})$ and/or $P(y|\boldsymbol{x}, \boldsymbol{y})$ cannot be performed analytically. There are various ways to perform approximations. One path is to resort to the Laplacian approximation of $p(f(x)|\boldsymbol{y})$ (see Section 12.3) [79]. Another is to use Monte Carlo techniques [47]. In [17], a variational approach has been used to obtain bounds on the logistic sigmoid and approximate the respective product with a product of Gaussians. The expectation propagation method has been used in [51].

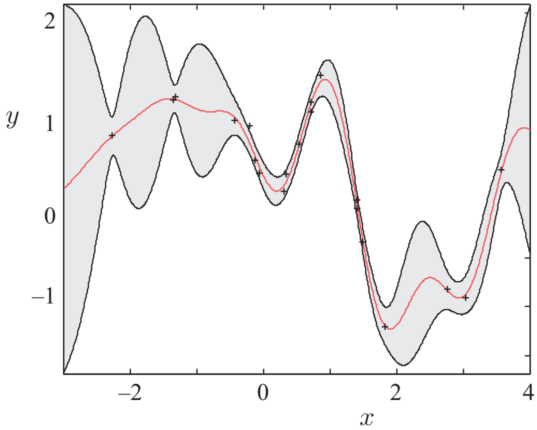


**FIGURE 13.17**

The red line corresponds to the mean of the posterior Gaussian process. The shaded area corresponds to ± twice the standard deviation.

For further reading on Gaussian processes, the interested reader may consult the classical reference [58].

**Example 13.6.** The goal of this example is to demonstrate the usage of Gaussian processes in regression. To this end, $N = 20$ points were randomly sampled from a realization of a Gaussian process, with zero mean and covariance function based on the Gaussian kernel with length scale $h = 0.5$. The corresponding input points were drawn according to a normal distribution of zero mean and unit variance. In the sequel, Gaussian noise was added to these GP points, with variance 0.01, to form the set of observed data (shown as '+' in Figure 13.17). Using these as the training data, predictions of the output variables, corresponding to $D = 1000$ equidistant input points in the interval $[-3, 4]$, were performed; for the prediction, the expressions for the posterior GP mean and variance in Eq. (13.138) were used. The mean of the posterior Gaussian process is illustrated in Figure 13.17 as a solid red line. The shaded area surrounding the curve of the posterior mean corresponds to the error bars $\mu_y \pm 2\sigma_y$ of the posterior prediction. Notice the increase of the posterior prediction variance in regions where observed data points are scarce.

## 13.14 A CASE STUDY: HYPERSPECTRAL IMAGE UNMIXING

*Hyperspectral image unmixing* (HSI) is a typical application of sparse regression modeling under a set of constraints. It is a good "excuse" for us to demonstrate the application of the hierarchical Bayesian modeling approach via a task of great practical importance.

In *hyperspectral remote sensing*, the electromagnetic solar energy emanating from the earth's surface is measured by sensitive scanners located aboard a satellite, an aircraft, or a space station. The scanners are sensitive to a number of wavelength bands of the electromagnetic radiation. Different properties of the earth's surface contribute to the reflection of the energy in the different bands.

For example, in the visible-infrared range, properties such as the mineral and moisture contents of soils, the sedimentation of water, and the moisture content of vegetation are the main contributors to the reflected energy. In contrast, at the thermal end of the infrared, it is the thermal capacity and thermal properties of the surface that contribute to the reflection. Thus, each band measures different properties of the same patch of the earth's surface. In this way, images of the earth's surface corresponding to the spatial distribution of the reflected energy in each band can be created. The task now is to exploit this information in order to identify the various ground cover types, that is, built-up land, agricultural land, forest, fire burn, water, diseased crop, and so on.

Figure 13.18 illustrates the process of generating a pixel's spectral signature out of a hyperspectral image data cube (the cube consists of two spatial and one spectral dimension). Each image corresponds to a single wavelength (band) and each pixel to a specific patch of the earth's surface. The *spectral signature* of a pixel is simply a vector containing radiance values measured in the various spectral bands. Technological advances in recent years have allowed the implementation of imaging spectrometers, which have the ability to collect data in hundreds of adjacent spectral bands. The highly increased volume of data conveys spatial/spectral information that can be properly exploited to accurately determine the type and nature of the objects being imaged.

An intimate limitation of hyperspectral remote sensing is that a single pixel often records a mixed spectral signature of different distinct materials, due to the low spatial resolution of the remote sensor. This raises the need for spectral unmixing (SU) [32], which is a very important step in hyperspectral image processing that has recently attracted strong scientific interest. SU is the procedure of decomposing the measured spectrum of an observed pixel into a collection of constituent spectral

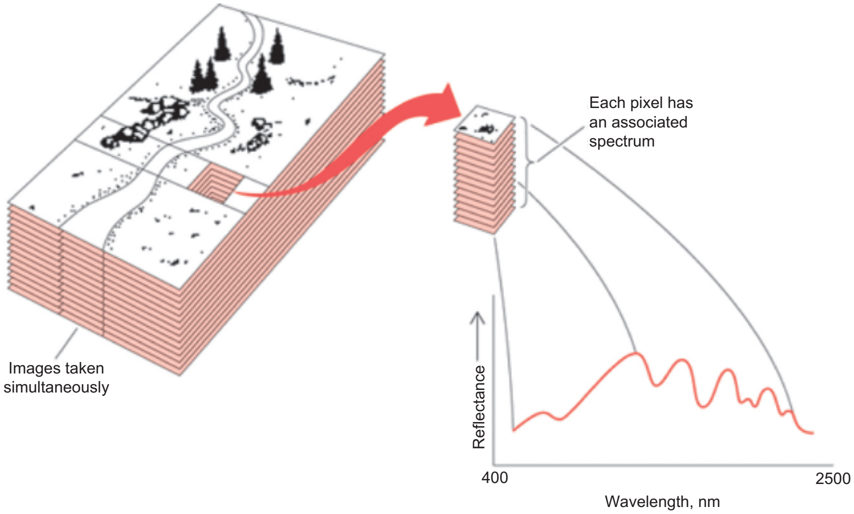

Each pixel has an associated spectrum
Images taken simultaneously
Reflectance
400
2500
Wavelength, nm

**FIGURE 13.18**

Each image corresponds to a specific wavelength band and each pixel to a particular patch of the earth's surface. The signature of a pixel is a vector whose coefficients measure the radiance of the respective earth's patch in the different bands (modified image taken from [61]).

signatures (or *endmembers*) and their corresponding proportions (or *abundances*). A widely used model to perform SU is the linear mixing model.

