10

Maximum Likelihood Inference, Part I: Optimization Through Exact Smoothing

In previous chapters, we have focused on structural results and methods for HMMs, considering in particular that the models under consideration were always perfectly known. In most situations, however, the model cannot be fully specified beforehand, and some of its parameters need to be calibrated based on observed data. Except for very simplistic instances of HMMs, the structure of the model is sufficiently complex to prevent the use of direct estimators such as those provided by moment or least squares methods. We thus focus in the following on computation of the *maximum likelihood estimator*.

Given the specific structure of the likelihood function in HMMs, it turns out that the key ingredient of any optimization method applicable in this context is the ability to compute smoothed functionals of the unobserved sequence of states. Hence the methods discussed in the second part of the book for evaluating smoothed quantities are instrumental in devising parameter estimation strategies.

This chapter only covers the class of HMMs discussed in Chapter 5, for which the smoothing recursions described in Chapters 3 and 4 may effectively be implemented on computers. For such models, the likelihood function is computable, and hence our main task will be to optimize a possibly complex but entirely known function. The topic of this chapter thus relates to the more general field of numerical optimization. For models that do not allow for exact numerical computation of smoothing distributions, this chapter provides a framework from which numerical approximations can be built. Those will be discussed in Chapter 11.

10.1 Likelihood Optimization in Incomplete Data Models

To describe the methods as concisely as possible, we adopt a very general viewpoint in which we only assume that the likelihood function of interest may be written as the marginal of a higher dimensional function. In the terminology introduced by Dempster *et al.* (1977), this higher dimensional function is described as the *complete data* likelihood; in this framework, the term *incomplete data* refers to the actual observed data while the *complete data* is a (not fully observable) higher dimensional random variable. In Section 10.2, we will exploit the specific structure of the HMM, and in particular the fact that it corresponds to a *missing data model* in which the observations simply are a subset of the complete data. We ignore these specifics for the moment however and consider the general likelihood optimization problem in incomplete data models.

10.1.1 Problem Statement and Notations

Given a σ-finite measure λ on $(\mathsf{X}, \mathcal{X})$, we consider a family $\{f(\cdot;\theta)\}_{\theta\in\Theta}$ of non-negative λ-integrable functions on X. This family is indexed by a parameter $\theta \in \Theta$, where Θ is a subset of $\mathbb{R}^{d_\theta}$ (for some integer d_θ). The task under consideration is the maximization of the integral

$$\mathrm{L}(\theta) \stackrel{\text{def}}{=} \int f(x\,;\theta)\,\lambda(dx) \tag{10.1}$$

with respect to the parameter θ. The function $f(\cdot\,;\theta)$ may be thought of as an *unnormalized probability density* with respect to λ. Thus $\mathrm{L}(\theta)$ is the normalizing constant for $f(\cdot\,;\theta)$. In typical examples, $f(\cdot\,;\theta)$ is a relatively simple function of θ. In contrast, the quantity $\mathrm{L}(\theta)$ usually involves high-dimensional integration and is therefore sufficiently complex to prevent the use of simple maximization approaches; even the direct evaluation of the function might turn out to be non-feasible.

In Section 10.2, we shall consider more specifically the case where f is the joint probability density function of two random variables X and Y, the latter being observed while the former is not. Then X is referred to as the *missing data*, f is the *complete data likelihood*, and L is the density of Y alone, that is, the *likelihood* available for estimating θ. Note however that thus far, the dependence on Y is not made explicit in the notation; this is reminiscent of the implicit conditioning convention discussed in Section 3.1.4 in that the observations do not appear explicitly. Having sketched these statistical ideas, we stress that we feel it is actually easier to understand the basic mechanisms at work without relying on the probabilistic interpretation of the above quantities. In particular, it is not required that L be a likelihood, as any function satisfying (10.1) is a valid candidate for the methods discussed here (cf. Remark 10.2.1).

In the following, we will assume that $\mathrm{L}(\theta)$ is positive, and thus maximizing $\mathrm{L}(\theta)$ is equivalent to maximizing

$$\ell(\theta) \stackrel{\text{def}}{=} \log \mathrm{L}(\theta)\ . \tag{10.2}$$

In a statistical setting, ℓ is the *log-likelihood*. We also associate to each function $f(\cdot\,;\theta)$ the probability density function $p(\cdot\,;\theta)$ (with respect to the dominating measure λ) defined by

$$p(x\,;\theta) \stackrel{\text{def}}{=} f(x\,;\theta)/\mathrm{L}(\theta)\;. \tag{10.3}$$

In the statistical setting sketched above, $p(x;\theta)$ is the conditional density of X given Y.

10.1.2 The Expectation-Maximization Algorithm

The most popular method for solving the general optimization problem outlined above is the EM (for *expectation-maximization*) algorithm introduced, in its full generality, by Dempster *et al.* (1977) in their landmark paper. Given the literature available on the topic, our aim is not to provide a comprehensive review of all the results related to the EM algorithm but rather to highlight some of its key features and properties in the context of hidden Markov models.

10.1.2.1 The Intermediate Quantity of EM

The central concept in the framework introduced by Dempster *et al.* (1977) is an auxiliary function (or, more precisely, a family of auxiliary functions) known as the intermediate quantity of EM.

Definition 10.1.1 (Intermediate Quantity of EM). *The* intermediate quantity of EM *is the family $\{\mathcal{Q}(\cdot\,;\theta')\}_{\theta'\in\Theta}$ of real-valued functions on Θ, indexed by θ' and defined by*

$$\mathcal{Q}(\theta\,;\theta') \stackrel{\text{def}}{=} \int \log f(x\,;\theta) p(x\,;\theta')\,\lambda(dx)\;. \tag{10.4}$$

Remark 10.1.2. To ensure that $\mathcal{Q}(\theta\,;\theta')$ is indeed well-defined for all values of the pair (θ,θ'), one needs regularity conditions on the family of functions $\{f(\cdot\,;\theta)\}_{\theta\in\Theta}$, which will be stated below (Assumption 10.1.3). To avoid trivial cases however, we use the convention $0\log 0 = 0$ in (10.4) and in similar relations below. In more formal terms, for every measurable set N such that *both* $f(x\,;\theta)$ and $p(x\,;\theta')$ vanish λ-a.e. on N, set

$$\int_N \log f(x\,;\theta) p(x\,;\theta')\,\lambda(dx) \stackrel{\text{def}}{=} 0\;.$$

With this convention, $\mathcal{Q}(\theta\,;\theta')$ stays well-defined in cases where there exists a non-empty set N such that *both* $f(x\,;\theta)$ and $f(x\,;\theta')$ vanish λ-a.e. on N. ■

The intermediate quantity $\mathcal{Q}(\theta\,;\theta')$ of EM may be interpreted as the expectation of the function $\log f(X\,;\theta)$ when X is distributed according to the probability density function $p(\cdot\,;\theta')$ indexed by a, possibly different, value θ' of the parameter. Using (10.2) and (10.3), one may rewrite the intermediate quantity of EM in (10.4) as

$$\mathcal{Q}(\theta\,;\theta') = \ell(\theta) - \mathcal{H}(\theta\,;\theta')\;, \tag{10.5}$$

where

$$\mathcal{H}(\theta\,;\theta') \stackrel{\text{def}}{=} -\int \log p(x\,;\theta) p(x\,;\theta')\,\lambda(dx)\;. \tag{10.6}$$

Equation (10.5) states that the intermediate quantity $\mathcal{Q}(\theta\,;\theta')$ of EM differs from (the log of) the objective function $\ell(\theta)$ by a quantity that has a familiar form. Indeed, $\mathcal{H}(\theta'\,;\theta')$ is recognized as the *entropy* of the probability density function $p(\cdot\,;\theta')$ (see for instance Cover and Thomas, 1991). More importantly, the increment of $\mathcal{H}(\theta\,;\theta')$,

$$\mathcal{H}(\theta\,;\theta') - \mathcal{H}(\theta'\,;\theta') = -\int \log \frac{p(x\,;\theta)}{p(x\,;\theta')} p(x\,;\theta')\,\lambda(dx)\;, \tag{10.7}$$

is recognized as the *Kullback-Leibler divergence* (or *relative entropy*) between the probability density functions p indexed by θ and θ', respectively.

The last piece of notation needed is the following: the gradient and Hessian of a function, say L, at θ' will be denoted by $\nabla_\theta \mathrm{L}(\theta')$ and $\nabla^2_\theta \mathrm{L}(\theta')$, respectively. To avoid ambiguities, the gradient of $\mathcal{H}(\cdot\,;\theta')$ with respect to its first argument, evaluated at θ'', will be denoted by $\nabla_\theta \mathcal{H}(\theta\,;\theta')|_{\theta=\theta''}$ (where the same convention will also be used, if needed, for the Hessian).

We conclude this introductory section by stating a minimal set of assumptions that guarantee that all quantities introduced so far are indeed well-defined.

Assumption 10.1.3.

(i) The parameter set Θ is an open subset of $\mathbb{R}^{d_\theta}$ (for some integer d_θ).
(ii) For any $\theta \in \Theta$, $\mathrm{L}(\theta)$ is positive and finite.
(iii) For any $(\theta, \theta') \in \Theta \times \Theta$, $\int |\nabla_\theta \log p(x\,;\theta)| p(x\,;\theta')\,\lambda(dx)$ is finite.

.

Assumption 10.1.3(iii) implies in particular that the probability distributions in the family $\{p(\cdot\,;\theta)\,d\lambda\}_{\theta\in\Theta}$ are all absolutely continuous with respect to one another. Any individual distribution $p(\cdot\,;\theta)\,d\lambda$ can only vanish on sets that are assigned null probability by all other probability distributions in the family. Thus both $\mathcal{H}(\theta\,;\theta')$ and $\mathcal{Q}(\theta\,;\theta')$ are well-defined for all pairs of parameters.

10.1.2.2 The Fundamental Inequality of EM

We are now ready to state the fundamental result that justifies the standard construction of the EM algorithm.

Proposition 10.1.4. *Under Assumption 10.1.3, for any $(\theta, \theta') \in \Theta \times \Theta$,*

$$\ell(\theta) - \ell(\theta') \geq \mathcal{Q}(\theta\,;\theta') - \mathcal{Q}(\theta'\,;\theta')\;, \tag{10.8}$$

where the inequality is strict unless $p(\cdot\,;\theta)$ and $p(\cdot\,;\theta')$ are equal λ-a.e.

Assume in addition that

(a) $\theta \mapsto \mathrm{L}(\theta)$ *is continuously differentiable on* Θ*;*
(b) for any $\theta' \in \Theta$, $\theta \mapsto \mathcal{H}(\theta\,;\theta')$ *is continuously differentiable on* Θ*.*

Then for any $\theta' \in \Theta$*,* $\theta \mapsto \mathcal{Q}(\theta\,;\theta')$ *is continuously differentiable on* Θ *and*

$$\nabla_\theta \ell(\theta') = \nabla_\theta \mathcal{Q}(\theta\,;\theta')|_{\theta=\theta'} \ . \tag{10.9}$$

Proof. The difference between the left-hand side and the right-hand side of (10.8) is the quantity defined in (10.7), which we already recognized as a Kullback-Leibler distance. Under Assumption 10.1.3(iii), this latter term is well-defined and known to be strictly positive (by direct application of Jensen's inequality) unless $p(\cdot\,;\theta)$ and $p(\cdot\,;\theta')$ are equal λ-a.e. (Cover and Thomas, 1991; Lehmann and Casella, 1998).

For (10.9), first note that $\mathcal{Q}(\theta\,;\theta')$ is a differentiable function of θ, as it is the difference of two functions that are differentiable under the additional assumptions (a) and (b). Next, the previous discussion implies that $\mathcal{H}(\theta\,;\theta')$ is maximal for $\theta = \theta'$, although this may not be the only point where the maximum is achieved. Thus its gradient vanishes at θ', which proves (10.9).

□

10.1.2.3 The EM Algorithm

The essence of the EM algorithm, which is suggested by (10.5), is that $\mathcal{Q}(\theta\,;\theta')$ may be used as a surrogate for $\ell(\theta)$. Both functions are not necessarily comparable but, in view of (10.8), any value of θ such that $\mathcal{Q}(\theta\,;\theta')$ is increased over its baseline $\mathcal{Q}(\theta'\,;\theta')$ corresponds to an increase of ℓ (relative to $\ell(\theta')$) that is at least as large.

The EM algorithm as proposed by Dempster *et al.* (1977) consists in iteratively building a sequence $\{\theta^i\}_{i\geq 1}$ of parameter estimates given an initial guess θ^0. Each iteration is classically broken into two steps as follows.

E-Step: Determine $\mathcal{Q}(\theta\,;\theta^i)$;
M-Step: Choose θ^{i+1} to be the (or any, if there are several) value of $\theta \in \Theta$ that maximizes $\mathcal{Q}(\theta\,;\theta^i)$.

It is certainly not obvious at this point that the M-step may be in practice easier to perform than the direct maximization of the function of interest ℓ itself. We shall return to this point in Section 10.1.2.4 below.

Proposition 10.1.4 provides the two decisive arguments behind the EM algorithm. First, an immediate consequence of (10.8) is that, by the very definition of the sequence $\{\theta^i\}$, the sequence $\{\ell(\theta^i)\}_{i\geq 0}$ of log-likelihood values is non-decreasing. Hence EM is a monotone optimization algorithm. Second, if the iterations ever stop at a point $\theta_\star$, then $\mathcal{Q}(\theta\,;\theta_\star)$ has to be maximal at $\theta_\star$ (otherwise it would still be possible to improve over $\theta_\star$), and hence $\theta_\star$ is such that $\nabla_\theta \mathrm{L}(\theta_\star) = 0$, that is, this is a *stationary point of the likelihood.*

Although this picture is largely correct, there is a slight flaw in the second half of the above intuitive reasoning in that the if part *(if the iterations ever* *stop at a point)* may indeed never happen. Stronger conditions are required to ensure that the sequence of parameter estimates produced by EM from any starting point indeed converges to a limit $\theta_\star \in \Theta$. However, it is actually true that when convergence to a point takes place, the limit has to be a stationary point of the likelihood. In order not to interrupt our presentation of the EM framework, convergence results pertaining to the EM algorithm are deferred to Section 10.5 at the end of this chapter; see in particular Theorems 10.5.3 and 10.5.4.

10.1.2.4 EM in Exponential Families

The EM algorithm defined in the previous section will only be helpful in situations where the following general conditions hold.

E-Step: It is possible to compute, at reasonable computational cost, the intermediate quantity $\mathcal{Q}(\theta\,;\theta')$ given a value of θ'.

M-Step: $\mathcal{Q}(\theta\,;\theta')$, considered as a function of its first argument θ, is sufficiently simple to allow closed-form maximization.

A rather general context in which both of these requirements are satisfied, or at least are equivalent to easily interpretable necessary conditions, is when the functions $\{f(\cdot\,;\theta)\}$ belong to an *exponential family.*

Definition 10.1.5 (Exponential Family). *The family* $\{f(\cdot\,;\theta)\}_{\theta\in\Theta}$ *defines an* exponential family *of positive functions on* X *if*

$$f(x\,;\theta) = \exp\{\psi(\theta)^t S(x) - c(\theta)\}h(x) \; , \tag{10.10}$$

where S *and* ψ *are vector-valued functions (of the same dimension) on* X *and* Θ *respectively,* c *is a real-valued function on* Θ *and* h *is a non-negative real-valued function on* X.

Here $S(x)$ is known as the vector of *natural sufficient statistics*, and $\eta = \psi(\theta)$ is the *natural parameterization.* If $\{f(\cdot\,;\theta)\}_{\theta\in\Theta}$ is an exponential family and if $\int |S(x)|f(x\,;\theta)\,\lambda(dx)$ is finite for any $\theta \in \Theta$, the intermediate quantity of EM reduces to

$$\mathcal{Q}(\theta\,;\theta') = \psi(\theta)^t \left[\int S(x)p(x\,;\theta')\,\lambda(dx)\right] - c(\theta) + \int p(x\,;\theta')\log h(x)\,\lambda(dx) \; . \tag{10.11}$$

Note that the right-most term does not depend on θ and thus plays no role in the maximization. It may as well be ignored, and in practice it is not required to compute it. Except for this term, the right-hand side of (10.11) has an explicit form as soon as it is possible to evaluate the expectation of the vector of sufficient statistics S under $p(\cdot\,;\theta')$. The other important feature of (10.11), ignoring the rightmost term, is that $\mathcal{Q}(\theta\,;\theta')$, viewed as a function of θ, is similar to the logarithm of (10.10) for the particular value $S_{\theta'} = \int S(x)p(x\,;\theta')\,\lambda(dx)$ of the sufficient statistic.

In summary, if $\{f(\cdot\,;\theta)\}_{\theta\in\Theta}$ is an exponential family, the two above general conditions needed for the EM algorithm to be practicable reduce to the following.

E-Step: The expectation of the vector of sufficient statistics $S(X)$ under $p(\cdot\,;\theta')$ must be computable.

M-Step: Maximization of $\psi(\theta)^t s - c(\theta)$ with respect to $\theta \in \Theta$ must be feasible in closed form for any s in the convex hull of $S(\mathsf{X})$ (that is, for any valid value of the expected vector of sufficient statistics).

For the sake of completeness, it should be mentioned that there are variants of the EM algorithm that are handy in cases where the maximization required in the M-step is not directly feasible (see Section 10.5.3 and further references in Section 10.5.4). In the context of HMMs, the main limitation of the EM algorithm rather appears in cases where the E-step is not feasible. This latter situation is the rule rather than the exception in models for which the state space X is not finite. For such cases, approaches that build on the EM concepts introduced in the current chapter will be fully discussed in Chapter 11.

10.1.3 Gradient-based Methods

A frequently ignored observation is that in any model where the EM strategy may be applied, it is also possible to evaluate derivatives of the objective function $\ell(\theta)$ with respect to the parameter θ. This is obvious from (10.9), and we will expand on this matter below. As a consequence, instead of resorting to a specific algorithm such as EM, one may borrow tools from the (comprehensive and well-documented) toolbox of gradient-based optimization methods.

10.1.3.1 Computing Derivatives in Incomplete Data Models

A first remark is that in cases where the EM algorithm is applicable, the objective function $\ell(\theta)$ is actually computable: because the EM requires the computation of expectations under the conditional density $p(\cdot\,;\theta)$, it is restricted to cases where the normalizing constant $\mathrm{L}(\theta)$—and hence $\ell(\theta) = \log \mathrm{L}(\theta)$—is available. The two equalities below show that it is indeed also the case for the first- and second-order derivatives of $\ell(\theta)$.

Proposition 10.1.6 (Fisher's and Louis' Identities). *Assume 10.1.3 and that the following conditions hold.*

(a) $\theta \mapsto \mathrm{L}(\theta)$ *is twice continuously differentiable on* Θ.

(b) For any $\theta' \in \Theta$, $\theta \mapsto \mathcal{H}(\theta\,;\theta')$ *is twice continuously differentiable on* Θ. *In addition,* $\int |\nabla_\theta^k \log p(x\,;\theta)| p(x\,;\theta')\,\lambda(dx)$ *is finite for* $k = 1, 2$ *and any* $(\theta,\theta') \in \Theta \times \Theta$, *and*

$$\nabla_\theta^k \int \log p(x\,;\theta) p(x\,;\theta')\,\lambda(dx) = \int \nabla_\theta^k \log p(x\,;\theta) p(x\,;\theta')\,\lambda(dx)\ .$$

Then the following identities hold:

$$\nabla_\theta \ell(\theta') = \int \nabla_\theta \log f(x\,;\theta)|_{\theta=\theta'}\, p(x\,;\theta')\,\lambda(dx)\;, \tag{10.12}$$

$$- \nabla^2_\theta \ell(\theta') = - \int \nabla^2_\theta \log f(x\,;\theta)\big|_{\theta=\theta'}\, p(x\,;\theta')\,\lambda(dx) \\ + \int \nabla^2_\theta \log p(x\,;\theta)\big|_{\theta=\theta'}\, p(x\,;\theta')\,\lambda(dx)\;. \tag{10.13}$$

The second equality may be rewritten in the equivalent form

$$\nabla^2_\theta \ell(\theta') + \{\nabla_\theta \ell(\theta')\}\,\{\nabla_\theta \ell(\theta')\}^t = \int \Big[\nabla^2_\theta \log f(x\,;\theta)\big|_{\theta=\theta'} \\ + \{\,\nabla_\theta \log f(x\,;\theta)|_{\theta=\theta'}\}\,\{\,\nabla_\theta \log f(x\,;\theta)|_{\theta=\theta'}\}^t \Big] p(x\,;\theta')\,\lambda(dx)\;. \tag{10.14}$$

Equation (10.12) is sometimes referred to as *Fisher's identity* (see the comment by B. Efron in the discussion of Dempster *et al.*, 1977, p. 29). In cases where the function L may be interpreted as the likelihood associated with some statistical model, the left-hand side of (10.12) is the *score function* (gradient of the log-likelihood). Equation (10.12) shows that the score function may be evaluated by computing the expectation, under $p(\cdot\,;\theta')$, of the function $\nabla_\theta \log f(X\,;\theta)|_{\theta=\theta'}$. This latter quantity, in turn, is referred to as the *complete score function* in a statistical context, as $\log f(x\,;\theta)$ is the joint log-likelihood of the complete data (X, Y); again we remark that at this stage, Y is not explicit in the notation.

Equation (10.13) is usually called the *missing information principle* after Louis (1982) who first named it this way, although it was mentioned previously in a slightly different form by Orchard and Woodbury (1972) and implicitly used in Dempster *et al.* (1977). In cases where L is a likelihood, the left-hand side of (10.13) is the associated *observed information matrix*, and the second term on the right-hand side is easily recognized as the (negative of the) Fisher information matrix associated with the probability density function $p(\cdot\,;\theta')$.

Finally (10.14), which is here written in a form that highlights its symmetry, was also proved by Louis (1982) and is thus known as *Louis' identity.* Together with (10.12), it shows that the first- and second-order derivatives of ℓ may be evaluated by computing expectations under $p(\cdot\,;\theta')$ of quantities derived from $f(\cdot\,;\theta)$. We now prove these three identities.

Proof (of Proposition 10.1.6). Equations (10.12) and (10.13) are just (10.5) where the right-hand side is differentiated once, using (10.9), and then twice under the integral sign.

To prove (10.14), we start from (10.13) and note that the second term on its right-hand side is the negative of an information matrix for the parameter

θ associated with the probability density function $p(\cdot\,;\theta)$ and evaluated at θ'. We rewrite this second term using the well-known information matrix identity

$$\int \nabla_\theta^2 \log p(x\,;\theta)\big|_{\theta=\theta'} p(x\,;\theta')\,\lambda(dx) \\ = -\int \{\nabla_\theta \log p(x\,;\theta)|_{\theta=\theta'}\}\{\nabla_\theta \log p(x\,;\theta)|_{\theta=\theta'}\}^t p(x\,;\theta')\,\lambda(dx)\;.$$

This is again a consequence of assumption (b) and the fact that $p(\cdot\,;\theta)$ is a probability density function for all values of θ, implying that

$$\int \nabla_\theta \log p(x\,;\theta)|_{\theta=\theta'} p(x\,;\theta')\,\lambda(dx) = 0\;.$$

Now use the identity $\log p(x\,;\theta) = \log f(x\,;\theta) - \ell(\theta)$ and (10.12) to conclude that

$$\int \{\nabla_\theta \log p(x\,;\theta)|_{\theta=\theta'}\}\{\nabla_\theta \log p(x\,;\theta)|_{\theta=\theta'}\}^t p(x\,;\theta')\,\lambda(dx) \\ = \int \{\nabla_\theta \log f(x\,;\theta)|_{\theta=\theta'}\}\{\nabla_\theta \log f(x\,;\theta)|_{\theta=\theta'}\}^t p(x\,;\theta')\,\lambda(dx) \\ - \{\nabla_\theta \ell(\theta')\}\{\nabla_\theta \ell(\theta')\}^t\;,$$

which completes the proof. □

Remark 10.1.7. As was the case for the intermediate quantity of EM, Fisher's and Louis' identities only involve expectations under $p(\cdot\,;\theta')$ of quantities derived from $f(\cdot\,;\theta)$. In particular, when the functions $f(\cdot\,;\theta)$ belong to an exponential family (see Definition 10.1.5), Fisher's identity, for instance, may be rewritten as

$$\nabla_\theta \ell(\theta') = \{\nabla_\theta \psi(\theta')\}^t \left(\int S(x) p(x\,;\theta')\,\lambda(dx)\right) - \nabla_\theta c(\theta')\;,$$

with the convention that $\nabla_\theta \psi(\theta')$ is the $d_\theta \times d_\theta$ matrix containing the partial derivatives $[\nabla_\theta \psi(\theta')]_{ij} = \partial \psi_i(\theta')/\partial \theta_j$. As a consequence, the only practical requirement for using Fisher's and Louis' identities is the ability to compute expectations of the sufficient statistic $S(x)$ under $p(\cdot\,;\theta)$ for any $\theta \in \Theta$. ■

10.1.3.2 The Steepest Ascent Algorithm

We briefly discuss the main features of gradient-based iterative optimization algorithms, starting with the simplest, but certainly not most efficient, approach. We restrict ourselves to the case where the optimization problem is *unconstrained* in the sense that $\Theta = \mathbb{R}^{d_\theta}$, so that any parameter value produced by the algorithms below is valid. For an in-depth coverage of the subject, we recommend the monographs by Luenberger (1984) and Fletcher (1987).

The simplest method is the *steepest ascent* algorithm in which the current value of the estimate θ^i is updated by adding a multiple of the gradient $\nabla_\theta \ell(\theta^i)$, referred to as the *search direction*:

$$\theta^{i+1} = \theta^i + \gamma_i \nabla_\theta \ell(\theta^i) \; . \tag{10.15}$$

Here the multiplier γ_i is a non-negative scalar that needs to be adjusted at each iteration to ensure, *a minima*, that the sequence $\{\ell(\theta^i)\}$ is non-decreasing—as was the case for EM. The most sensible approach consists in choosing γ_i as to maximize the objective function in the search direction:

$$\gamma_i = \arg\max\nolimits_{\gamma \geq 0} \ell[\theta^i + \gamma \nabla_\theta \ell(\theta^i)] \; . \tag{10.16}$$

It can be shown (Luenberger, 1984, Chapter 7) that under mild assumptions, the steepest ascent method with multipliers (10.16) is globally convergent, with a set of limit points corresponding to the stationary points of ℓ (see Section 10.5 for precise definitions of these terms and a proof that this property holds for the EM algorithm).

It remains that the use of the steepest ascent algorithm is not recommended, particularly in large-dimensional parameter spaces. The reason for this is that its speed of convergence *linear* in the sense that if the sequence $\{\theta^i\}_{i\geq 0}$ converges to a point $\theta_\star$ such that the Hessian $\nabla^2_\theta \ell(\theta_\star)$ is negative definite (see Section 10.5.2), then

$$\lim_{i\to\infty} \frac{\left|\theta^{i+1}(k) - \theta_\star(k)\right|}{\left|\theta^{i}(k) - \theta_\star(k)\right|} = \rho_k < 1 \; ; \tag{10.17}$$

here $\theta(k)$ denotes the kth coordinate of the parameter vector. For large-dimensional problems it frequently occurs that, at least for some components k, the factor ρ_k is close to one, resulting in very slow convergence of the algorithm. It should be stressed however that the same is true for the EM algorithm, which also exhibits speed of convergence that is linear, and often very poor (Dempster *et al.*, 1977; Jamshidian and Jennrich, 1997; Meng, 1994; Lange, 1995; Meng and Dyk, 1997). For gradient-based methods however, there exists a whole range of approaches, based on the second-order properties of the objective function, to guarantee faster convergence.

10.1.3.3 Newton and Second-order Methods

The prototype of second-order methods is the Newton, or Newton-Raphson, algorithm:

$$\theta^{i+1} = \theta^i - H^{-1}(\theta^i)\nabla_\theta \ell(\theta^i) \; , \tag{10.18}$$

where $H(\theta^i) = \nabla^2_\theta \ell(\theta^i)$ is the Hessian of the objective function. The Newton iteration is based on the second-order approximation

$$\ell(\theta) \approx \ell(\theta') + \nabla \ell(\theta')\,(\theta - \theta') + \frac{1}{2}\,(\theta - \theta')^t\, H(\theta')\,(\theta - \theta') \; .$$

If the sequence $\{\theta^i\}_{i\geq 0}$ produced by the algorithm converges to a point $\theta_\star$ at which the Hessian is negative definite, the convergence is, at least, quadratic in the sense that for sufficiently large i there exists a positive constant β such that $\|\theta^{i+1} - \theta_\star\| \leq \beta\|\theta^i - \theta_\star\|^2$. Therefore the procedure can be very efficient.

The practical use of the Newton algorithm is however hindered by two serious difficulties. The first is analogous to the problem already encountered for the steepest ascent method: there is no guarantee that the algorithm meets the minimal requirement to provide a final parameter estimate that is at least as good as the starting point θ^0. To overcome this difficulty, one may proceed as for the steepest ascent method and introduce a multiplier γ_i controlling the step-length in the search direction, so that the method takes the form

$$\theta^{i+1} = \theta^i - \gamma_i H^{-1}(\theta^i)\nabla_\theta \ell(\theta^i) \ . \tag{10.19}$$

Again, γ_i may be set to maximize $\ell(\theta^{i+1})$. In practice, it is most often impossible to obtain the exact maximum point called for by the ideal line-search, and one uses approximate directional maximization procedures. Generally speaking, a *line-search algorithm* is an algorithm to find a reasonable multiplier γ_i in a step of the form (10.19). A frequently used algorithm consists in determining the (approximate) maximum based on a polynomial interpolation of $\ell(\theta)$ along the line-segment between the current point θ^i and the proposed update given by (10.18).

A more serious problem is that except in the particular case where the function $\ell(\theta)$ is strictly concave, the direct implementation of (10.18) is prone to numerical instabilities: there may well be whole regions of the parameter space where the Hessian $H(\theta)$ is either non-invertible (or at least very badly conditioned) or not negative semi-definite (in which case $-H^{-1}(\theta^i)\nabla_\theta \ell(\theta^i)$ is not necessarily an ascent direction). To combat this difficulty, Quasi-Newton methods[1] use the modified recursion

$$\theta^{i+1} = \theta^i + \gamma_i W^i \nabla \ell(\theta^i) \ ; \tag{10.20}$$

here W^i is a weight matrix that may be tuned at each iteration, just like the multiplier γ_i. The rationale is that if W^i becomes close to $-H^{-1}(\theta^i)$ when convergence occurs, the modified algorithm will share the favorable convergence properties of the Newton algorithm. On the other hand, by using a weight matrix W^i different from $-H^{-1}(\theta^i)$, numerical issues associated with the matrix inversion may be avoided. We again refer to Luenberger (1984) and Fletcher (1987) for a more precise discussion of the available approaches and simply mention here the fact that usually the methods only take profit of gradient information to construct W^i, for instance using finite difference calculations, without requiring the direct evaluation of the Hessian $H(\theta)$.

In some contexts, it may be possible to build explicit strategies that are not as good as the Newton algorithm—failing in particular to reach quadratic

[1] *Conjugate gradient* methods are another alternative approach that we do not discuss here.

convergence rates—but yet significantly faster at converging than the basic steepest ascent approach. For incomplete data models, Lange (1995) suggested to use in (10.20) a weight matrix $I_c^{-1}(\theta^i)$ given by

$$I_c(\theta') = -\int \nabla_\theta^2 \log f(x\,;\theta)\big|_{\theta=\theta'} p(x\,;\theta')\,\lambda(dx)\;. \tag{10.21}$$

This is the first term on the right-hand side of (10.13). In many models of interest, this matrix is positive definite for all $\theta' \in \Theta$, and thus its inversion is not subject to numerical instabilities. Based on (10.13), it is also to be expected that in some circumstances, $I_c(\theta')$ is a reasonable approximation to the Hessian $\nabla_\theta^2 \ell(\theta')$ and hence that the weighted gradient algorithm converges faster than the steepest ascent or EM algorithms (see Lange, 1995, for further results and examples). In a statistical context, where $f(x\,;\theta)$ is the joint density of two random variables X and Y, $I_c(\theta')$ is the conditional expectation given Y of the observed information matrix of associated with this pair.

10.1.4 Pros and Cons of Gradient-based Methods

A quick search through the literature shows that for HMMs in particular and incomplete data models in general, the EM algorithm is much more popular than are gradient-based alternatives. A first obvious reason for this is that the EM approach is more generally known than its gradient-based counterparts. We list below a number of additional significant differences between both approaches, giving first the arguments in favor of the EM algorithm.

- *The EM algorithm is usually very simple to implement from scratch.* This is not the case for gradient-based methods, which require several specialized routines, for Hessian approximation, line-search, etc. This argument is however made less pregnant by the wide availability of generic numerical optimization code, so that implementing a gradient-based method usually only requires the computation of the objective function ℓ and its gradient. In most situations, this is not more complicated than is implementing EM.
- *The EM algorithm often deals with parameter constraints implicitly.* It is generally the case that the M-step equations are so simple that they can be solved even for parameters that are subject to constraints (see the case of normal HMMs in Section 10.3 for an example). For gradient-based methods this is not the case, and parameter constraints have to be dealt with explicitly, either through reparameterization (see Example 10.3.2) or using constrained optimization routines.
- *The EM algorithm is parameterization independent.* Because the M-step is defined by a maximization operation, it is independent of the way the parameters are represented, as is the maximum likelihood estimator for instance. Thus any (invertible) transformation of the parameter vector θ leaves the EM recursion unchanged. This is obviously not the case for gradient-based methods for which reparameterization will change the gradient and Hessian, and hence the convergence behavior of the algorithm.