Assume a remotely sensed hyperspectral image consisting of $M$ spectral bands, and let $\boldsymbol{y} \in \mathbb{R}^M$ be the vector containing the measured spectral signature (i.e., the radiance values in all spectral bands) of a single pixel (specific earth patch). Also let $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_l]$ stand for the $M \times l$ endmember signature matrix, where $\boldsymbol{x}_i \in \mathbb{R}^M,\ i = 1, 2, \dots, l$, comprises the spectral signatures of the $i$th endmember, and $l$ is the total number of (possible) distinct endmembers (earth-surface/material types) present in the scene. Finally, let $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_l]^T$ be the *abundance vector* associated with $\boldsymbol{y}$, where $\theta_i$ denotes the abundance fraction of $\boldsymbol{x}_i$ in $\boldsymbol{y}$. The linear mixing model assumes that there is a linear relationship between the spectra of the measured pixel and the endmembers, expressed as

$$\boldsymbol{y} = X\boldsymbol{\theta} + \boldsymbol{\eta} \tag{13.140}$$

where $\boldsymbol{\eta}$ stands for the additive noise values, which are assumed to be samples of a zero-mean Gaussian distributed random vector, with (i.i.d.) elements, that is, $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\eta}|\boldsymbol{0}, \beta^{-1}I_M)$, where $\beta$ denotes the inverse of the noise variance (precision), and $I_M$ is the $M \times M$ identity matrix. Note that the model in Eq. (13.140) is a typical regression model in its multivariate formulation, because now the output for each measurement is a vector and not a scalar (see also Section 4.9 of Chapter 4). The output variables are measured and the matrix $X$ is assumed known, and indeed there are methods to estimate its elements.

Treating such a model to recover the abundance coefficients would be a straightforward application of what has been said so far in the current and previous chapters of this book. However, there is a physical constraint that has to be considered and that makes the task more interesting. The abundance coefficients are nonnegative, that is,

$$\theta_i \geq 0, \quad i = 1, 2, \dots, l. \tag{13.141}$$

Additionally, a valid assumption is that only a few of the endmembers present in the image will contribute to the spectrum of a single pixel $\boldsymbol{y}$. In other words, the abundance vector $\boldsymbol{\theta}$ accepts a *sparse* representation in $X$.

Thus, our goal is to estimate $\boldsymbol{\theta}$ subject to the nonnegativity as well as the sparsity constraints, given the spectral measurements, $\boldsymbol{y}$, and the endmember matrix, $X$. Obviously, there are different paths to achieve this goal. Because we are currently exploring the Bayesian world, we will employ the Bayesian framework. To this end, an appropriate prior model that expresses our prior belief on the parameters of interest will be first be adopted, and we will then perform Bayesian inference using the variational Bayes methodology, as has been previously discussed.

### 13.14.1 HIERARCHICAL BAYESIAN MODELING

The presence of Gaussian noise in Eq. (13.140) dictates that

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{\theta}, \beta) &= \mathcal{N}(\boldsymbol{y}|X\boldsymbol{\theta}, \beta^{-1}I_M) \\ &= (2\pi)^{-\frac{M}{2}}\, \beta^{\frac{M}{2}} \exp\left(-\frac{\beta}{2}\|\boldsymbol{y} - X\boldsymbol{\theta}\|^2\right). \end{aligned} \tag{13.142}$$

We now turn our attention to selecting suitable priors for the model parameters, which are treated as random variables, $\theta$, $\beta$. As a prior for the nonnegative noise precision $\beta$ we adopt a Gamma distribution (Section 13.3, Eq. (13.24)), expressed as

$$p(\beta) = \text{Gamma}(\beta|c,d) = \frac{d^c}{\Gamma(c)}\beta^{c-1}\exp\left(-d\beta\right), \tag{13.143}$$

where $c$ and $d$ are the respective parameters (set equal to $10^{-6}$ in the experiments).

For the abundance vector θ, we define a two-level hierarchical prior that is expressed in a conjugate form and imposes sparsity as well as nonnegativity on the abundance coefficients. Inspired by [60], a nonnegatively truncated Gaussian prior is selected, that is,