In contrast, gradient-based methods may be preferred for the following reasons.

- *Gradient-based methods do not require the M-step.* Thus they may be applied to models for which the M-step does not lead to simple closed-form solutions.
- *Gradient-based methods converge faster.* As discussed above, gradient-based methods can reach quadratic convergence whereas EM usually converges only linearly, following (10.17)—see Example 10.3.2 for an illustration and further discussion of this aspect.

10.2 Application to HMMs

We now return to our primary focus and discuss the application of the previous methods to the specific case of hidden Markov models.

10.2.1 Hidden Markov Models as Missing Data Models

HMMs correspond to a sub-category of incomplete data models known as missing data models. In missing data models, the observed data Y is a subset of some not fully observable *complete data* (X, Y). We here assume that the joint distribution of X and Y, for a given parameter value θ, admits a joint probability density function $f(x, y\,; \theta)$ with respect to the product measure $\lambda \otimes \mu$. As mentioned in Section 10.1.1, the function f is sometimes referred to as the *complete data likelihood.* It is important to understand that f is a probability density function only when considered as a function of both x and y. For a fixed value of y and considered as a function of x only, f is a positive integrable function. Indeed, the actual *likelihood* of the observation, which is defined as the probability density function of Y with respect to μ, is obtained by marginalization as

$$\mathrm{L}(y\,; \theta) = \int f(x, y\,; \theta)\, \lambda(dx)\ . \tag{10.22}$$

For a given value of y this is of course a particular case of (10.1), which served as the basis for developing the EM framework in Section 10.1.2. In missing data models, the family of probability density functions $\{p(\cdot\,; \theta)\}_{\theta \in \Theta}$ defined in (10.3) may thus be interpreted as

$$p(x|y\,; \theta) = \frac{f(x, y\,; \theta)}{\int f(x, y\,; \theta)\, \lambda(dx)}\ , \tag{10.23}$$

the conditional probability density function of X given Y.

In the last paragraph, slightly modified versions of the notations introduced in (10.1) and (10.3) were used to reflect the fact that the quantities of interest now depend on the observed variable Y. This is obviously mostly a change regarding terminology, with no impact on the contents of Section 10.1.2, except that we may now think of integrating with respect to $p(\cdot\,;\theta)\,d\lambda$ as taking the conditional expectation with respect to the *missing data* X, given the observed data Y, in the model indexed by the parameter value θ.

Remark 10.2.1. Applying the EM algorithm defined in Section 10.1.2 in the case of (10.22) yields a sequence of parameter values $\{\theta^i\}_{i\geq 0}$ whose likelihoods $\mathrm{L}(y\,;\theta^i)$ cannot decrease with the iteration index i. Obviously, this connects to *maximum likelihood estimation*. Another frequent use of the EM algorithm is for *maximum a posteriori (MAP)* estimation, in which the objective function to be maximized is a *Bayesian posterior* (Dempster *et al.*, 1977). Indeed, we may replace (10.22) by

$$\mathrm{L}(y\,;\theta) = \pi(\theta) \int f(x, y\,;\theta)\,\lambda(dx)\ , \tag{10.24}$$

where π is a positive function on Θ. In the Bayesian framework (see Section 13.1 for a brief presentation of the Bayesian approach), π is usually selected to be a probability density function (with respect to some measure on Θ) and (10.24) is then interpreted as being proportional, up to a factor that depends on y only, to the *posterior* probability density function of the unknown parameter θ, conditional on the observation Y. In that case, π is referred to as a *prior density* on the parameter θ. But π in (10.24) may also be thought of as a *regularization functional* (sometimes also called a penalty) that may not have a probabilistic interpretation (Green, 1990).

Whether L is defined according to (10.22) or to (10.24) does not modify the definition of $p(\cdot\,;\theta)$ in (10.23), as the factor $\pi(\theta)$ cancels out in the renormalization. Thus the E-step in the EM algorithm is left unchanged and only the M-step depends on the precise choice of π. ■

10.2.2 EM in HMMs

We now consider more specifically hidden Markov models using the notations introduced in Section 2.2, assuming that observations Y_0 to Y_n (or, in short, $Y_{0:n}$) are available. Because we only consider HMMs that are fully dominated in the sense of Definition 2.2.3, we will use the notations ν and $\phi_{k|n}$ to refer to the probability density functions of these distributions (of X_0 and of X_k given $Y_{0:n}$) with respect to the dominating measure λ. The joint probability density function of the hidden states $X_{0:n}$ and associated observations $Y_{0:n}$, with respect to the product measure $\lambda^{\otimes(n+1)} \otimes \mu^{\otimes(n+1)}$, is given by

$$\begin{aligned} f_n(x_{0:n}, y_{0:n}\,;\theta) = \nu(x_0\,;\theta)g(x_0, y_0\,;\theta)q(x_0, x_1\,;\theta)g(x_1, y_1\,;\theta) \\ \cdots q(x_{n-1}, x_n\,;\theta)g(x_n, y_n\,;\theta)\ , \end{aligned} \tag{10.25}$$

where we used the same convention as above to indicate dependence with respect to the parameter θ.

Because we mainly consider estimation of the HMM parameter vector θ from a single sequence of observations, it does not make much sense to consider ν as an independent parameter. There is no hope to estimate ν consistently, as there is only one random variable X_0 (that is not even observed!) drawn from this density. In the following, we shall thus consider that ν is either fixed (and known) or fully determined by the parameter θ that appears in q and g. A typical example of the latter consists in assuming that ν is the stationary distribution associated with the transition function $q(\cdot, \cdot\,; \theta)$ (if it exists). This option is generally practicable only in very simple models (see Example 10.3.3 below for an example) because of the lack of analytical expressions relating the stationary distribution of $q(\cdot, \cdot\,; \theta)$ to θ for general parameterized hidden chains. Irrespective of whether ν is fixed or determined by θ, it is convenient to omit dependence with respect to ν in our notations, writing, for instance, E_θ for expectations under the model parameterized by (θ, ν).

Note that for left-to-right HMMs (discussed Section 1.4), the case is rather different as the model is trained from several independent sequences and the initial distribution is often a key parameter. Handling the case of multiple training sequences is straightforward as the quantities corresponding to different sequences simply need to be added together due to the independence assumption (see Section 10.3.2 below for the details in the normal HMM case).

The likelihood of the observations $\mathrm{L}_n(y_{0:n}\,; \theta)$ is obtained by integrating (10.25) with respect to the x (state) variables under the measure $\lambda^{\otimes(n+1)}$. Note that here we use yet another slight modification of the notations adopted in Section 10.1 to acknowledge that both the observations and the hidden states are indeed sequences with indices ranging from 0 to n (hence the subscript n). Upon taking the logarithm in (10.25),

$$\begin{aligned}\log f_n(x_{0:n}, y_{0:n}\,; \theta) = \log \nu(x_0\,; \theta) &+ \sum_{k=0}^{n-1} \log q(x_k, x_{k+1}\,; \theta) \\ &+ \sum_{k=0}^{n} \log g(x_k, y_k\,; \theta)\ ,\end{aligned}$$

and hence the intermediate quantity of EM has the additive structure

$$\begin{aligned}\mathcal{Q}(\theta\,; \theta') = \mathrm{E}_{\theta'}[\log \nu(X_0\,; \theta) \,|\, Y_{0:n}] &+ \sum_{k=0}^{n-1} \mathrm{E}_{\theta'}[\log q(X_k, X_{k+1}\,; \theta) \,|\, Y_{0:n}] \\ &+ \sum_{k=0}^{n} \mathrm{E}_{\theta'}[\log g(X_k, Y_k\,; \theta) \,|\, Y_{0:n}]\ .\end{aligned}$$

In the following, we will adopt the “implicit conditioning” convention that we have used extensively from Section 3.1.4 and onwards, writing $g_k(x\,; \theta)$ instead of $g(x, Y_k\,; \theta)$. With this notation, the intermediate quantity of EM may be rewritten as

$$\begin{aligned}\mathcal{Q}(\theta\,;\theta') = \mathrm{E}_{\theta'}[\log \nu(X_0\,;\theta) \mid Y_{0:n}] + \sum_{k=0}^{n} \mathrm{E}_{\theta'}[\log g_k(X_k\,;\theta) \mid Y_{0:n}] \\ + \sum_{k=0}^{n-1} \mathrm{E}_{\theta'}[\log q(X_k, X_{k+1}\,;\theta) \mid Y_{0:n}] \;. \quad (10.26)\end{aligned}$$

Equation (10.26) shows that in great generality, evaluating the intermediate quantity of EM only requires the computation of expectations under the marginal $\phi_{k|n}(\cdot\,;\theta')$ and bivariate $\phi_{k:k+1|n}(\cdot\,;\theta')$ smoothing distributions, given the parameter vector θ'. The required expectations may thus be computed using either any of the variants of the forward-backward approach presented in Chapter 3 or the recursive smoothing approach discussed in Section 4.1. To make the connection with the recursive smoothing approach of Section 4.1, we simply rewrite (10.26) as $\mathrm{E}_{\theta'}[t_n(X_{0:n}\,;\theta) \mid Y_{0:n}]$, where

$$t_0(x_0\,;\theta) = \log \nu(x_0\,;\theta) + \log g_0(x_0\,;\theta) \quad (10.27)$$

and

$$t_{k+1}(x_{0:k+1}\,;\theta) = t_k(x_{0:k}\,;\theta) + \{\log q(x_k, x_{k+1}\,;\theta) + \log g_{k+1}(x_{k+1}\,;\theta)\} \;. \quad (10.28)$$

Proposition 4.1.3 may then be applied directly to obtained the smoothed expectation of the sum functional t_n.

Although the exact form taken by the M-step will obviously depend on the way g and q depend on θ, the EM update equations follow a very systematic scheme that does not change much with the exact model under consideration. For instance, all discrete state space models for which the transition matrix q is parameterized by its $r \times r$ elements and such that g and q do not share common parameters (or parameter constraints) give rise to the same update equations for q, given in (10.43) below. Several examples of the EM update equations will be reviewed in Sections 10.3 and 10.4.

10.2.3 Computing Derivatives

Recall that the Fisher identity—(10.12)—provides an expression for the gradient of the log-likelihood $\ell_n(\theta)$ with respect to the parameter vector θ, closely related to the intermediate quantity of EM. In the HMM context, (10.12) reduces to

$$\begin{aligned}\nabla_\theta \ell_n(\theta) = \mathrm{E}_{\theta}[\nabla_\theta \log \nu(X_0\,;\theta) \mid Y_{0:n}] + \sum_{k=0}^{n} \mathrm{E}_{\theta}[\nabla_\theta \log g_k(X_k\,;\theta) \mid Y_{0:n}] \\ + \sum_{k=0}^{n-1} \mathrm{E}_{\theta}[\nabla_\theta \log q(X_k, X_{k+1}\,;\theta) \mid Y_{0:n}] \;. \quad (10.29)\end{aligned}$$

Hence the gradient of the log-likelihood may also be evaluated using either the forward-backward approach or the recursive technique discussed in Chapter 4. For the latter, we only need to redefine the functional of interest, replacing (10.27) and (10.28) by their gradients with respect to θ.

Louis' identity (10.14) gives rise to more complicated expressions, and we only consider here the case where g does depend on θ, whereas the state transition density q and the initial distribution ν are assumed to be fixed and known (the opposite situation is covered in detail in a particular case in Section 10.3.4). In this case, (10.14) may be rewritten as

$$\begin{aligned}
&\nabla^2_\theta \ell_n(\theta) + \{\nabla_\theta \ell_n(\theta)\}\{\nabla_\theta \ell_n(\theta)\}^t \qquad (10.30)\\
&\quad = \sum_{k=0}^n \mathrm{E}_\theta[\nabla^2_\theta \log g_k(X_k\,;\theta) \,\big|\, Y_{0:n}] \\
&\qquad + \sum_{k=0}^n \sum_{j=0}^n \mathrm{E}_\theta\left[\{\nabla_\theta \log g_k(X_k\,;\theta)\}\{\nabla_\theta \log g_j(X_j\,;\theta)\}^t \,\Big|\, Y_{0:n}\right] .
\end{aligned}$$

The first term on the right-hand side of (10.30) is obviously an expression that can be computed proceeding as for (10.29), replacing first- by second-order derivatives. The second term is however more tricky because it (seemingly) requires the evaluation of the joint distribution of X_k and X_j given the observations $Y_{0:n}$ for all pairs of indices k and j, which is not obtainable by the smoothing approaches based on some form of the forward-backward decomposition. The rightmost term of (10.30) is however easily recognized as a squared sum functional similar to (4.4), which can thus be evaluated recursively (in n) proceeding as in Example 4.1.4. Recall that the trick consists in observing that if

$$\tau_{n,1}(x_{0:n}\,;\theta) \stackrel{\text{def}}{=} \sum_{k=0}^n \nabla_\theta \log g_k(x_k\,;\theta) ,$$

$$\tau_{n,2}(x_{0:n}\,;\theta) \stackrel{\text{def}}{=} \left\{\sum_{k=0}^n \nabla_\theta \log g_k(x_k\,;\theta)\right\}\left\{\sum_{k=0}^n \nabla_\theta \log g_k(x_k\,;\theta)\right\}^t ,$$

then

$$\begin{aligned}
\tau_{n,2}(x_{0:n}\,;\theta) = \tau_{n-1,2}(x_{0:n-1}\,;\theta) &+ \{\nabla_\theta \log g_n(x_n\,;\theta)\}\{\nabla_\theta \log g_n(x_n\,;\theta)\}^t \\
&+ \tau_{n-1,1}(x_{0:n-1}\,;\theta)\{\nabla_\theta \log g_n(x_n\,;\theta)\}^t \\
&+ \nabla_\theta \log g_n(x_n\,;\theta)\{\tau_{n-1,1}(x_{0:n-1}\,;\theta)\}^t .
\end{aligned}$$

This last expression is of the general form given in Definition 4.1.2, and hence Proposition 4.1.3 may be applied to update recursively in n

$$\mathrm{E}_\theta[\tau_{n,1}(X_{0:n}\,;\theta)\,|\,Y_{0:n}] \qquad \text{and} \qquad \mathrm{E}_\theta[\tau_{n,2}(X_{0:n}\,;\theta)\,|\,Y_{0:n}] .$$

To make this approach more concrete, we will describe below, in Section 10.3.4, its application to a very simple finite state space HMM.

10.2.4 Connection with the Sensitivity Equation Approach

The method outlined above for evaluating the gradient of the likelihood is coherent with the general approach of Section 4.1. There is however a (seemingly) distinct approach for evaluating the same quantity, which does not require the use of Fisher's identity, and has been used for a very long time in the particular case of Gaussian linear state-space models. The method, known under the name of *sensitivity equations* (see for instance Gupta and Mehra, 1974), postulates that since the log-likelihood can be computed recursively based on the Kalman prediction recursion, its derivatives can also be computed by a recursion—the so-called *sensitivity equations*—which is obtained by differentiating the Kalman relations with respect to the model parameters. For such models, the remark that the gradient of the log-likelihood may also be obtained using Fisher's identity was made by Segal and Weinstein (1989); see also Weinstein *et al.* (1994).

The sensitivity equations approach is in no way limited to Gaussian linear state-space models but may be applied to HMMs in general. This remark, put forward by Campillo and Le Gland (1989) and Le Gland and Mevel (1997), has been subsequently used for finite state-space HMMs (Cappé *et al.*, 1998; Collings and Rydén, 1998) as well as for general HMMs (Cérou *et al.*, 2001; Doucet and Tadić, 2003). In the latter case, it is necessary to resort to some form of sequential Monte Carlo approach discussed in Chapter 7 because exact filtering is not available. It is interesting that the sequential Monte Carlo approximation method used by both Cérou *et al.* (2001) and Doucet and Tadić (2003) has also been derived by Cappé (2001a) using Fisher's identity and the smoothing framework discussed in Section 4.1. Indeed, we show below that the sensitivity equation approach is *exactly* equivalent to the use of Fisher's identity.