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \mathcal{N}_{\mathbb{R}^l_+}\left(\boldsymbol{\theta}|\mathbf{0}, A^{-1}\right), \tag{13.144}$$

where $\boldsymbol{\alpha} := [\alpha_1, \alpha_2, \ldots, \alpha_l]^T$ is the precision parameter vector, $A = \text{diag}\{\alpha_1, \ldots, \alpha_l\}$ is the corresponding diagonal matrix, and $\mathcal{N}_{\mathbb{R}^l_+}$ signifies the $l$-variate normal distribution truncated at the nonnegative orthant of $\mathbb{R}^l$, denoted by $\mathbb{R}^l_+$ [71]. In the second level of hierarchy, the precision parameters are also considered random variables, $\alpha_i$'s, $i = 1, 2, \ldots, l$, that follow an inverse Gamma distribution, that is,

$$p(\alpha_i) = \text{IGamma}\left(\alpha_i|1, \frac{b_i}{2}\right) = \frac{b_i}{2}\alpha_i^{-2}\exp\left(-\frac{b_i}{2}\frac{1}{\alpha_i}\right), \tag{13.145}$$

where $b_i, i = 1, 2, \ldots, N$, are scale hyperparameters. These two levels of hierarchy form a nonnegatively truncated multivariate Laplace prior over the abundance vector θ, which can be established by integrating out the precision $\boldsymbol{\alpha}$ [71], that is,

$$p(\boldsymbol{\theta}|\mathbf{b}, \beta) = \prod_{i=1}^{l}\sqrt{\beta b_i}\exp\left(-\sqrt{\beta b_i}|\theta_i|\right) I_{\mathbb{R}^l_+}(\boldsymbol{\theta}), \tag{13.146}$$

where $I_{\mathbb{R}^l_+}(\boldsymbol{\theta})$ is the indicator function, with $I_{\mathbb{R}^l_+}(\boldsymbol{\theta}) = 1$ (resp. 0) if $\boldsymbol{\theta} \in \mathbb{R}^l_+$ (resp. $\boldsymbol{\theta} \notin \mathbb{R}^l_+$). In our formulation, the sparsity-promoting scale hyperparameters in Eq. (13.145) are also assumed to be random and are inferred from the data, by assuming the following Gamma prior distribution for each $\mathrm{b}_i, i = 1, 2, \ldots, l$,

$$p(b_i) = \text{Gamma}(b_i|\kappa, \nu) = \frac{\nu^\kappa}{\Gamma(\kappa)}b_i^{\kappa-1}\exp\left(-\nu b_i\right). \tag{13.147}$$

Hyperparameters $\kappa$ and $\nu$ in Eq. (13.147) are also set to small values ($10^{-6}$ in the experiments).

Having adopted the hierarchical Bayesian model, the variational EM algorithm discussed in Section 13.3 is applied with the goal of obtaining estimates, $q(\theta_i), i = 1, 2, \ldots, l$, of the posteriors of the abundance parameters given the observations. In the experiments, the respective mean values of $q(\theta_i)$ will be used as estimates of the unknown parameter values. Details on the derivation can be obtained from [72]. The alternative path to the variational EM algorithm is to employ Monte Carlo techniques (see, e.g., [12]).

### 13.14.2 EXPERIMENTAL RESULTS

The previously described model was applied to a real hyperspectral image, collected by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) over a Cuprite mining district in Nevada in the

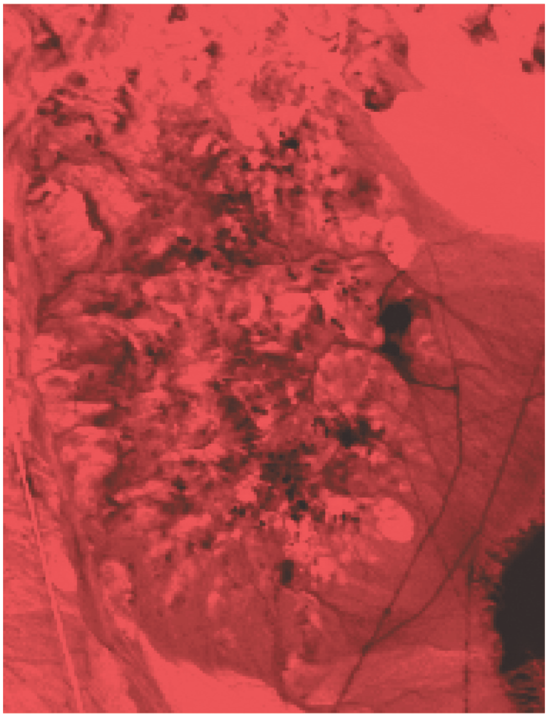


FIGURE 13.19

Composite of the AVIRIS Cuprite subimage using bands 183, 193, and 203 (from [61]). The full RGB color image is available from the site of this book.

summer of 1997.[13] The Cuprite data set has been extensively used to evaluate remote sensing technologies and spectral unmixing algorithms (e.g., [26, 45, 71]). It comprises 224 spectral bands in the range from 400 to 2500 nanometers. A subimage of the Cuprite data set with size 250 × 191 pixels is used in our experiments. Figure 13.19 displays a composite of our image, where bands 183, 193, and 203 have been used, respectively.

After removing some low signal-to-noise ratio (SNR) bands and water-vapor absorption bands, $M = 188$ spectral bands remain available for processing. As a preprocessing step, the VCA algorithm[14] has been used to extract 14 endmembers from our hyperspectral image, as in [45]. The vertex component analysis (VCA) algorithm identifies the signatures of the "pure" pixels in the image and considers them pure material signatures. A plot of the spectral signatures of the extracted endmembers versus the wavelength is displayed in Figure 13.20.

Figure 13.21 shows the resulting abundance maps for six different endmembers, using the variational Bayes method. A dark (resp. light) pixel reveals a low (resp. high) proportional percentage for the

[13] The data are publicly available at http://aviris.jpl.nasa.gov/data/free_data.html.

[14] The VCA code is available at http://www.lx.it.pt/~bioucas/code.htm.

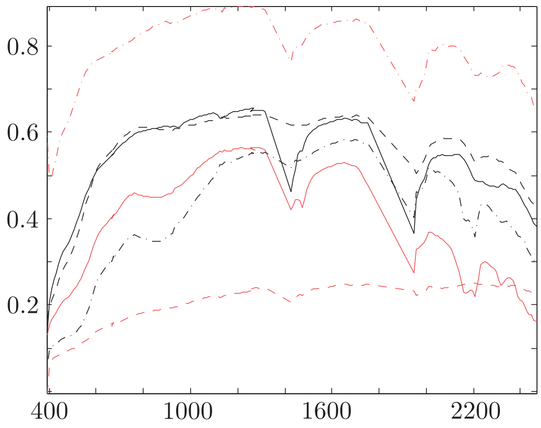


**FIGURE 13.20**

Spectral signatures of 6 out of the 14 endmembers extracted from the Cuprite image using the VCA algorithm [45]. A figure showing all 14 signatures can be downloaded from the site of this book.

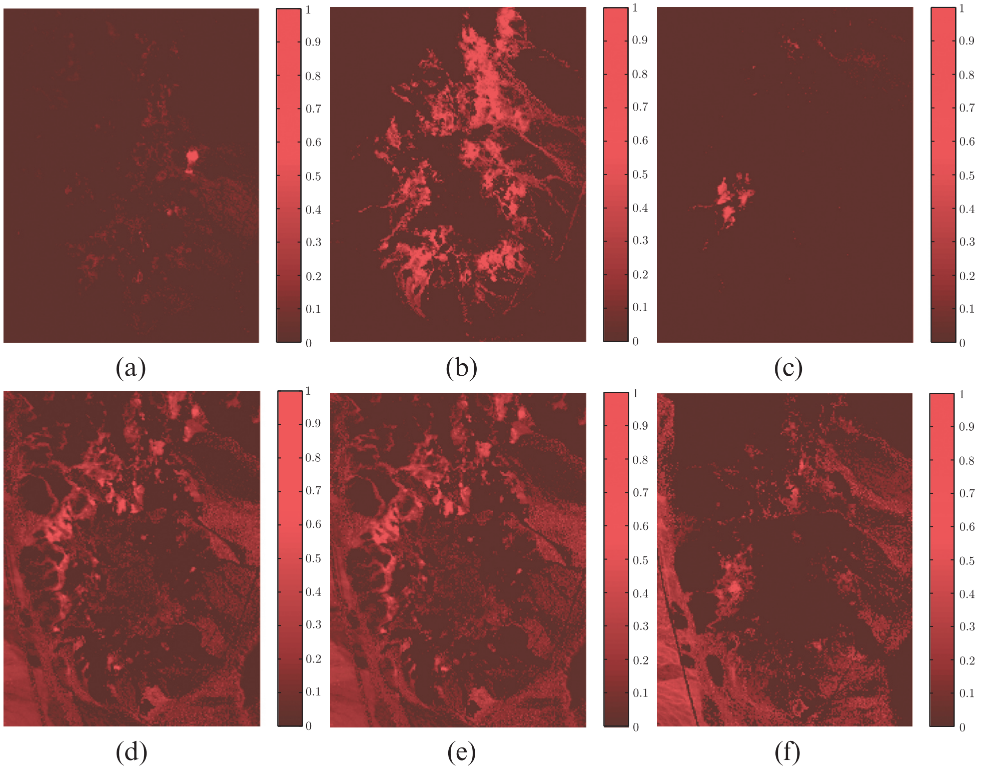


**FIGURE 13.21**

Estimated abundance maps for the materials (a) Muscovite, (b) Alunite, (c) Buddingtonite, (d) Montmorillonite, (e) Kaolinite 1, and (f) Kaolinite 2. The full-color image is available from the site of this book.

respective endmember in that pixel. In other words, each image shows the distribution of values of a specific abundance coefficient, $\theta_i$, over the sensed earth surface.

More important, we are able to identify the presented endmembers in Figure 13.21 as muscovite, alunite, buddingtonite, montmorillonite, kaolinite 1, and kaolinite 2.

## PROBLEMS

**13.1** Show Eq. (13.5).

**13.2** Show Eq. (13.37).

**13.3** Show Eqs. (13.42)–(13.44).

**13.4** Show that if

$$p(x) \propto \frac{1}{x},$$

then the random variable z := ln x follows a uniform distribution.

**13.5** Derive the lower bound after convergence of the variational Bayesian EM for the linear regression task, which is modeled in Section 13.3.

**13.6** Consider the Gaussian mixture model

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} P_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, Q_k^{-1}),$$

with priors

$$p(\boldsymbol{\mu}_k) = \mathcal{N}(\boldsymbol{\mu}_k|0, \beta^{-1}I), \tag{13.148}$$

and

$$p(Q_k) = \mathcal{W}(Q_k|\nu_0, W_0).$$

Given the set of observations $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}, \boldsymbol{x} \in \mathbb{R}^l$, derive the respective variational Bayesian EM algorithm, using the mean field approximation for the involved posterior pdfs. Consider $P_k, k = 1, 2, \dots, K$, as deterministic parameters and optimize the respective lower bound of the evidence with respect to the $P_k$'s.

**13.7** Consider the Gaussian mixture model of Problem 13.6, with the following priors imposed on μ, Q, and **P**:

$$\begin{aligned} p(\boldsymbol{\mu}, Q) &= p(\boldsymbol{\mu}|Q)p(Q) \\ &= \prod_{k=1}^{K} \mathcal{N}\left(\boldsymbol{\mu}_k|\boldsymbol{0}, (\lambda Q_k)^{-1}\right) \mathcal{W}(Q_k|\nu_0, W_o), \end{aligned}$$

that is, a Gaussian-Wishart product and

$$p(\boldsymbol{P}) = \text{Dir}(\boldsymbol{P}|a) \propto \prod_{k=1}^{K} P_k^{a-1},$$

that is, a Dirichlet prior. That is, **P** is treated as a random vector. Derive the E algorithmic steps of the variational Bayesian approximation, adopting the mean field approximation for the involved posterior pdfs. We have adopted the notation $\boldsymbol{\mu}$ in place of $\boldsymbol{\mu}_{1:K}$ and $Q$ in place of $Q_{1:K}$, for notational simplicity.