Recall that the log-likelihood may be written according to (3.29) as a sum of terms that only involve the prediction density,

$$\ell_n(\theta) = \sum_{k=0}^{n} \log \underbrace{\int \phi_{k|k-1}(x_k\,;\theta) g_k(x_k\,;\theta)\, \lambda(dx_k)}_{c_k(\theta)} \,, \tag{10.31}$$

where the integral is also the normalizing constant that appears in the prediction and filtering recursion (Remark 3.2.6), which we denoted by $c_k(\theta)$. The filtering recursion as given by (3.27) implies that

$$\phi_{k+1}(x_{k+1}\,;\theta) = c_{k+1}^{-1}(\theta) \int \phi_k(x_k\,;\theta) q(x_k, x_{k+1}\,;\theta) g_{k+1}(x_{k+1}\,;\theta)\, \lambda(dx_k) \,. \tag{10.32}$$

To differentiate (10.32) with respect to θ, we assume that $c_{k+1}(\theta)$ does not vanish and we use the obvious identity

$$\nabla_\theta \frac{u(\theta)}{v(\theta)} = v^{-1}(\theta) \nabla_\theta u(\theta) - \frac{u(\theta)}{v(\theta)}\, \nabla_\theta \log v(\theta)$$

to obtain

$$\nabla_\theta \phi_{k+1}(x_{k+1}\,;\theta) = \rho_{k+1}(x_{k+1}\,;\theta) - \phi_{k+1}(x_{k+1}\,;\theta)\,\nabla_\theta \log c_{k+1}(\theta)\ , \quad (10.33)$$

where

$$\rho_{k+1}(x_{k+1}\,;\theta) \overset{\text{def}}{=} c_{k+1}^{-1}(\theta)\nabla_\theta \int \phi_k(x_k\,;\theta) q(x_k, x_{k+1}\,;\theta) g_{k+1}(x_{k+1}\,;\theta)\,\lambda(dx_k)\ . \quad (10.34)$$

We further assume that as in Proposition 10.1.6, we may interchange integration with respect to λ and differentiation with respect to θ. Because $\phi_{k+1}(\cdot\,;\theta)$ is a probability density function, $\int \phi_{k+1}(x_{k+1}\,;\theta)\,\lambda(dx_{k+1}) = 1$ and $\nabla_\theta \int \phi_{k+1}(x_{k+1}\,;\theta)\,\lambda(dx_{k+1}) = \int \nabla_\theta \phi_{k+1}(x_{k+1}\,;\theta)\,\lambda(dx_{k+1}) = 0$. Therefore, integration of both sides of (10.33) with respect to $\lambda(dx_{k+1})$ yields

$$0 = \int \rho_{k+1}(x_{k+1}\,;\theta)\,\lambda(dx_{k+1}) - \nabla_\theta \log c_{k+1}(\theta)\ .$$

Hence, we may evaluate the gradient of the incremental log-likelihood in terms of ρ_{k+1} according to

$$\nabla_\theta \log c_{k+1}(\theta) \overset{\text{def}}{=} \nabla_\theta(\ell_{k+1}(\theta) - \ell_k(\theta)) = \int \rho_{k+1}(x_{k+1}\,;\theta)\,\lambda(dx_{k+1})\ . \quad (10.35)$$

Now we evaluate the derivative in (10.34) assuming also that q and g_k are non-zero to obtain

$$\begin{aligned}\rho_{k+1}(x_{k+1}\,;\theta) = c_{k+1}^{-1}(\theta) \int \Big\{ &[\nabla_\theta \log q(x_k, x_{k+1}\,;\theta) + \nabla_\theta \log g_{k+1}(x_{k+1}\,;\theta)] \\ &\times \phi_k(x_k\,;\theta) + \nabla_\theta \phi_k(x_k\,;\theta)\Big\} q(x_k, x_{k+1}\,;\theta) g_{k+1}(x_{k+1}\,;\theta)\,\lambda(dx_k)\ .\end{aligned}$$

Plugging (10.33) into the above equation yields an update formula for ρ_{k+1},

$$\begin{aligned}\rho_{k+1}(x_{k+1}\,;\theta) = c_{k+1}^{-1}(\theta) \int \Big\{ &[\nabla_\theta \log q(x_k, x_{k+1}\,;\theta) + \nabla_\theta \log g_{k+1}(x_{k+1}\,;\theta)] \\ &\times \phi_k(x_k\,;\theta) + \rho_k(x_k\,;\theta)\Big\} q(x_k, x_{k+1}\,;\theta) g_{k+1}(x_{k+1}\,;\theta)\,\lambda(dx_k) \\ &- \phi_{k+1}(x_{k+1}\,;\theta)\nabla_\theta \log c_k(\theta)\ , \quad (10.36)\end{aligned}$$

where (10.32) has been used for the last term on the right-hand side. We collect these results in the form of the algorithm below.

Algorithm 10.2.2 (Sensitivity Equations). In addition to the usual filtering recursions, do:

Initialization: Compute

$$\rho(x_0) = [\nabla_\theta \log \nu(x_0\,;\theta) + \nabla_\theta \log q_0(x_0\,;\theta)]\,\phi_0(x_0\,;\theta)$$

and $\nabla_\theta \ell_0(\theta) = \int \rho(x_0)\,\lambda(dx_0)$.

Recursion: For $k = 0, 1 \ldots$, use (10.36) to compute ρ_{k+1} and (10.35) to evaluate $\nabla_\theta \ell_{k+1}(\theta) - \nabla_\theta \ell_k(\theta)$.

Algorithm 10.2.2 updates the intermediate function $\rho_k(\cdot\,;\theta)$, defined in (10.34), whose integral is the quantity of interest $\nabla_\theta \log c_k(\theta)$. Obviously, one can equivalently use as intermediate quantity the derivative of the filtering probability density function $\nabla_\theta \phi_k(\cdot\,;\theta)$, which is directly related to $\rho_k(\cdot\,;\theta)$ by (10.33). The quantity $\nabla_\theta \phi_k(\cdot\,;\theta)$, which is referred to as the *tangent filter* by Le Gland and Mevel (1997), is also known as the *filter sensitivity* and may be of interest in its own right. Using $\nabla_\theta \phi_k(\cdot\,;\theta)$ instead of $\rho_k(\cdot\,;\theta)$ does not however modify the nature of algorithm, except for slightly more involved mathematical expressions.

It is interesting to contrast Algorithm 10.2.2 with the smoothing approach based on Fisher's identity (10.29). Recall from Section 4.1 that in order to evaluate (10.29), we recursively define a sequence of functions by

$$t_0(x_0) = \nabla_\theta \log \nu(x_0\,;\theta) + \nabla_\theta \log g_0(x_0\,;\theta)\ ,$$

and

$$t_{k+1}(x_{0:k+1}) = t_k(x_{0:k}) + \nabla_\theta \log q(x_k, x_{k+1}\,;\theta) + \nabla_\theta \log g_{k+1}(x_k\,;\theta)$$

for $k \geq 0$.

Proposition (4.1.3) asserts that $\mathrm{E}_\theta\left[t_k(X_{0:k})\,|\,Y_{0:k}\right] = \int \tau_k(x_k\,;\theta)\,\lambda(dx_k)$, where τ_k may be updated according to the recursion

$$\tau_{k+1}(x_{k+1}\,;\theta) = c_{k+1}^{-1}(\theta) \int \Big\{ [\nabla_\theta \log q(x_k, x_{k+1}\,;\theta) + \nabla_\theta \log g_{k+1}(x_{k+1}\,;\theta)] \\ \times\ \phi_k(x_k\,;\theta) + \tau_k(x_k\,;\theta) \Big\} q(x_k, x_{k+1}\,;\theta) g_{k+1}(x_{k+1}\,;\theta)\,\lambda(dx_k) \tag{10.37}$$

for $k \geq 0$, where $\tau_0(x_0\,;\theta) = c_0(\theta)^{-1}\nu(x_0)t_0(x_0)g_0(x_0)$.

Comparing (10.37) and (10.36), it is easily established by recurrence on k that $\rho_0(\cdot\,;\theta) = \tau_0(\cdot\,;\theta)$ and

$$\rho_k(\cdot\,;\theta) = \tau_k(\cdot\,;\theta) - \left(\sum_{l=0}^{k-1} \nabla_\theta \log c_l(\theta) \right) \phi_k(\cdot\,;\theta) \tag{10.38}$$

for $k \geq 1$. Hence, whereas $\int \tau_k(x_k\,;\theta)\,\lambda(dx_k)$ gives access to $\nabla_\theta \ell_k(\theta)$, the gradient of the log-likelihood up to index k, $\int \rho_k(x_k\,;\theta)\,\lambda(dx_k)$ equals the gradient of the increment $\ell_k(\theta) - \ell_{k-1}(\theta)$, where the second term is decomposed into the telescoping sum $\ell_{k-1}(\theta) = \sum_{l=0}^{k-1} \nabla_\theta \log c_l(\theta)$ of increments.

The sensitivity equations and the use of Fisher's identity combined with the recursive smoothing algorithm of Proposition 4.1.3 are thus completely equivalent. The fundamental reason for this rather surprising observation is that whereas the log-likelihood may be written, according to (10.31), as a sum of integrals under the successive prediction distributions, the same is no more true when differentiating with respect to θ. To compute the gradient of (10.31), one needs to evaluate $\rho_k(\cdot\,;\theta)$—or, equivalently, $\nabla_\theta \phi_k(\cdot\,;\theta)$—which depends on all the previous values of $c_l(\theta)$ through the sum $\sum_{l=0}^{k-1} \nabla_\theta \log c_l(\theta)$.

To conclude this section, let us stress again that there are only two different options for computing the gradient of the log-likelihood.

Forward-backward algorithm: based on Fisher's identity (10.29) and forward-backward smoothing.

Recursive algorithm: which can be equivalently derived either through the sensitivity equations or as an application of Proposition 4.1.3 starting from Fisher's identity. Both arguments give rise to the same algorithm.

These two options only differ in the way the computations are organized, as both evaluate *exactly* the sum of terms appearing in (10.29). In considering several examples below, we shall observe that the former solution is generally more efficient from the computational point of view.

10.3 The Example of Normal Hidden Markov Models

In order to make the general principles outlined in the previous section more concrete, we now work out the details on selected examples of HMMs. We begin with the case where the state space is finite and the observation transition function g corresponds to a (univariate) Gaussian distribution. Only the most standard case where the parameter vector is split into two sub-components that parameterize, respectively, g and q, is considered.

10.3.1 EM Parameter Update Formulas

In the widely used normal HMM discussed in Section 1.3.2, X is a finite set, identified with $\{1,\ldots,r\}$, $\mathsf{Y} = \mathbb{R}$, and g is a Gaussian probability density function (with respect to Lebesgue measure) given by

$$g(x, y\,;\theta) = \frac{1}{\sqrt{2\pi\upsilon_x}} \exp\left\{-\frac{(y-\mu_x)^2}{2\upsilon_x}\right\} .$$

By definition, $g_k(x\,;\theta)$ is equal to $g(x, Y_k\,;\theta)$. We first assume that the initial distribution ν is known and fixed, before examining the opposite case briefly in Section 10.3.2 below. The parameter vector θ thus encompasses the transition probabilities q_{ij} for $i, j = 1,\ldots,r$ as well as the means μ_i and variances υ_i for $i = 1,\ldots,r$. Note that in this section, because we will often need to differentiate with respect to υ_i, it is simpler to use the variances $\upsilon_i = \sigma_i^2$ rather than the standard deviations σ_i as parameters. The means and variances are unconstrained, except for the positivity of the latter, but the transition probabilities are subject to the equality constraints $\sum_{j=1}^{r} q_{ij} = 1$

for $i = 1, \ldots, r$ (in addition to the obvious constraint that q_{ij} should be non-negative). When considering the parameter vector denoted by θ', we will denote by μ_i', υ_i', and q_{ij}' its various elements.

For the model under consideration, (10.26) may be rewritten as

$$\mathcal{Q}(\theta\,;\theta') = C^{st} - \frac{1}{2}\sum_{k=0}^{n} \mathrm{E}_{\theta'}\left[\sum_{i=1}^{r} \mathbb{1}\{X_k = i\}\left(\log \upsilon_i + \frac{(Y_k - \mu_i)^2}{\upsilon_i}\right) \middle| \, Y_{0:n}\right] \\ + \sum_{k=1}^{n} \mathrm{E}_{\theta'}\left[\sum_{i=1}^{r}\sum_{j=1}^{r} \mathbb{1}\{(X_{k-1}, X_k) = (i,j)\} \log q_{ij} \middle| \, Y_{0:n}\right] ,$$

where the leading term does not depend on θ. Using the notations introduced in Section 3.1 for the smoothing distributions, we may write

$$\mathcal{Q}(\theta\,;\theta') = C^{st} - \frac{1}{2}\sum_{k=0}^{n}\sum_{i=1}^{r} \phi_{k|n}(i\,;\theta')\left[\log \upsilon_i + \frac{(Y_k - \mu_i)^2}{\upsilon_i}\right] \\ + \sum_{k=1}^{n}\sum_{i=1}^{r}\sum_{j=1}^{r} \phi_{k-1:k|n}(i, j\,;\theta') \log q_{ij} \ . \tag{10.39}$$

In the above expression, we use the same convention as in Chapter 5 and denote the smoothing probability $\mathrm{P}_{\theta'}(X_k = i \,|\, Y_{0:n})$ by $\phi_{k|n}(i\,;\theta')$ rather than by $\phi_{k|n}(\{i\}\,;\theta')$. The variable θ' is there to recall the dependence of the smoothing probability on the unknown parameters.

Now, given the initial distribution ν and parameter θ', the smoothing distributions appearing in (10.39) can be evaluated by any of the variants of forward-backward smoothing discussed in Chapter 3. As already explained above, the E-step of EM thus reduces to solving the smoothing problem. The M-step is specific and depends on the model parameterization: the task consists in finding a global optimum of $\mathcal{Q}(\theta\,;\theta')$ that satisfies the constraints mentioned above. For this, simply introduce the Lagrange multipliers $\lambda_1, \ldots, \lambda_r$ that correspond to the equality constraints $\sum_{j=1}^{r} q_{ij} = 1$ for $i = 1, \ldots, r$ (Luenberger, 1984, Chapter 10). The first-order partial derivatives of the Lagrangian

$$\mathfrak{L}(\theta, \lambda\,;\theta') = \mathcal{Q}(\theta\,;\theta') + \sum_{i=1}^{r} \lambda_i \left(1 - \sum_{j=1}^{r} q_{ij}\right)$$

are given by

$$\frac{\partial}{\partial \mu_i}\mathfrak{L}(\theta, \lambda\,;\theta') = \frac{1}{\upsilon_i}\sum_{k=0}^{n} \phi_{k|n}(i\,;\theta')(Y_k - \mu_i) \ ,$$

$$\frac{\partial}{\partial \upsilon_i}\mathfrak{L}(\theta, \lambda\,;\theta') = -\frac{1}{2}\sum_{k=0}^{n} \phi_{k|n}(i\,;\theta')\left[\frac{1}{\upsilon_i} - \frac{(Y_k - \mu_i)^2}{\upsilon_i^2}\right] ,$$

$$\frac{\partial}{\partial q_{ij}} \mathfrak{L}(\theta, \lambda\,;\theta') = \sum_{k=1}^{n} \frac{\phi_{k-1:k|n}(i, j\,;\theta')}{q_{ij}} - \lambda_i \ ,$$
$$\frac{\partial}{\partial \lambda_i} \mathfrak{L}(\theta, \lambda\,;\theta') = 1 - \sum_{j=1}^{r} q_{ij} \ . \tag{10.40}$$

Equating all expressions in (10.40) to zero yields the parameter vector

$$\theta^* = \left[(\mu_i^*)_{i=1,\ldots,r}, (\upsilon_i^*)_{i=1,\ldots,r}, (q_{ij}^*)_{i,j=1,\ldots,r}\right]$$

which achieves the maximum of $\mathcal{Q}(\theta\,;\theta')$ under the applicable parameter constraints:

$$\mu_i^* = \frac{\sum_{k=0}^{n} \phi_{k|n}(i\,;\theta') Y_k}{\sum_{k=0}^{n} \phi_{k|n}(i\,;\theta')} \ , \tag{10.41}$$
$$\upsilon_i^* = \frac{\sum_{k=0}^{n} \phi_{k|n}(i\,;\theta')(Y_k - \mu_i^*)^2}{\sum_{k=0}^{n} \phi_{k|n}(i\,;\theta')} \ , \tag{10.42}$$
$$q_{ij}^* = \frac{\sum_{k=1}^{n} \phi_{k-1:k|n}(i, j\,;\theta')}{\sum_{k=1}^{n} \sum_{l=1}^{r} \phi_{k-1:k|n}(i, l\,;\theta')} \tag{10.43}$$

for $i, j = 1, \ldots, r$, where the last equation may be rewritten more concisely as

$$q_{ij}^* = \frac{\sum_{k=1}^{n} \phi_{k-1:k|n}(i, j\,;\theta')}{\sum_{k=1}^{n} \phi_{k-1|n}(i\,;\theta')} \ . \tag{10.44}$$

Equations (10.41)–(10.43) are emblematic of the intuitive form taken by the parameter update formulas derived though the EM strategy. These equations are simply the maximum likelihood equations for the *complete model* in which both $\{X_k\}_{0 \leq k \leq n}$ and $\{Y_k\}_{0 \leq k \leq n}$ would be observed, except that the functions $\mathbb{1}\{X_k = i\}$ and $\mathbb{1}\{X_{k-1} = i, X_k = j\}$ are replaced by their conditional expectations, $\phi_{k|n}(i\,;\theta')$ and $\phi_{k-1:k|n}(i, j\,;\theta')$, given the actual observations $Y_{0:n}$ and the available parameter estimate θ'. As discussed in Section 10.1.2.4, this behavior is fundamentally due to the fact that the probability density functions associated with the complete model form an exponential family. As a consequence, the same remark holds more generally for all discrete HMMs for which the conditional probability density functions $g(i, \cdot\,;\theta)$ belong to an exponential family. A final word of warning about the way in which (10.42) is written: in order to obtain a concise and intuitively interpretable expression, (10.42) features the value of μ_i^* as given by (10.41). It is of course possible to rewrite (10.42) in a way that only contains the current parameter value θ' and the observations $Y_{0:n}$ by combining (10.41) and (10.42) to obtain

$$\upsilon_i^* = \frac{\sum_{k=0}^{n} \phi_{k|n}(i\,;\theta') Y_k^2}{\sum_{k=0}^{n} \phi_{k|n}(i\,;\theta')} - \left[\frac{\sum_{k=0}^{n} \phi_{k|n}(i\,;\theta') Y_k}{\sum_{k=0}^{n} \phi_{k|n}(i\,;\theta')}\right]^2 \ . \tag{10.45}$$

For normal HMMs, the M-step thus reduces to computing averages and ratios of simple expressions that involve the marginal and bivariate smoothing probabilities evaluated during the E-step. The number of operations associated with the implementation of these expressions scales with respect to r and n like $r^2 \times n$, which is similar to the complexity of forward-backward smoothing (see Chapter 5). In practice however, the M-step is usually faster than the E-step because operations such as sums, products, or squares are carried out faster than the exponential (recall that forward-backward smoothing requires the computation of $g_{\theta'}(i, y_k)$ for all $i = 1, \dots, r$ and $k = 0, \dots, n$). Although the difference may not be very significant for scalar models, it becomes more and more important for high-dimensional multivariate generalizations of the normal HMM, such as those used in speech recognition.