**13.8** If $\mu$ and Q are distributed according to a Gaussian-Wishart product,

$$p(\boldsymbol{\mu}, Q) = \mathcal{N}(\boldsymbol{\mu}|\hat{\boldsymbol{\mu}}, (\lambda Q)^{-1})\mathcal{W}(Q|\nu, W).$$

Compute the expectation

$$\mathbb{E}[\mu^T Q \mu].$$

**13.9** Derive the Hessian matrix with respect to $\boldsymbol{\theta}$ of the cost function

$$J(\boldsymbol{\theta}) = \sum_{n=1}^{N}\Big[y_n \ln \sigma\big(\boldsymbol{\phi}^T(x_n)\boldsymbol{\theta}\big) + (1 - y_n)\ln\big(1 - \sigma(\boldsymbol{\phi}^T(x_n)\boldsymbol{\theta})\big)\Big] \\ - \frac{1}{2}\boldsymbol{\theta}^T A\boldsymbol{\theta},$$

where

$$\sigma(z) = \frac{1}{1 + \exp(-z)}.$$

**13.10** Show that the marginal of a Gaussian pdf with a gamma prior on the variance, after integrating out the variance, is the student's-t pdf, given by

$$\mathrm{st}(x|\mu, \lambda, \nu) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})}\left(\frac{\lambda}{\pi\nu}\right)^{1/2}\frac{1}{\left(1 + \frac{\lambda(x-\mu)^2}{\nu}\right)^{\frac{\nu+1}{2}}}. \qquad (13.149)$$

**13.11** Derive the pair of recursions Eqs. (13.61)–(13.62).

**13.12** Consider a two-class classification task and assume that the feature vectors in each one of the two classes, $\omega_1,\ \omega_2$, are distributed according to the Gaussian pdf. Both classes share the same covariance matrix $\Sigma$, and the mean values are $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, respectively. Prove that, given an observed feature vector, $\boldsymbol{x} \in \mathbb{R}^l$, the posterior probabilities for deciding in favor of one of the classes is given by the logistic function, that is,

$$P(\omega_2|\boldsymbol{x}) = \frac{1}{1 + \exp\left(-\boldsymbol{\theta}^T\boldsymbol{x} + \theta_0\right)},$$

where

$$\boldsymbol{\theta} := \Sigma^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1),$$

and

$$\theta_0 = \frac{1}{2}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T\Sigma^{-1}(\boldsymbol{\mu}_2 + \boldsymbol{\mu}_1) + \ln\frac{P(\omega_1)}{P(\omega_2)}$$

**13.13** Derive Eq. (13.73).

**13.14** Show Eq. (13.74).

**13.15** Derive the recursion Eq. (13.76).

**13.16** Show that if $f$ is a convex function, $f : \mathbb{R}^l \to \mathbb{R}$, then it is equal to the conjugate of its conjugate, i.e., $(f^*)^* = f$.

**13.17** Prove that

$$f(x) = \ln \frac{\lambda}{2} - \lambda\sqrt{x}, \quad x \geq 0$$

is a convex function.

**13.18** Derive variational bounds for the logistic regression function

$$\sigma(x) = \frac{1}{1+e^{-x}},$$

one of them in terms of a Gaussian function. For the latter case, use the transformation, $t = \sqrt{x}$.

**13.19** Prove Eq. (13.99).

**13.20** Derive Eqs. (13.110) and (13.111).

**13.21** Derive Eq. (13.112) from Eq. (13.109).

### *MATLAB Exercises*

**13.22** Generate $N = 60$ data points from each of the five Gaussian distributions given in Example 13.1. Implement the EM algorithm to obtain estimates of the parameters of the Gaussian mixture model (Exercise 12.17). Run the EM algorithm on our generated data, assuming $K = 25$ clusters, using randomly chosen values for the initial mean values and the covariance matrices. Next, implement the variational Bayes algorithm that treats the same problem, according to the steps reported in Section 13.4. Plot the initial and final estimates of the EM and the variational Bayes algorithm to reproduce the results of Figure 13.4. Play with different values of the parameters.

**13.23** Generate a vector comprising $N = 100$ equidistant sampling points $x_n$ in the interval $[-10, 10]$. Compute $N$ basis functions, each one located at a sampling point $x_n$, of the form $\phi_n(x) = \exp\left(-(x-x_n)^2/2\sigma_\phi^2\right)$, where $\sigma_\phi^2 = 0.1$. Select two of the basis functions randomly to compute the output samples, $y_n$, according to the regression model of Example 13.2. The additive noise power should correspond to an SNR level of 6dB. Implement the EM algorithm expressed in Eqs. (12.70), (12.71), (12.78) and (12.79), in order to fit a (generalized) linear regression model comprising the $N$ basis functions to the generated data $y_n$. Also, implement the variational Bayes EM, summarized in Algorithm 13.1. Plot the reconstructed signals and compare the results.

**13.24** Generate $N = 150$ two-dimensional data points $\boldsymbol{x}_n$, uniformly distributed in the region $[-5, 5] \times [-5, 5]$. Assign a binary label to each $\boldsymbol{x}_n$, according to the graph of the function

$$f(x) = 0.5x^3 + 0.5x^2 + 0.5x + 1,$$

in the two-dimensional space. To generate the training data, each $\boldsymbol{x}_n = [x_{n1}, x_{n2}]^T$ is assigned to one of two classes $\omega_1, \omega_2$, depending on which side of the above graph the following quantity,

$$y_n = 0.5x_{n1}^3 + 0.5x_{n1}^2 + 0.5x_{n1} + 1 + \eta,$$

lies, where $\eta$ stands for zero-mean Gaussian noise of variance $\sigma_\eta^2 = 4$. In other words, the class assignment is done according to whether $y_n > f(x_{n1})$ or $y_n < f(x_{n1})$. Download and run the MATLAB code of the RVM classifier[15] for the generated data set. Use the Gaussian kernel with $\sigma^2 = 3$. Repeat the experiments with different values of $\sigma^2$. Plot and discuss the classification results.

**13.25** Consider an one-dimensional Gaussian process with zero mean and Gaussian (kernel) covariance function, with length scale $h = 0.5$

**(a)** Sample $D = 100$ equidistant input points in the interval $[-2, 2]$. Use these as input points to compute the covariance function of the GP and form the respective $100 \times 100$ covariance matrix. Use the corresponding multivariate Gaussian to generate samples for five different realizations and plot the results, as in Figure 13.16. Repeat the same experiment with different values for the parameter $h$.

**(b)** Now, sample $N = 20$ input points from a zero-mean, unit-variance normal distribution. Based on these input points, evaluate the covariance function and the respective $20 \times 20$ covariance matrix, as before. Then, generate noisy GP data, by first sampling $N$ points from our Gaussian process, and then adding zero-mean Gaussian noise with variance 0.1. Next, sample $D = 100$ points in the interval $[-3, 4]$. Compute the corresponding mean and the variance of the predictive GP, as given in Eq. (13.138). In a single figure, plot the observed data, the posterior mean, and the error bars of the predictive mean as in Figure 13.17.

**13.26** Download the Matlab code for the Chinese restaurant process mixture model from *http://sites.google.com/site/kenichikurihara/academic-software*. Generate two-dimensional data from the Gaussian mixture model of Example 13.5 and reproduce the results in Figure 13.15.

**13.27** Reproduce the hyperspectral unmixing results of Figure 13.21 by running the script "HSIvB.m," which is available at the website of the book.

## REFERENCES

[1] D. Aldous, Exchangeability and related topics, in: École d' Été de Probabilités de Saint-Flour XIII-1983, Lecture Notes in Mathematics, Springer, New York, 1985, pp. 1-198.

[2] S. Amari, Differential Geometrical Methods in Statistics, Springer, New York, 1985.

[3] H. Attias, Inferring parameters and structure of latent variable models by variational Bayes, in: K.B. Laskey, H. Prade (Eds.), Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence, Morgan-Kaufmann, San Mateo, 1999, pp. 21-30.

[4] S. Babacan, R. Molina, A. Katsaggelos, Fast Bayesian compressive sensing using Laplace priors, in: Proceedings International Conference on Acoustics, Speech and Signal Processing, ICASSP, Taipei, Taiwan, 2009.

[5] S.D. Babacan, L. Maniera, R. Molina, A. Katsaggelos, Non-convex priors in Bayesian compressive sensing, in: Proceedings, 17th European Signal Processing Conference, EURASIP, Glasgow, Scotland, 2009.

[6] M.J. Beal, Variational Algorithms for Approximate Bayesian Inference, Ph.D. Thesis, University College London, 2003.

[15] The RVM software can be found at http://www.miketipping.com/sparsebayes.htm.

[7] C. Bishop, M. Tipping, Variational relevance vector machines, in: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, 2000, pp. 46-53.

[8] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, 2006.

[9] D. Blei, M. Jordan, Variational inference for Dirichlet process mixtures, Bayesian Anal. 1 (1) (2006) 121-144.

[10] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, 2004.

[11] A. Ben-Israel, T.N.E. Greville, Variational Bayesian model selection for mixture distribution, in: T. Jaakula, T. Richardshon (Eds.), Artificial Intelligence and Statistics, Morgan-Kaufmann, San Mateo, 2001, pp. 27-34.

[12] N. Dobigeon, J.-Y. Tourneret, C.-I. Chang, Semi-supervised linear spectral unmixing using a hierarchical Bayesian model for hyperspectral imagery, IEEE Trans. Signal Process. 56 (7) (2008) 2684-2695.

[13] T. Ferguson, A Bayesian analysis of some nonparametric problems, Ann. Stat. 1 (2) (1973) 209-230.

[14] R.P. Feyman, A Set of Lectures, Perseus, Reading, MA, 1972.

[15] M.A.P. Figuerido, Adaptive sparseness for supervised learning, IEEE Trans. Pattern Anal. Mach. Learn. 25 (9) (2003) 1150-1159.

[16] J. Gershman, D.M. Blei, A tutorial on Bayesian nonparametric models, J. Math. Psychol. 56 (2012) 1-12.

[17] M.N. Gibs, D.J.C. MacKay, Variational Gaussian process classifiers, IEEE Trans. Neural Netw. 11 (6) (2000) 1458-1464.

[18] M. Girolami, A variational method for learning sparse and overcomplete representations, Neural Comput 13 (2001) 2517-2532.

[19] P. Green, S. Richardson, Modeling heterogeneity with and without the Dirichlet process, Scand. J. Stat. 28 (2) (2001) 355-375.

[20] L.K. Hansen, C.E. Rasmussen, Pruning from adaptive regularization, Neural Comput. 6 (1993) 1223-1232.

[21] G.E. Hinton, Van Camp. D., Keeping neural networks simple by minimizing the description length of weight, in: Proceedings 6th ACM Conference on Computing Learning, Santa Cruz, 1993.

[22] G.E. Hinton, D.S. Zemel, Autoencoders, minimum description length and Helmholtz free energy, in: J.D. Conan, G. Tesauro, J. Alspector (Eds.), Advances in Neural Information Processing System, vol. 6, Morgan-Kaufmann, San Mateo, 1999.

[23] N. Hjort, C. Holmes, P. Muller, S. Walker, Bayesian Nonparametrics, Cambridge University Press, Cambridge, 2010.