10.3.2 Estimation of the Initial Distribution

As mentioned above, in this chapter we generally assume that the initial distribution ν, that is, the distribution of X_0, is fixed and known. There are cases when one wants to treat this as an unknown parameter however, and we briefly discuss below this issue in connection with the EM algorithm for the normal HMM. We shall assume that $\nu = (\nu_i)_{1 \le i \le r}$ is an unknown probability vector (that is, with non-negative entries summing to unity), which we accommodate within the parameter vector θ. The complete log-likelihood will then be as above, where the initial term

$$\log \nu_{X_0} = \sum_{i=1}^{r} \mathbb{1}\{X_0 = i\} \log \nu_i$$

goes into $\mathcal{Q}(\theta\,; \theta')$ as well, giving the additive contribution

$$\sum_{i=1}^{r} \phi_{0|n}(i\,; \theta') \log \nu_i$$

to (10.39). This sum is indeed part of (10.39) already, but hidden within C^{st} when ν is not a parameter to be estimated. Using Lagrange multipliers as above, it is straightforward to show that the M-step update of ν is $\nu_i^* = \phi_{0|n}(i\,; \theta')$.

It was also mentioned above that sometimes it is desirable to link ν to q_θ as being the stationary distribution of q_θ. Then there is an additive contribution to $\mathcal{Q}(\theta\,; \theta')$ as above, with the difference that ν can now not be chosen freely but is a function of q_θ. As there is no simple formula for the stationary distribution of q_θ, the M-step is no longer explicit. However, once the sums (over k) in (10.39) have been computed for all i and j, we are left with an optimization problem over the q_{ij} for which we have an excellent initial guess, namely the standard update (ignoring ν) (10.43). A few steps of a standard numerical optimization routine (optimizing over the q_{ij}) is then often enough to find the maximum of $\mathcal{Q}(\cdot\,; \theta')$ under the stationarity assumption. Variants of the basic EM strategy, to be discussed in Section 10.5.3, may also be useful in this situation.

10.3.3 Recursive Implementation of E-Step

An important observation about (10.41)–(10.43) is that all expressions are ratios in which both the numerator and the denominator may be interpreted as smoothed expectations of simple additive functionals. As a consequence, the recursive smoothing techniques discussed in Chapter 4 may be used to evaluate separately the numerator and denominator of each expression. The important point here is that to implement the E-step of EM, forward-backward smoothing is not strictly required and it may be replaced by a purely recursive evaluation of the quantities involved in the M-step update.

As an example, consider the case of the first update equation (10.41) that pertains to the means μ_i. For *each pre-specified state* i, say $i = i_0$, one can devise a recursive filter to compute the quantities needed to update μ_{i_0} as follows. First define the two functionals

$$t_{n,1}(X_{0:n}) = \sum_{k=0}^{n} \mathbb{1}\{X_k = i_0\} Y_k \; ,$$
$$t_{n,2}(X_{0:n}) = \sum_{k=0}^{n} \mathbb{1}\{X_k = i_0\} \; . \tag{10.46}$$

Comparing with the general form considered in Chapter 4, the two functionals above are clearly of additive type. Hence the multiplicative functions $\{m_k\}_{0 \le k \le n}$ that appear in Definition 4.1.2 are constant and equal to one in this case. Proceeding as in Chapter 4, we associate with the functionals defined in (10.46) the sequence of signed measures

$$\tau_{n,1}(i\,;\theta') = \mathrm{E}_{\theta'}[\mathbb{1}\{X_n = i\} t_{n,1}(X_{0:n}) \,|\, Y_{0:n}] \; ,$$
$$\tau_{n,2}(i\,;\theta') = \mathrm{E}_{\theta'}[\mathbb{1}\{X_n = i\} t_{n,2}(X_{0:n}) \,|\, Y_{0:n}] \; , \tag{10.47}$$

for $i = 1, \ldots, r$. Note that we adopt here the same convention as for the smoothing distributions, writing $\tau_{n,1}(i\,;\theta')$ rather than $\tau_{n,1}(\{i\}\,;\theta')$. In this context, the expression "signed measure" is also somewhat pompous because the state space X is finite and $\tau_{n,1}$ and $\tau_{n,2}$ can safely be identified with vectors in $\mathbb{R}^r$. The numerator and denominator of (10.41) for the state $i = i_0$ are given by, respectively,

$$\sum_{i=1}^{r} \tau_{n,1}(i\,;\theta') \quad \text{and} \quad \sum_{i=1}^{r} \tau_{n,2}(i\,;\theta') \; ,$$

which can also be checked directly from (10.47), as $\sum_{i=1}^{r} \mathbb{1}\{X_n = i\}$ is identically equal to one. Recall from Chapter 4 that $\tau_{n,1}$ and $\tau_{n,2}$ are indeed quantities that may be recursively updated following the general principle of Proposition 4.1.3. Algorithm 10.3.1 below is a restatement of Proposition 4.1.3 in the context of the finite normal hidden Markov model.

Algorithm 10.3.1 (Recursive Smoothing for a Mean).

Initialization: Compute the first filtering distribution according to

$$\phi_0(i\,;\theta') = \frac{\nu(i)g_0(i\,;\theta')}{c_0(\theta')} \;,$$

for $i = 1, \ldots, r$, where $c_0(\theta') = \sum_{j=1}^r \nu(j)g_0(j\,;\theta')$. Then

$$\tau_{0,1}(i_0\,;\theta') = \phi_0(i_0\,;\theta')Y_0 \quad \text{and} \quad \tau_{0,2}(i_0\,;\theta') = \phi_0(i_0\,;\theta') \;,$$

and both $\tau_{0,1}(i\,;\theta')$ and $\tau_{0,2}(i\,;\theta')$ are set to zero for $i \neq i_0$.

Recursion: For $k = 0, \ldots, n-1$, update the filtering distribution

$$\phi_{k+1}(j\,;\theta') = \frac{\sum_{i=1}^r \phi_k(i\,;\theta')\, q'_{ij}\, g_{k+1}(j\,;\theta')}{c_{k+1}(\theta')}$$

for $j = 1, \ldots, r$, where

$$c_{k+1}(\theta') = \sum_{j=1}^r \sum_{i=1}^r \phi_k(i\,;\theta')\, q'_{ij}\, g_{k+1}(j\,;\theta') \;.$$

Next,

$$\tau_{k+1,1}(j\,;\theta') = \frac{\sum_{i=1}^r \tau_{k,1}(i\,;\theta')\, q'_{ij}\, g_{k+1}(j\,;\theta')}{c_{k+1}(\theta')} + Y_{k+1}\phi_{k+1}(i_0\,;\theta')\delta_{i_0}(j) \qquad (10.48)$$

for $j = 1, \ldots, r$, where $\delta_{i_0}(j)$ is equal to one when $j = i_0$ and zero otherwise. Likewise,

$$\tau_{k+1,2}(j\,;\theta') = \frac{\sum_{i=1}^r \tau_{k,2}(i\,;\theta')\, q'_{ij}\, g_{k+1}(j\,;\theta')}{c_{k+1}(\theta')} + \phi_{k+1}(i_0\,;\theta')\delta_{i_0}(j) \qquad (10.49)$$

for $j = 1, \ldots, r$.

Parameter Update: When the final observation index n is reached, the updated mean $\mu^*_{i_0}$ is obtained as

$$\mu^*_{i_0} = \frac{\sum_{i=1}^r \tau_{n,1}(i\,;\theta')}{\sum_{i=1}^r \tau_{n,2}(i\,;\theta')} \;.$$

It is clear that one can proceed similarly for parameters other than the means. For the same given state $i = i_0$, the alternative form of the variance update equation given in (10.45) shows that, in addition to $t_{n,1}$ and $t_{n,2}$ defined in (10.46), the functional

$$t_{n,3}(X_{0:n}) = \sum_{k=0}^n \mathbb{1}\{X_k = i_0\}Y_k^2$$

is needed to compute the updated variance $\upsilon^*_{i_0}$. The recursive smoother associated with this quantity is updated as prescribed by Algorithm 10.3.1 for $t_{n,1}$ by simply replacing Y_k by Y_k^2.

In the case of the transition probabilities, considering *a fixed pair* of states (i_0, j_0), (10.44) implies that in addition to evaluating $\tau_{n-1,2}$, one needs to derive a smoother for the functional

$$t_{n,4}(X_{0:n}) = \sum_{k=1}^{n} \mathbb{1}\{X_{k-1} = i_0, X_k = j_0\} \,, \tag{10.50}$$

where $t_{0,4}(X_0)$ is defined to be null. Following Proposition 4.1.3, the associated smoothed quantity

$$\tau_{n,4}(i\,; \theta') = \mathrm{E}_{\theta'}[\mathbb{1}\{X_n = i\} t_{n,4}(X_{0:n}) \,|\, Y_{0:n}]$$

may be updated recursively according to

$$\tau_{k+1,4}(j\,; \theta') = \frac{\sum_{i=1}^{r} \tau_{k,4}(i\,; \theta')\, q'_{ij}\, g_{k+1}(j\,; \theta')}{c_{k+1}(\theta')} + \frac{\phi_k(i_0\,; \theta')\, q'_{i_0 j_0}\, g_{k+1}(j_0\,; \theta') \delta_{j_0}(j)}{c_{k+1}(\theta')} \,, \tag{10.51}$$

where $\delta_{j_0}(j)$ equal to one when $j = j_0$ and zero otherwise, and c_{k+1} and ϕ_k should be computed recursively as in Algorithm 10.3.1. Because $\tau_{0,4}$ is null, the recursion is initialized by setting $\tau_{0,4}(i\,; \theta') = 0$ for all states $i = 1, \ldots, r$.

The case of the transition probabilities clearly illustrates the main weakness of the recursive approach, namely that a specific recursive smoother must be implemented for each statistic of interest. Indeed, for each time index k, (10.48), (10.49), or (10.51) require of the order of r^2 operations, which is comparable with the computational cost of the (normalized) forward or filtering recursion (Algorithm 5.1.1). The difference is that after application of the complete forward-backward recursions, one may compute *all the statistics* involved in the EM re-estimation equations (10.41)–(10.43). In contrast, the recursive smoothing recursion only provides the smoothed version of *one particular statistic*: in the case of (10.51) for instance, this is (10.50) with a fixed choice of the pair i_0, j_0. Hence implementing the EM algorithm with recursive smoothing requires the order of $r^2 \times (n+1) \times \dim(\theta)$ operations, where $\dim(\theta)$ refers to the number of parameters. In the case of the complete (scalar) normal HMM, $\dim(\theta)$ equals $2r$ for the means and variances, plus $r \times (r-1)$ for the transition probabilities. Hence recursive smoothing is clearly not competitive with approaches based on the forward-backward decomposition.

To make it short, the recursive smoothing approach is not a very attractive option in finite state space HMMs and normal HMMs in particular. More precisely, both the intermediate quantity of EM in (10.26) and the gradient of the log-likelihood in (10.29) are additive. In the terminology used in Section 4.1.2, they both correspond to additive functionals of the form $t_{n+1}(x_{0:n+1}) = t_n(x_{0:n}) + s_n(x_n, x_{n+1})$. In such cases, smoothing approaches based on the forward-backward decompositions such as Algorithms 5.1.2 or 5.1.3 that evaluate the bivariate smoothing distributions $\phi_{k:k+1|n}$ for $k = 0, \ldots, n-1$ are more efficient because they do not require that the functions $\{s_k\}_{k=0,\ldots,n-1}$ be specified. There are however some situations in which the recursive smoothing approach developed in Section 4.1 and illustrated above in the case of normal HMMs may be useful.

- First, because it is recursive, it does not require that the intermediate computation results be stored, which is in sharp contrast with the other smoothing approaches where either the forward or backward variables need to be stored. This is of course of interest when processing very large data sets.
- When the functional whose conditional expectation is to be evaluated is not of the additive type, approaches based on the evaluation of bivariate smoothing distributions are not applicable anymore. In contrast, recursive smoothing stays feasible as long as the functional follows the general pattern of Definition 4.1.2. The most significant functional of practical interest that is not additive is the second-order derivative of the log-likelihood function. The use of recursive smoothing for this purpose will be illustrated on a simple example in Section 10.3.4.

Finally, another different motivation for computing either the intermediate quantity of EM or the gradient of the log-likelihood recursively has to do with recursive estimation. As noted by several authors, including Le Gland and Mevel (1997), Collings and Rydén (1998), and Krishnamurthy and Yin (2002), being able to compute recursively the intermediate quantity of EM or the gradient of the log-likelihood is a key step toward efficient recursive (also called on-line or adaptive) parameter estimation approaches. It is important however to understand that recursive computation procedures do not necessarily directly translate into recursive estimation approaches. Algorithm 10.3.1 for instance describes how to compute the EM update of the mean μ_i given some observations $Y_0, \ldots, Y_n$ and a *fixed* current parameter value $\theta = \theta'$. In recursive estimation on the other hand, once a new observation Y_k is collected, the parameter estimate, $\hat{\theta}_k$ say, needs to be updated. Using the equations of Algorithm 10.3.1 with $\hat{\theta}_k$ substituted for θ' is of course a natural idea, but not one that is guaranteed to produce the desired result. This is precisely the objective of works such as Le Gland and Mevel (1997) and Krishnamurthy and Yin (2002), to study if and when such recursive approaches do produce expected results. It is fair to say that, as of today, this remains a largely open issue.

10.3.4 Computation of the Score and Observed Information

For reasons discussed above, computing the gradient of the log-likelihood is not a difficult task in finite state space HMMs and should preferably be done using smoothing algorithms based on the forward-backward decomposition. The only new requirement is to evaluate the derivatives with respect to θ that appear in (10.29). In the case of the normal HMM, we already met the appropriate expressions in (10.40), as Fisher's identity (10.12) implies that the gradient of the intermediate quantity at the current parameter estimate coincides with the gradient of the log-likelihood. Hence

$$\begin{aligned}
\frac{\partial}{\partial \mu_i}\ell_n(\theta) &= \frac{1}{\upsilon_i}\sum_{k=0}^{n}\phi_{k|n}(i\,;\theta)(Y_k-\mu_i)\ ,\\
\frac{\partial}{\partial \upsilon_i}\ell_n(\theta) &= -\frac{1}{2}\sum_{k=0}^{n}\phi_{k|n}(i\,;\theta)\left[\frac{1}{\upsilon_i}-\frac{(Y_k-\mu_i)^2}{\upsilon_i^2}\right]\ ,\\
\frac{\partial}{\partial q_{ij}}\ell_n(\theta) &= \sum_{k=1}^{n}\frac{\phi_{k-1:k|n}(i,j\,;\theta)}{q_{ij}}\ .
\end{aligned}$$

Recall also that the log-likelihood itself is directly available from the filtering recursion, following (5.4).