[24] M.D. Hoffman, M.D. Blei, C. Wang, J. Paisley, Stochastic variational inference, J. Mach. Learn. Res. 14 (2013) 1303-1347.

[25] H. Ishwaran, J.S. Rao, Spike and slab variable selection: Frequentist and Bayesian strategies, Ann. Stat. 33 (2) (2005) 730-773.

[26] M.D. Iordache, J.M. Bioucas-Dias, A. Plaza, Collaborative sparse regression for hyperspectral unmixing, IEEE Trans. Geosci. Remote Sens. 52 (1)(2014) 341-354.

[27] T.J. Jaakola, Variational methods for inference and estimation in graphical models, Ph.D. Thesis, Department of Brain and Cognitive Sciences, MIT, Cambridge, USA, 1997.

[28] T.J. Jaakola, M.I. Jordan, Improving the mean field approximation via the use of mixture distributions, in: M.I. Jordan (Ed.), Learning in Graphical Models, Kluwer, Dordrecht, 1998, pp. 163-173.

[29] T.J. Jaakola, M.I. Jordan, Bayesian logistic regression: a variational approach, Stat. Comput. 10 (2000) 25-37.

[30] S. Ji, Y. Xue, L. Carin, Bayesian compressive sensing, IEEE Trans. Signal Process. 56 (6) (2008) 2346-2356.

[31] M.I. Jordan, Z. Ghahramaniz, T.J. Jaakola, L.K. Saul, An introduction to variational methods in graphical models, Mach. Learn. 37 (1999) 183-233.

[32] N. Keshava, A survey of spectral unmixing algorithms, Lincoln Lab. J. 14 (1) (2003) 55-78.

[33] K. Kurihara, M. Welling, Y. Teh, Collapsed variational Dirichlet process mixture models, in: Proceedings of the International Joint Conference on Artificial Intelligence, vol. 20, 2007, pp. 2796-2801.

[34] M. Kuss, C. Rasmussen, Assessing approximations for Gaussian classification, in: Advances in Neural Information Processing Systems, vol. 18, MIT Press, Cambridge, MA, 2006.

[35] M. Lazaro-Gredilla, A. Figueiras-Vidal, Inter-domain Gaussian processes for sparse inference using inducing features, in: Advances in Neural Information Processing Systems, vol. 22, MIT Press, Cambridge, MA, 2010.

[36] M. Lazaro-Gredilla, Bayesian warped Gaussian processes, in: Advances in Neural Information Processing Systems, vol. 25, MIT Press, Cambridge, MA, 2013.

[37] F.B. Lempers, Posterior Probabilities of Alternative Linear Models, Rotterdam University Press, Rotterdam, 1971.

[38] D.J.C. McKay, Bayesian interpolation. Neural Comput. 4 (3) (1992) 417-447.

[39] D.J.C. MacKay, The evidence framework applied to classification networks, Neural Comput. 4 (1992) 720-736.

[40] D.J.C. MacKay, Information Theory, Inference and Learning Algorithms, Cambridge University Press, Cambridge, 2003.

[41] D.J.C. MacKay, Bayesian nonlinear modeling for the energy prediction competition, ASHRAE Trans. 100 (2) (1994) 1053-1062.

[42] T. Minka, Expectation propagation for approximate Bayesian inference, in: J. Breese, D. Koller (Eds.), Proceedings 17th Conference on Uncertainty in Artificial Intelligence, 2001, pp. 362-369.

[43] T. Minka, Divergence measures and message passing, Technical Report, Microsoft Research Laboratory, Cambridge, UK, 2005.

[44] T. Mitchell, J. Beauchamp, Bayesian variable selection in linear regression, J. Am. Stat. Assoc. 83 (1988) 1023-1036.

[45] J.M.P. Nascimento, J.M. Bioucas-Dias, Vertex component analysis: a fast algorithm to unmix hyperspectral data, IEEE Trans. Geosci. Remote Sens. 43 (4) (2005) 898-910.

[46] R.M. Neal, Bayesian learning for neural networks, in: Lecture Notes in Statistics, vol. 118, Springer-Verlag, New York, 1996.

[47] R.M. Neal, Monte Carlo implementation for Gaussian process models for Bayesian regression and classification, Technical Report CRG-TR-97-2, Department of Computer Science, University of Toronto, 1997.

[48] R.M. Neal, Assessing relevance determination methods using DELVE, in: C. Bishop (Ed.), Neural Networks and Machine Learning, Springer-Verlag, New York, 1998, pp. 97-120.

[49] R. Neal, Markov chain sampling methods for Dirichlet process mixture models, J. Comput. Graph. Stat. 9 (2) (2000) 249-265.

[50] A. O'Hagan, J.F. Kingman, Curve fitting and optimal design for prediction, J. R. Stat. Soc. B 40 (1) (1978) 1783-1816.

[51] M. Opper, O. Winther, A Bayesian approach to on-line learning, in: D. Saad (Ed.), On-line Learning in Neural Networks, Cambridge University Press, Cambridge, 1999, pp. 363-378.

[52] J. Palmer, D. Wipf, K. Krentz-Delgade, B. Rao, Variational EM algorithms for non-Gaussian latent variable models, in: Advances in Neural Information Systems, vol. 18, 2006, pp. 1059-1066.

[53] G. Parisi, Statistical Field Theory, Addison Wesley, New York, 1988.

[54] F. Perez-Cruz, S. Van Vaerenbergh, J.J. Murillo-Fuentes, M. Lazaro-Gredilla, I. Santamaria, Gaussian processes for nonlinear signal processing, IEEE Signal Process. Mag. 30 (4) (2013) 40-50.

[55] J. Pitman, Combinatorial stochastic processes, Technical report 621, Notes for Saint Flour Summer School, Department of Statistics, UC, Berkeley, 2002.