Before considering the computation of the Hessian, we first illustrate the performance of the optimization methods introduced in Section 10.1.3, which only require the evaluation of the log-likelihood and its gradient.

Example 10.3.2 (Binary Deconvolution Model). We consider the simple binary deconvolution model of Cappé *et al.* (1999), which is somewhat related to the channel coding situation described in Example 1.3.2, except that the channel is unknown. This model is of interest in digital communications (see for instance Krishnamurthy and White, 1992; Kaleh and Vallet, 1994; Fonollosa *et al.*, 1997). It is given by

$$Y_k = \sum_{i=0}^{p} h_i B_{k-i} + N_k\ , \tag{10.52}$$

where $\{Y_k\}_{k\geq 0}$ is the observed sequence, $\{N_k\}_{k\geq 0}$ is a stationary sequence of white Gaussian noise with zero mean and variance υ, and $\{B_k\}_{k\geq 0}$ is a sequence of transmitted symbols. For simplicity, we assume that $\{B_k\}_{k\geq 0}$ is a binary, i.e., $B_k \in \{-1, 1\}$, sequence of i.i.d. fair Bernoulli draws. We consider below that $p = 1$, so that to cast the model into the HMM framework, we only need to define the state as the vector $X_k = (B_k, B_{k-1})^t$, which takes one of the four possible values

$$s_1 \stackrel{\text{def}}{=} \begin{pmatrix}-1\\-1\end{pmatrix},\quad s_2 \stackrel{\text{def}}{=} \begin{pmatrix}-1\\1\end{pmatrix},\quad s_3 \stackrel{\text{def}}{=} \begin{pmatrix}1\\-1\end{pmatrix},\quad s_4 \stackrel{\text{def}}{=} \begin{pmatrix}1\\1\end{pmatrix}.$$

Hence, upon defining the vector $h \stackrel{\text{def}}{=} (h_0\ h_1)^t$ of filter coefficients, we may view (10.52) as a four-states normal HMM such that $\mu_i = s_i^t h$ and $\upsilon_i = \upsilon$ for $i = 1, \ldots, 4$. The transition matrix Q is entirely fixed by our assumption that the binary symbols are equiprobable, and is given by

$$Q = \begin{pmatrix} 1/2 & 0 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \\ 0 & 1/2 & 0 & 1/2 \end{pmatrix} .$$

The model parameters to be estimated are thus the vector h of filter coefficients and the common variance υ. For simplicity, we assume that the distribution of the initial state X_0 is known.

To make the connection with the general (unconstrained) normal hidden Markov model discussed previously, we need only take into account the facts that $\nabla_h \mu_i = s_i$ and $\partial \upsilon_i / \partial \upsilon = 1$, as all variances are equal. Hence, using the chain rule, the gradient of the intermediate quantity of EM may be evaluated from (10.40) as

$$\begin{aligned} \nabla_h \mathcal{Q}(\theta\,;\theta') &= \sum_{i=1}^{4} \frac{\partial}{\partial \mu_i} \mathcal{Q}(\theta\,;\theta') \nabla_h \mu_i \\ &= \frac{1}{\upsilon} \sum_{i=1}^{4} \sum_{k=0}^{n} \phi_{k|n}(i\,;\theta')(Y_k s_i - s_i s_i^t h) \end{aligned} \tag{10.53}$$

and

$$\begin{aligned} \frac{\partial}{\partial \upsilon} \mathcal{Q}(\theta\,;\theta') &= \sum_{i=1}^{4} \frac{\partial}{\partial \upsilon_i} \mathcal{Q}(\theta\,;\theta') \frac{\partial \upsilon_i}{\partial \upsilon} \\ &= -\frac{1}{2} \left[\frac{n}{\upsilon} - \sum_{i=1}^{4} \sum_{k=0}^{n} \phi_{k|n}(i\,;\theta') \frac{(Y_k - s_i^t h)^2}{\upsilon^2} \right] . \end{aligned} \tag{10.54}$$

The M-step update equations (10.41) and (10.42) of the EM algorithm should thus be replaced by

$$h^* = \left[\sum_{i=1}^{4} \sum_{k=0}^{n} \phi_{k|n}(i\,;\theta') s_i s_i^t \right]^{-1} \left[\sum_{i=1}^{4} \sum_{k=0}^{n} \phi_{k|n}(i\,;\theta') Y_k s_i \right] ,$$

$$\begin{aligned} \upsilon^* &= \frac{1}{n} \sum_{i=1}^{4} \sum_{k=0}^{n} \phi_{k|n}(i\,;\theta')(Y_k - s_i^t h^*)^2 \\ &= \frac{1}{n} \left\{ \sum_{k=0}^{n} Y_k^2 - \left[\sum_{i=1}^{4} \sum_{k=0}^{n} \phi_{k|n}(i\,;\theta') Y_k s_i \right]^t h^* \right\} . \end{aligned}$$

For computing the log-likelihood gradient, we may resort to Fisher's identity, setting $\theta = \theta'$ in (10.53) and (10.54) to obtain $\nabla_h \ell_n(\theta')$ and $\frac{\partial}{\partial \upsilon} \ell_n(\theta')$, respectively.

We now compare the results of the EM algorithm and of a quasi-Newton method for this model. In both cases, the forward-backward recursions are used to compute the smoothing probabilities $\phi_{k|n}(i\,;\theta')$ for $k = 0, \ldots, n$ and $i = 1, \ldots, 4$. To avoid parameter constraints, we compute the partial derivative with respect to $\log \upsilon$ rather than with respect to υ, as the parameter $\log \upsilon$ is unconstrained. This modification is not needed for the EM algorithm, which is parameterization independent. The quasi-Newton optimization is performed using the so-called BFGS weight update and cubic line-searches (see Fletcher, 1987, for details concerning the former).

The data set under consideration is the same as in Cappé *et al.* (1999) and consists of 150 synthetic observations generated with the model corresponding to $h_0 = 1.3$, $h_1 = 0.6$ and $\upsilon = (h_0^2 + h_1^2)/4$ (6 dB signal to noise ratio). There are three parameters for this model, and Figures 10.1 and 10.2 show plots of the profile log-likelihood for values of h_0 and h_1 on a regular grid. The profile log-likelihood is $\ell_n(h, \hat{\upsilon}(h))$ with $\hat{\upsilon}(h) = \arg\max_{\upsilon} \ell_n(h, \upsilon)$, that is, the largest possible log-likelihood for a fixed value of h. The figures show that the profile log-likelihood has a global maximum, the MLE, as well as a local one. The location of the local maximum (or maxima) as well as its presence obviously depends on the particular outcome of the simulated noise $\{N_k\}$. It is

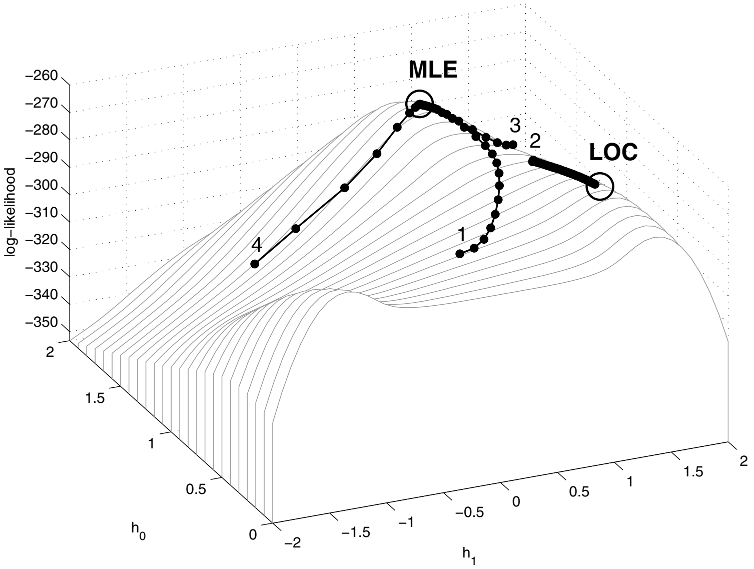

Fig. 10.1. Profile log-likelihood surface over $(h_0,\, h_1)$ for a particular realization of the binary deconvolution model. The true model parameters are $h_0 = 1.3$ and $h_1 = 0.6$, and 150 observations were taken. The two circled positions labeled MLE and LOC are, respectively, the global maximum of the profile log-likelihood and a local maximum. Also shown are trajectories of 35 iterations of the EM algorithm, initialized at four different points marked 1–4.

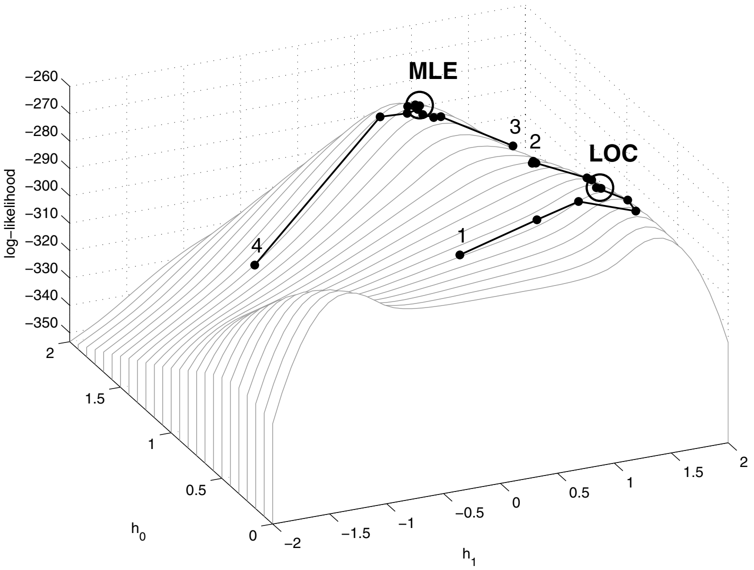

Fig. 10.2. Same profile log-likelihood surface as in Figure 10.1. Also shown are trajectories of 5 iterations of a quasi-Newton algorithm, initialized at the same four different points marked 1–4 as in Figure 10.1.

a fundamental feature of the model however that the parameters $h = (h_0\ h_1)^t$ and $h = (h_1\ h_0)^t$, which govern identical second-order statistical properties of the model, are difficult to discriminate, especially with few observations. Note that as swapping the signs of both h_0 and h_1 leaves the model unchanged, the profile log-likelihood surface is symmetric, and only the half corresponding to positive values of h_0 is shown here.

A first remark is that even in such a simplistic model, there is a local maximum and, depending on the initialization, both algorithms may converge to this point. Because the algorithms operate differently, it may even occur that the EM and quasi-Newton algorithms initialized at the same point eventually converge to different values, as in the case of initialization at point 1. The other important remark is that the EM algorithm (Figure 10.1) shows very different convergence behavior depending on the region of the parameter space where it starts: when initialized at point 4, the algorithm gets real close to the MLE in about seven iterations, whereas when initialized at point 1 or 2, it is still far from having reached convergence after 20 iterations. In contrast, the quasi-Newton method (Figure 10.2) updates the parameter by doing steps that are much larger than those of EM, especially during the first iterations, and provides very accurate parameter estimates with as few as five iterations. It is fair to say that due to the necessity of evaluating the weight matrix (with finite difference computations) and to the cubic line-search procedure, each iteration of the quasi-Newton method requires on average seven evaluations of the log-likelihood and its gradient, which means in particular seven instances of the forward-backward procedure. From a computational point of view, the time needed to run the 5 iterations of the quasi-Newton method in this example is thus roughly equivalent to that required for 35 iterations of the EM algorithm. ■

As discussed earlier, computing the observed information in HMMs is more involved, as the only computationally feasible option consists in adopting the recursive smoothing framework of Proposition 4.1.3. Rather than embarking into the general normal HMM case, we consider another simpler illustrative example where the parameter of interest is scalar.

Example 10.3.3. Consider a simplified version of the ion channel model (Example 1.3.5) in which the state space X is composed of two states that are (by convention) labeled 0 and 1, and $g(0,y)$ and $g(1,y)$ respectively correspond to the $\mathrm{N}(0,\upsilon)$ and $\mathrm{N}(1,\upsilon)$ distributions. This model may also be interpreted as a state space model in which

$$Y_k = X_k + V_k \ ,$$

where $\{V_k\}$ is an i.i.d. $\mathrm{N}(0,\upsilon)$-distributed sequence, independent of the Markov chain $\{X_k\}$, which takes its values in the set $\{0,1\}$. The transition matrix Q of $\{X_k\}$ is parameterized in the form

$$Q = \begin{pmatrix} \rho_0 & 1-\rho_0 \\ 1-\rho_1 & \rho_1 \end{pmatrix} .$$

It is also most logical in this case to assume that the initial distribution ν of X_0 coincides with the stationary distribution associated with Q, that is, $\nu(0) = \rho_0/(\rho_0+\rho_1)$ and $\nu(1) = \rho_1/(\rho_0+\rho_1)$. In this model, the distributions of holding times (number of consecutive steps k for which X_k stays constant) have geometric distributions with expectations $(1-\rho_0)^{-1}$ and $(1-\rho_1)^{-1}$ for states 0 and 1, respectively. ■

We now focus on the computation of the derivatives of the log-likelihood in the model of Example 10.3.3 with respect to the transition parameters ρ_0 and ρ_1. As they play a symmetric role, it is sufficient to consider, say, ρ_0 only. The variance υ is considered as fixed so that the only quantities that depend on the parameter ρ_0 are the initial distribution ν and the transition matrix Q. We will, as usual, use the simplified notation $g_k(x)$ rather than $g(x, Y_k)$ to denote the Gaussian density function $(2\pi\upsilon)^{-1/2}\exp\{-(Y_k - x)^2/(2\upsilon)\}$ for $x \in \{0,1\}$. Furthermore, in order to simplify the expressions below, we also omit to indicate explicitly the dependence with respect to ρ_0 in the rest of this section. Fisher's identity (10.12) reduces to

$$\frac{\partial}{\partial \rho_0}\ell_n = \mathrm{E}\left[\frac{\partial}{\partial \rho_0} \log \nu(X_0) + \sum_{k=0}^{n-1} \frac{\partial}{\partial \rho_0} \log q_{X_k X_{k+1}} \,\middle|\, Y_{0:n} \right] ,$$

where the notation q_{ij} refers to the element in the $(1+i)$-th row and $(1+j)$-th column of the matrix Q (in particular, q_{00} and q_{11} are alternative notations for ρ_0 and ρ_1). We are thus in the framework of Proposition 4.1.3 with a smoothing functional $t_{n,1}$ defined by

$$\begin{aligned} t_{0,1}(x) &= \frac{\partial}{\partial \rho_0} \log \nu(x) \ , \\ s_{k,1}(x, x') &= \frac{\partial}{\partial \rho_0} \log q_{xx'} \qquad \text{for } k \geq 0 \ , \end{aligned}$$

where the multiplicative functions $\{m_{k,1}\}_{k \geq 0}$ are equal to 1. Straightforward calculations yield

$$\begin{aligned} t_{0,1}(x) &= (\rho_0 + \rho_1)^{-1} \left[\frac{\rho_1}{\rho_0} \delta_0(x) - \delta_1(x) \right] , \\ s_{k,1}(x, x') &= \frac{1}{\rho_0} \delta_{(0,0)}(x, x') - \frac{1}{1 - \rho_0} \delta_{(0,1)}(x, x') \ . \end{aligned}$$

Hence a first recursion, following Proposition 4.1.3.

Algorithm 10.3.4 (Computation of the Score in Example 10.3.3).

Initialization: Compute $c_0 = \sum_{i=0}^{1} \nu(i) g_0(i)$ and, for $i = 0, 1$,

$$\begin{aligned} \phi_k(i) &= c_0^{-1} \nu(i) g_0(i) \ , \\ \tau_{0,1}(i) &= t_{0,1}(i) \phi_0(i) \ . \end{aligned}$$

Recursion: For $k = 0, 1, \ldots$, compute $c_{k+1} = \sum_{i=0}^{1} \sum_{j=0}^{1} \phi_k(i) q_{ij} g_k(j)$ and, for $j = 0, 1$,

$$\begin{aligned} \phi_{k+1}(j) &= c_{k+1}^{-1} \sum_{i=0}^{1} \phi_k(i) q_{ij} g_k(j) \ , \\ \tau_{k+1,1}(j) &= c_{k+1}^{-1} \Bigg\{ \sum_{i=0}^{1} \tau_{k,1}(i) q_{ij} g_{k+1}(j) \\ &\qquad + \phi_k(0) g_{k+1}(0) \delta_0(j) - \phi_k(0) g_{k+1}(1) \delta_1(j) \Bigg\} \ . \end{aligned}$$

At each index k, the log-likelihood is available via $\ell_k = \sum_{l=0}^{k} \log c_l$, and its derivative with respect to ρ_0 may be evaluated as

$$\frac{\partial}{\partial \rho_0} \ell_k = \sum_{i=0}^{1} \tau_{k,1}(i) \ .$$

For the second derivative, Louis' identity (10.14) shows that

$$\frac{\partial^2}{\partial \rho_0^2}\ell_n + \left\{\frac{\partial}{\partial \rho_0}\ell_n\right\}^2 = \mathrm{E}\left[\frac{\partial^2}{\partial \rho_0^2}\log \nu(X_0) + \sum_{k=0}^{n-1}\frac{\partial^2}{\partial \rho_0^2}\log q_{X_k X_{k+1}} \,\middle|\, Y_{0:n}\right]$$
$$+ \mathrm{E}\left[\left(\frac{\partial}{\partial \rho_0}\log \nu(X_0) + \sum_{k=0}^{n-1}\frac{\partial}{\partial \rho_0}\log q_{X_k X_{k+1}}\right)^2 \,\middle|\, Y_{0:n}\right] . \quad (10.55)$$

The first term on the right-hand side of (10.55) is very similar to the case of $\tau_{n,1}$ considered above, except that we now need to differentiate the functions twice, replacing $t_{0,1}$ and $s_{k,1}$ by $\frac{\partial}{\partial \rho_0}t_{0,1}$ and $\frac{\partial}{\partial \rho_0}s_{k,1}$, respectively. The corresponding smoothing functional $t_{n,2}$ is thus now defined by

$$t_{0,2}(x) = -\frac{\rho_1(2\rho_0+\rho_1)}{\rho_0^2(\rho_0+\rho_1)^2}\delta_0(x) + \frac{1}{(\rho_0+\rho_1)^2}\delta_1(x) ,$$
$$s_{k,2}(x,x') = -\frac{1}{\rho_0^2}\delta_{(0,0)}(x,x') - \frac{1}{(1-\rho_0)^2}\delta_{(0,1)}(x,x') .$$

The second term on the right-hand side of (10.55) is more difficult, and we need to proceed as in Example 4.1.4: the quantity of interest may be rewritten as the conditional expectation of

$$t_{n,3}(x_{0:n}) = \left[t_{0,1}(x_0) + \sum_{k=0}^{n-1} s_{k,1}(x_k, x_{k+1})\right]^2 .$$

Expanding the square in this equation yields the update formula

$$t_{k+1,3}(x_{0:k+1}) = t_{k,3}(x_{0:k}) + s_{k,1}^2(x_k,x_{k+1}) + 2t_{k,1}(x_{0:k})s_{k,1}(x_k,x_{k+1}) .$$

Hence $t_{k,1}$ and $t_{k,3}$ jointly are of the form prescribed by Definition 4.1.2 with incremental additive functions $s_{k,3}(x,x') = s_{k,1}^2(x,x')$ and multiplicative updates $m_{k,3}(x,x') = 2s_{k,1}(x,x')$. As a consequence, the following smoothing recursion holds.

Algorithm 10.3.5 (Computation of the Observed Information in Example 10.3.3).

Initialization: For $i = 0, 1$,

$$\tau_{0,2}(i) = t_{0,2}(i)\phi_0(i) .$$
$$\tau_{0,3}(i) = t_{0,1}^2(i)\phi_0(i) .$$

Recursion: For $k = 0, 1, \ldots$, compute for $j = 0, 1$,

$$\begin{aligned}\tau_{k+1,2}(j) = c_{k+1}^{-1}\Bigg\{ &\sum_{i=0}^{1} \tau_{k,2}(i) q_{ij} g_{k+1}(j) \\ &- \frac{1}{\rho_0}\phi_k(0) g_{k+1}(0)\delta_0(j) - \frac{1}{(1-\rho_0)}\phi_k(0) g_{k+1}(1)\delta_1(j)\Bigg\} , \\ \tau_{0,3}(j) = c_{k+1}^{-1}\Bigg\{ &\sum_{i=0}^{1} \tau_{k,3}(i) q_{ij} g_{k+1}(j) \\ &+ 2\left[\tau_{k,1}(0) g_{k+1}(0)\delta_0(j) - \tau_{k,1}(0) g_{k+1}(1)\delta_1(j)\right] \\ &+ \frac{1}{\rho_0}\phi_k(0) g_{k+1}(0)\delta_0(j) + \frac{1}{(1-\rho_0)}\phi_k(0) g_{k+1}(1)\delta_1(j)\Bigg\} .\end{aligned}$$

At each index k, the second derivative of the log-likelihood satisfies

$$\frac{\partial^2}{\partial\rho_0^2}\ell_k + \left(\frac{\partial}{\partial\rho_0}\ell_k\right)^2 = \sum_{i=0}^{1} \tau_{k,2}(i) + \sum_{i=0}^{1} \tau_{k,3}(i) ,$$

where the second term on the left-hand side may be evaluated in the same recursion, following Algorithm 10.3.4.

To illustrate the results obtained with Algorithms 10.3.4–10.3.5, we consider the model with parameters $\rho_0 = 0.95$, $\rho_1 = 0.8$, and $\upsilon = 0.1$ (using the notations introduced in Example 10.3.3). Figure 10.3 displays the typical aspect of two sequences of length 200 simulated under slightly different values of ρ_0. One possible use of the output of Algorithms 10.3.4–10.3.5 consists in testing for changes in the parameter values. Indeed, under conditions to be detailed in Chapter 12 (and which hold here), the normalized score $n^{-1/2}\frac{\partial}{\partial\rho_0}\ell_n$ satisfies a central limit theorem with variance given by the limit of the normalized information $-n^{-1}(\partial^2/\partial\rho_0^2)\ell_n$. Hence it is expected that

$$\mathfrak{R}_n = \frac{\frac{\partial}{\partial\rho_0}\ell_n}{\sqrt{-\frac{\partial^2}{\partial\rho_0^2}\ell_n}}$$

be asymptotically $\mathrm{N}(0,1)$-distributed under the null hypothesis that ρ_0 is indeed equal to the value used for computing the score and information recursively with Algorithms 10.3.4–10.3.5.

Figure 10.4 displays the empirical quantiles of $\mathfrak{R}_n$ against normal quantiles for $n = 200$ and $n =$1,000. For the longer sequences ($n =$1,000), the result is clearly as expected with a very close fit to the normal quantiles. When $n = 200$, asymptotic normality is not yet reached and there is a significant bias toward high values of $\mathfrak{R}_n$. Looking back at Figure 10.3, even if υ was equal to zero—or in other words, if we were able to identify without ambiguity the 0 and 1 states from the data—there would not be much information about ρ_0 to be gained from runs of length 200: when $\rho_0 = 0.95$ and $\rho_1 = 0.8$, the

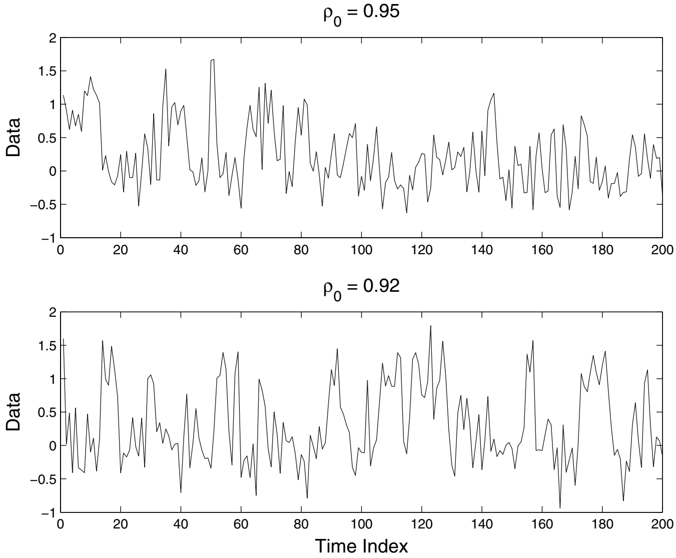

Fig. 10.3. Two simulated trajectories of length $n = 200$ from the simplified ion channel model of Example 10.3.3 with $\rho_0 = 0.95$, $\rho_1 = 0.8$, and $\sigma^2 = 0.1$ (top), and $\rho_0 = 0.92$, $\rho_1 = 0.8$, and $\sigma^2 = 0.1$ (bottom).

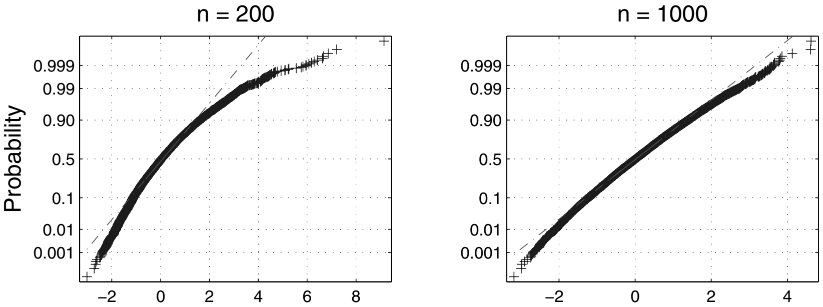

Fig. 10.4. QQ-plot of empirical quantiles of the test statistic $\mathfrak{R}_n$ (abscissas) for the simplified ion channel model of Example 10.3.3 with $\rho_0 = 0.95$, $\rho_1 = 0.8$, and $\sigma^2 = 0.1$ vs. normal quantiles (ordinates). Samples sizes were $n = 200$ (left) and n =1,000 (right), and 10,000 independent replications were used to estimate the empirical quantiles.

average number of distinct runs of 0s that one can observe in 200 consecutive data points is only about $200/(20 + 5) = 8$. To construct a goodness of fit test from $\mathfrak{R}_n$, one can monitor values of $\mathfrak{R}_n^2$, which asymptotically has a chi-square distribution with one degree of freedom. Testing the null hypothesis $\rho_0 = 0.95$ gives p-values of 0.87 and 0.09 for the two sequences in the top and bottom plots, respectively, of Figure 10.3. When testing at the 10% level, both sequences thus lead to the correct decision: no rejection and rejection of the null hypothesis, respectively. Interestingly, testing the other way around, that is, postulating $\rho_0 = 0.92$ as the null hypothesis, gives p-values of 0.20 and 0.55 for the top and bottom sequences of Figure 10.3, respectively. The outcome of the test is now obviously less clear-cut, which reveals an asymmetry in its discrimination ability: it is easier to detect values of ρ_0 that are smaller than expected than the converse. This is because smaller values of ρ_0 means more changes (on average) in the state sequence and hence more usable information about ρ_0 to be obtained from a fixed size record. This asymmetry is connected to the upward bias visible in the left plot of Figure 10.4.

10.4 The Example of Gaussian Linear State-Space Models

We now consider more briefly the case of Gaussian linear state-space models that form the other major class of hidden Markov models for which the methods discussed in Section 10.1 are directly applicable. It is worth mentioning that Gaussian linear state-space models are perhaps the only important subclass of the HMM family for which there exist reasonable simple non-iterative parameter estimation algorithms not based on maximum likelihood arguments but are nevertheless useful in practical applications. These sub-optimal algorithms, proposed by Van Overschee and De Moor (1993), rely on the linear structure of the model and use only eigendecompositions of empirical covariance matrices—a general principle usually referred to under the denomination of *subspace methods* (Van Overschee and De Moor, 1996). Keeping in line with the general topic of this chapter, we nonetheless consider below only algorithms for maximum likelihood estimation in Gaussian linear state-space models.

The Gaussian linear state-space model introduced in Section 1.3.3 is given in so-called state-space form by (1.7)–(1.8), which we recall here:

$$\begin{aligned} X_{k+1} &= AX_k + RU_k \ , \\ Y_k &= BX_k + SV_k \ , \end{aligned}$$

where X_0, $\{U_k\}_{k\geq 0}$ and $\{V_k\}_{k\geq 0}$ are jointly Gaussian. The parameters of the model are the four matrices A, R, B, and S. Note that except for scalar models, it is not possible to estimate R and S because both $\{U_k\}$ and $\{V_k\}$ are unobservable and hence R and S are only identifiable up to an orthonormal matrix. In other words, multiplying R or S by any orthonormal matrix of suitable dimension does not modify the distribution of the observations. Hence the parameters that are identifiable are the covariance matrices $\Upsilon_R = RR^t$ and $\Upsilon_S = SS^t$, which we consider below. Likewise, the matrices A and B are identifiable up to a similarity transformation only. Indeed, setting $X'_k = TX_k$ for some invertible matrix T, that is, making a change of basis for the state process, it is straightforward to check that the joint process $\{(X'_k, Y_k)\}$ satisfies the model assumptions with TAT^{-1}, BT^{-1}, and TR replacing A, B, and R, respectively. Nevertheless, we work with A and B in the algorithm below. If a unique representation is desired, one may use, for instance, the companion form of A given its eigenvalues; this matrix may contain complex entries though. As in the case of finite state space HMMs (Section 10.2.2), it is not sensible to consider the initial covariance matrix Σ_ν as an independent parameter when using a single observed sequence. On the other hand, for such models it is very natural to assume that Σ_ν is associated with the stationary distribution of $\{X_k\}$. Except for the particular case of the scalar AR(1) model however (to be discussed in Example 11.1.2), this option typically renders the EM update equations non-explicit and it is thus standard practice to treat Σ_ν as a fixed matrix unrelated to the parameters (Ghosh, 1989). We shall also assume that both Υ_R and Υ_S are full rank covariance matrices so that all Gaussian distributions admit densities with respect to (multi-dimensional) Lebesgue measure.

10.4.1 The Intermediate Quantity of EM

With the previous notations, the intermediate quantity $\mathcal{Q}(\theta\,;\theta')$ of EM, defined in (10.26), may be expressed as

$$\begin{aligned} &-\frac{1}{2}\,\mathrm{E}_{\theta'}\left[n\log|\Upsilon_R| + \sum_{k=0}^{n-1}(X_{k+1}-AX_k)^t\Upsilon_R^{-1}(X_{k+1}-AX_k) \,\middle|\, Y_{0:n}\right] \\ &\qquad -\frac{1}{2}\,\mathrm{E}_{\theta'}\left[(n+1)\log|\Upsilon_S| + \sum_{k=0}^{n}(Y_k-BX_k)^t\Upsilon_S^{-1}(Y_k-BX_k) \,\middle|\, Y_{0:n}\right], \end{aligned} \tag{10.56}$$

up to terms that do not depend on the parameters. In order to elicit the M-step equations or to compute the score, we differentiate (10.56) using elementary perturbation calculus as well as the identity $\nabla_C \log|C| = C^{-t}$ for an invertible matrix C—which is a consequence of the adjoint representation of the inverse (Horn and Johnson, 1985, Section 0.8.2):

$$\nabla_A \mathcal{Q}(\theta\,;\theta') = -\Upsilon_R^{-1}\,\mathrm{E}_{\theta'}\left[\sum_{k=0}^{n-1}(AX_kX_k^t - X_{k+1}X_k^t)\,\middle|\, Y_{0:n}\right], \tag{10.57}$$

$$\begin{aligned} \nabla_{\Upsilon_R^{-1}} \mathcal{Q}(\theta\,;\theta') = -\frac{1}{2}\Bigg\{ &-n\Upsilon_R \\ &+ \mathrm{E}_{\theta'}\left[\sum_{k=0}^{n-1}(X_{k+1}-AX_k)(X_{k+1}-AX_k)^t \,\middle|\, Y_{0:n}\right]\Bigg\}, \end{aligned} \tag{10.58}$$

$$\nabla_B \mathcal{Q}(\theta\,;\theta') = -\Upsilon_S^{-1}\,\mathrm{E}_{\theta'}\left[\sum_{k=0}^{n}(BX_kX_k^t - Y_kX_k^t)\,\middle|\, Y_{0:n}\right], \tag{10.59}$$

$$\nabla_{\Upsilon_S^{-1}} \mathcal{Q}(\theta\,;\theta') = -\frac{1}{2}\Bigg\{ -(n+1)\Upsilon_S \tag{10.60}$$
$$+ \mathrm{E}_{\theta'}\left[\sum_{k=0}^{n} (Y_k - BX_k)(Y_k - BX_k)^t \,\middle|\, Y_{0:n} \right] \Bigg\} .$$

Note that in the expressions above, we differentiate with respect to the inverses of Υ_R and Υ_S rather than with respect to the covariance matrices themselves, which is equivalent, because we assume both of the covariance matrices to be positive definite, but yields simpler formulas. Equating all derivatives simultaneously to zero defines the EM update of the parameters. We will denote these updates by A^*, B^*, Υ_R^*, and Υ_S^*, respectively. To write them down, we will use the notations introduced in Chapter 5: $\hat{X}_{k|n}(\theta') = \mathrm{E}_{\theta'}[X_k \,|\, Y_{0:n}]$ and $\Sigma_{k|n}(\theta') = \mathrm{E}_{\theta'}[X_k X_k' \,|\, Y_{0:n}] - \hat{X}_{k|n}(\theta')\hat{X}_{k|n}^t(\theta')$, where we now indicate explicitly that these first two smoothing moments indeed depend on the current estimates of the model parameters (they also depend on the initial covariance matrix Σ_ν, but we ignore this fact here because this quantity is considered as being fixed). We also need to evaluate the conditional covariances

$$C_{k,k+1|n}(\theta') \overset{\text{def}}{=} \mathrm{Cov}_{\theta'}[X_k, X_{k+1} \,|\, Y_{0:n}]$$
$$= \mathrm{E}_{\theta'}[X_k X_{k+1}^t \,|\, Y_{0:n}] - \hat{X}_{k|n}(\theta')\hat{X}_{k+1|n}^t(\theta') .$$

For Gaussian models, the latter expression coincides with the definition given in (5.99), and hence one may use expression (5.100) to evaluate $C_{k,k+1|n}(\theta')$ during the final forward recursion of Algorithm 5.2.15.