[56] P. Pal, P.P. Vaidyanathan, Parameter identifiability in sparse Bayesian learning, in: Proceedings International Conference on Acoustics, Speech and Signal Processing, ICASSP, Florence, Italy, 2014.

[57] J. Quionero-Candela, C.E. Rasmussen, A unifying view of sparse approximate Gaussian process regression, Mach. Learn. Res. 6 (2005) 1939-1959.

[58] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, MIT Press, Cambridge, MA, 2006.

[59] R. Rockaffelar, Convex Analysis, Princeton University Press, Princeton, NJ, 1970.

[60] G.A. Rodriguez-Yam, R.A. Davis, L.L. Scharf, A Bayesian model and Gibbs sampler for hyperspectral imaging, in: Proceedings, IEEE Sensor Array and Multichannel Signal Processing Workshop, 2002, pp. 105-109.

[61] S. Ryan, M. Lewis, Mapping soils using high resolution airborne imagery, Barossa Valley, SA, in: Proceedings of the Inaugural Australian Geospatial Information and Agriculture Conference Incorporating Precision Agriculture in Australasia 5th Annual Symposium, 2001, pp. 17-19.

[62] S. Sarkka, A. Solin, J. Hartikainen, Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing, IEEE Signal Process. Mag. 30 (4) (2013) 51-61.

[63] M.A. Sato, Online model selection based on the variational Bayes, Neural Comput. 13 (7) (2001) 1649-1681.

[64] M.N. Schmidt, M. Morup, Nonparametric Bayesian modeling of complex networks, IEEE Signal Process. Mag. 30 (3) (2013) 110-128.

[65] M.W. Seeger, H. Nickish, Large scale variational inference and experimental design for sparse generalized linear models, Technical report, # TR-175, Max Plank Institute für Biologische Kybernetic, 2008.

[66] M.W. Seeger, D.P. Wipf, Variational Bayesian inference techniques, IEEE Signal Process. Mag. 27 (1) (2010) 81-91.

[67] J. Sethuraman, A constructive definition of Dirichlet priors, Stat. Sin. 4 (2) (1994) 639-650.

[68] E. Snelson, C.E. Rasmussen, Z. Ghahramani, Warped Gaussian processes, in: Advances in Neural Information Processing Systems, vol. 16, MIT Press, Cambridge, MA, 2003.

[69] E. Snelson, Z. Ghahramani, Sparse Gaussian processes using pseudo-inputs, in: Advances in Neural Information Processing Systems, vol. 18, MIT Press, Cambridge, MA, 2006, pp. 1259-1266.

[70] C. Soussen, J. Idier, D. Brie, J. Duan, From Bernoulli-Gaussian deconvolution to sparse signal restoration, IEEE Trans. Signal Process. 59 (10) (2011) 4572-4584.

[71] K.E. Themelis, A.A. Rontogiannis, K.D. Koutroumbas, A novel hierarchical Bayesian approach for sparse semisupervised hyperspectral unmixing, IEEE Trans. Signal Process. 60 (2) (2012) 585-599.

[72] K.E. Themelis, A.A. Rontogiannis, K.D. Koutroumbas, Semisupervised hyperspectral image unmixing using a variational Bayes algorithm, 2014, http://arxiv.org/abs/1406.4705.

[73] K. Themelis, A. Rontogiannis, K. Koutroumbas, A variational Bayes framework for sparse adaptive estimation, IEEE Trans. Signal Process. (2014), http://dx.doi.org/10.1109/TSP.2014.2338839, to appear 2015.

[74] S. Theodoridis, K. Koutroumbas, Pattern Recognition, fourth ed., Academic Press, Boston, 2009.

[75] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, J. Mach. Learn. Res. 1 (2001) 211-244.

[76] M.E. Tipping, A.C. Faul, Fast marginal likelihood maximisation for sparse Bayesian models, in: C.M. Bishop, B.J. Frey (Eds.), Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, Key West, FL, 2003.

[77] M.K. Titsias, Variational learning of inducing variables in sparse Gaussian processes, in: Proceedings 12th International Workshop on Artificial Intelligence and Statistics, 2009, pp. 567-574.

[78] S. Van Vaerenbergh, M. Lazaro-Gredilla, I. Santamaria, Kernel recursive least-squares tracker for time–varying regression, IEEE Trans. Neural Netw. Learn. Syst. 23 (8) (2012) 1313-1326.

[79] C.K.I. Williams, D. Barber, Bayesian classification with Gaussian processes, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 1342-1351.

[80] D. Wipf, Bayesian methods for finding sparse representations, Ph.D. Thesis, University of California, San Diego, 2006.

[81] D.P. Wipf, B.D. Rao, An empirical Bayesian strategy for solving the simultaneous sparse approximation problem, IEEE Trans. Signal Process. 55 (7) (2007) 3704-3716.

[82] D. Wipf, S. Nagarajan, A new view of automatic relevance determination, in: Advances in Neural Information Systems (NIPS), vol. 20, 2008.

[83] D. Wipf, B. Rao, S. Nagarajan, Latent variable models for promoting sparsity, IEEE Trans. Informat. Theory 57(9) (2011) 6236-6255.

[84] D.P. Wipf, B.D. Rao, S. Nagarajan, Latent variable Bayesian methods for promoting sparsity, IEEE Trans. Inf. Theory 57 (9) (2011) 6236-6255.

[85] X. Zhang, B.D. Rao, Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning, IEEE Trans. Select. Areas Signal Process. 5 (5) (2011) 912-926.

[86] X. Zhang, B.D. Rao, Extension of SBL Algorithms for the recovery of block sparse signals with intra-block correlation, IEEE Trans. Signal Process. 61 (8) (2013) 2019-2015.