With these notations, the EM update equations are given by

$$A^* = \left[\sum_{k=0}^{n-1} C_{k,k+1|n}(\theta') + \hat{X}_{k|n}(\theta')\hat{X}_{k+1|n}^t(\theta') \right]^t \tag{10.61}$$
$$\left[\sum_{k=0}^{n-1} \Sigma_{k|n}(\theta') + \hat{X}_{k|n}(\theta')\hat{X}_{k|n}^t(\theta') \right]^{-1} ,$$

$$\Upsilon_R^* = \frac{1}{n} \sum_{k=0}^{n-1} \Bigg\{ \left[\Sigma_{k+1|n}(\theta') + \hat{X}_{k+1|n}(\theta')\hat{X}_{k+1|n}^t(\theta') \right] \tag{10.62}$$
$$- A^* \left[C_{k,k+1|n}(\theta') + \hat{X}_{k|n}(\theta')\hat{X}_{k+1|n}^t(\theta') \right] \Bigg\} ,$$

$$B^* = \left[\sum_{k=0}^{n} \hat{X}_{k|n}(\theta') Y_k^t \right]^t \tag{10.63}$$
$$\left[\sum_{k=0}^{n} \Sigma_{k|n}(\theta') + \hat{X}_{k|n}(\theta')\hat{X}_{k|n}^t(\theta') \right]^{-1} ,$$

$$\Upsilon_S^* = \frac{1}{n+1} \sum_{k=0}^{n} \left[Y_k Y_k^t - B^* \hat{X}_{k|n}(\theta') Y_k^t \right] . \tag{10.64}$$

In obtaining the covariance update, we used the same remark that made it possible to rewrite, in the case of normal HMMs, (10.42) as (10.45).

10.4.2 Recursive Implementation

As in the case of finite state space HMMs, it is possible to implement the parameter update equations (10.61)–(10.64) or to compute the gradient (10.57)–(10.60) of the log-likelihood recursively in n. Here we only sketch the general principles and refer to the paper by Elliott and Krishnamurthy (1999) in which the details of the EM re-estimation equations are worked out. Proceeding as in Section 4.1, it is clear that all expressions under consideration may be rewritten term by term as the expectation[2] $\mathrm{E}[t_n(X_{0:n}) \mid Y_{0:n}]$ of well chosen additive functionals t_n. More precisely, the functionals of interest are of the form $t_n(x_{0:n}) = t_0(x_0) + \sum_{k=0}^{n-1} s_k(x_k, x_{k+1})$, where the individual terms in the sum are of one of the types

$$s_{k-1,1}(x_k) = h_k^t x_k , \tag{10.65}$$

$$s_{k-1,2}(x_k) = x_k^t M_k x_k , \tag{10.66}$$

$$s_{k-1,3}(x_{k-1}, x_k) = x_{k-1}^t T_{k-1} x_k , \tag{10.67}$$

and $\{h_k\}_{k \geq 0}$, $\{M_k\}_{k \geq 0}$, and $\{T_k\}_{k \geq 0}$, respectively, denote sequences of vectors and matrices with dimension that of the state vectors (d_x) and which may depend on the model parameters or on the observations.

For illustration purposes, we focus on the example of (10.63): the first factor on the right-hand side of (10.63) is a matrix whose ij elements (ith row, jth column) corresponds to $\mathrm{E}[\sum_{k=0}^{n} h_k^t X_k \mid Y_{0:n}]$ for the particular choice

$$h_k = \begin{array}{c} \begin{pmatrix} 0 \ldots & 0 & Y_k(i) & 0 & \ldots \; 0 \end{pmatrix}^t \\ \; 1 \ldots & j-1 & j & j+1 & \ldots \; d_x \end{array} . \tag{10.68}$$

Likewise, the ij element of the second factor on the right-hand side of (10.63)—before inverting the matrix—corresponds to the expectation of a functional of the second of the three types above with M_k being a matrix of zeros except for a unit entry at position ij.

Let $\tau_{n,1}$ denote the expectation $\mathrm{E}[\sum_{k=0}^{n} h_k^t X_k \mid Y_{0:n}]$ for an additive functional of the first type given in (10.65). To derive a recursion for $\tau_{n,1}$, we use the innovation decomposition (Section 5.2.2) to obtain

[2] Note that in this section, we omit to indicate explicitly the dependence with respect to the model parameters to alleviate the notations.

$$\begin{aligned}
\tau_{n+1,1} &\overset{\text{def}}{=} \mathrm{E}_{\theta'}\left[\sum_{k=0}^{n+1} h_k^t X_k \,\middle|\, Y_{0:n+1}\right] \\
&= h_{n+1}^t \hat{X}_{n+1|n+1} \\
&\quad + \sum_{k=0}^{n} h_k^t \left(\hat{X}_{k|n} + \mathrm{E}[X_k \epsilon_{n+1}^t] \Gamma_{n+1}^{-1} \epsilon_{n+1}\right) \\
&= h_{n+1}^t \hat{X}_{n+1|n+1} + \mathrm{E}\left[\sum_{k=0}^{n} h_k^t X_k \,\middle|\, Y_{0:n}\right] \\
&\quad + \underbrace{\left(\sum_{k=0}^{n} h_k^t \Sigma_{k|k-1} \Lambda_k^t \; \Lambda_{k+1}^t \; \dots \; \Lambda_n^t\right)}_{r_{n+1}} B^t \Gamma_{n+1}^{-1} \epsilon_{n+1} \; ,
\end{aligned}$$

where (5.93) was used to obtain the last expression, which also features the notation $\Lambda_k = A - H_k B$ with H_k being the Kalman (prediction) gain introduced in the statement of Algorithm 5.2.15. The term that we denoted by r_{n+1} is an intermediate quantity that has some similarities with the variable p_k (or more precisely p_0) that is instrumental in the disturbance smoothing algorithm (Algorithm 5.2.15). The same key remark applies here as r_n can be computed recursively (in n) according to the equations

$$\begin{aligned}
r_0 &= 0 \; , \\
r_{n+1} &= \left(r_n + h_n \Sigma_{n|n-1}\right) \Lambda_n^t \qquad \text{for } n \geq 0 \; .
\end{aligned}$$

Hence the following recursive smoothing algorithm, which collects all necessary steps.

Algorithm 10.4.1 (Recursive Smoothing for a Linear Sum Functional).

Initialization: Apply the Kalman filtering recursion for $k = 0$ (Algorithm 5.2.13) and set

$$\begin{aligned}
r_0 &= 0 \; , \\
\tau_0 &= \mathrm{E}[h_0^t X_0 \,|\, Y_0] = h_0^t \hat{X}_{0|0} \; .
\end{aligned}$$

Recursion: For $n = 1, 2, \dots$, run one step of the Kalman filtering and prediction recursions (Algorithms 5.2.9 and 5.2.13) and compute

$$\begin{aligned}
r_n &= \left(r_{n-1} + h_{n-1} \Sigma_{n-1|n-2}\right) \Lambda_{n-1}^t \; , \\
\tau_n &= \mathrm{E}\left[\sum_{k=0}^{n} h_k^t X_k \,\middle|\, Y_{0:n}\right] = h_n^t \hat{X}_{n|n} + \tau_{n-1} + r_n B^t \Gamma_n^{-1} \epsilon_n \; .
\end{aligned}$$

Algorithm 10.4.1 illustrates the fact that as in the case of finite state space models, recursive computation is in general less efficient than is forward-backward smoothing from a computational point of view: although Algorithm 10.4.1 capitalizes on a common framework formed by the Kalman filtering and prediction recursions, it does however require the update of a quantity (r_n) that is specific to the choice of the sequence of vectors $\{h_k\}_{k\geq 0}$. To compute the first factor on the right-hand side of (10.63) for instance, one needs to apply the recursion of Algorithm 10.4.1 for the $d_y \times d_x$ possible choices of $\{h_k\}_{k\geq 0}$ given by (10.68). Thus, except for low-dimensional models or particular cases in which the system matrices A, Υ_R, B, and Υ_S are very sparse, recursive computation is usually not the method of choice for Gaussian linear state-space models (see Elliott and Krishnamurthy, 1999, for a discussion of the complexity of the complete set of equations required to carry out the EM parameter update).

10.5 Complements

To conclude this chapter, we briefly return to an issue mentioned in Section 10.1.2 regarding the conditions that ensure that the EM iterations indeed converge to stationary points of the likelihood.

10.5.1 Global Convergence of the EM Algorithm

As a consequence of Proposition 10.1.4, the EM algorithm described in Section 10.1.2 has the property that the log-likelihood function ℓ can never decrease in an iteration. Indeed,

$$\ell(\theta^{i+1}) - \ell(\theta^i) \geq \mathcal{Q}(\theta^{i+1}\,;\theta^i) - \mathcal{Q}(\theta^i\,;\theta^i) \geq 0 \;.$$

This class of algorithms, sometimes referred to as *ascent algorithms* (Luenberger, 1984, Chapter 6), can be treated in a unified manner following a theory developed mostly by Zangwill (1969). Wu (1983) showed that this general theory applies to the EM algorithm as defined above, as well as to some of its variants that he calls generalized EM (or GEM). The main result is a strong stability guarantee known as *global convergence*, which we discuss below.

We first need a mathematical formalism that describes the EM algorithm. This is done by identifying any homogeneous (in the iterations) iterative algorithm with a specific choice of a mapping M that associates θ^{i+1} to θ^i. In the theory of Zangwill (1969), one indeed considers families of algorithms by allowing for *point-to-set* maps M that associate a set $M(\theta') \subseteq \Theta$ to each parameter value $\theta' \in \Theta$. A specific algorithm in the family is such that θ^{i+1} is selected in $M(\theta^i)$. In the example of EM, we may define M as

$$M(\theta') = \left\{\theta \in \Theta \,:\, \mathcal{Q}(\theta\,;\theta') \geq \mathcal{Q}(\tilde{\theta}\,;\theta') \text{ for all } \tilde{\theta} \in \Theta\right\} \;, \tag{10.69}$$

that is, $M(\theta')$ is the set of values θ that maximize $\mathcal{Q}(\theta\,;\theta')$ over Θ. Usually $M(\theta')$ reduces to a singleton, and the mapping M is then simply a point-to-point map (a usual function from Θ to Θ). But the use of point-to-set maps makes it possible to deal also with cases where the intermediate quantity of EM may have several global maxima, without going into the details of what is done in such cases. We next need the following definition before stating the main convergence theorem.

Definition 10.5.1 (Closed Mapping). *A map T from points of Θ to subsets of Θ is said to be* closed *on a set $\mathcal{S} \subseteq \Theta$ if for any converging sequences $\{\theta^i\}_{i\geq 0}$ and $\{\tilde{\theta}^i\}_{i\geq 0}$, the conditions*

(a) $\theta^i \to \theta \in \mathcal{S}$,
(b) $\tilde{\theta}^i \to \tilde{\theta}$ with $\tilde{\theta}^i \in T(\theta^i)$ for all $i \geq 0$,

imply that $\tilde{\theta} \in T(\theta)$.

Note that for point-to-point maps, that is, if $T(\theta)$ is a singleton for all θ, the definition above is equivalent to the requirement that T be continuous on $\mathcal{S}$. Definition 10.5.1 is thus a generalization of continuity for general (point-to-set) maps. We are now ready to state the main result, which is proved in Zangwill (1969, p. 91) or Luenberger (1984, p. 187).

Theorem 10.5.2 (Global Convergence Theorem). *Let Θ be a subset of $\mathbb{R}^{d_\theta}$ and let $\{\theta^i\}_{i\geq 0}$ be a sequence generated by $\theta^{i+1} \in T(\theta^i)$ where T is a point-to-set map on Θ. Let $\mathcal{S} \subseteq \Theta$ be a given "solution" set and suppose that*

(1) the sequence $\{\theta^i\}_{i\geq 0}$ is contained in a compact subset of Θ;
(2) T is closed over $\Theta \setminus \mathcal{S}$ (the complement of $\mathcal{S}$);
(3) there is a continuous "ascent" function s on Θ such that $s(\theta) \geq s(\theta')$ for all $\theta \in T(\theta')$, with strict inequality for points θ' that are not *in $\mathcal{S}$.*

Then the limit of any convergent subsequence of $\{\theta^i\}$ is in the solution set $\mathcal{S}$. In addition, the sequence of values of the ascent function, $\{s(\theta^i)\}_{i\geq 0}$, converges monotonically to $s(\theta_\star)$ for some $\theta_\star \in \mathcal{S}$.

The final statement of Theorem 10.5.2 should not be misinterpreted: that $\{s(\theta^i)\}$ converges to a value that is the image of a point in $\mathcal{S}$ is a simple consequence of the first and third assumptions. It does however not imply that the sequence of parameters $\{\theta^i\}$ is itself convergent in the usual sense, but only that the limit points of $\{\theta^i\}$ have to be in the solution set $\mathcal{S}$. An important property however is that because $\{s(\theta^{i(l)})\}_{l\geq 0}$ converges to $s(\theta_\star)$ for any convergent subsequence $\{\theta^{i(l)}\}$, all limit points of $\{\theta^i\}$ must be in the set $\mathcal{S}_\star = \{\theta \in \Theta : s(\theta) = s(\theta_\star)\}$ (in addition to being in $\mathcal{S}$). This latter statement means that the sequence of iterates $\{\theta^i\}$ will ultimately approach a set of points that are "equivalent" as measured by the ascent function s.

The following general convergence theorem following the proof by Wu (1983) is a direct application of the previous theory to the case of EM.

Theorem 10.5.3. *Suppose that in addition to the hypotheses of Proposition 10.1.4 (Assumptions 10.1.3 as well as parts (a) and (b) of Proposition 10.1.4), the following hold.*

(i) $\mathcal{H}(\theta\,;\theta')$ *is continuous in its second argument,* θ'*, on* Θ*.*
(ii) For any θ^0*, the level set* $\Theta^0 = \left\{\theta \in \Theta : \ell(\theta) \geq \ell(\theta^0)\right\}$ *is compact and contained in the interior of* Θ*.*

Then all limit points of any instance $\{\theta^i\}_{i\geq 0}$ *of an EM algorithm initialized at* θ^0 *are in* $\mathcal{L}^0 = \{\theta \in \Theta^0 : \nabla_\theta \ell(\theta) = 0\}$*, the set of stationary points of* ℓ *with log-likelihood larger than that of* θ^0*. The sequence* $\{\ell(\theta^i)\}$ *of log-likelihoods converges monotonically to* $\ell_\star = \ell(\theta_\star)$ *for some* $\theta_\star \in \mathcal{L}^0$*.*

Proof. This is a direct application of Theorem 10.5.2 using $\mathcal{L}^0$ as the solution set and ℓ as the ascent function. The first hypothesis of Theorem 10.5.2 follows from (ii) and the third one from Proposition 10.1.4. The closedness assumption (2) follows from Proposition 10.1.4 and (i): for the EM mapping M defined in (10.69), $\tilde{\theta}^i \in M(\theta^i)$ amounts to the condition

$$\mathcal{Q}(\tilde{\theta}^i\,;\theta^i) \geq \mathcal{Q}(\theta\,;\theta^i) \quad \text{for all } \theta \in \Theta\ ,$$

which is also satisfied by the limits of the sequences $\{\tilde{\theta}^i\}$ and $\{\theta^i\}$ (if these converge) by continuity of the intermediate quantity $\mathcal{Q}$, which follows from that of ℓ and $\mathcal{H}$ (note that it is here important that $\mathcal{H}$ be continuous with respect to both arguments). Hence the EM mapping is indeed closed on Θ as a whole and Theorem 10.5.3 follows. □

The assumptions of Proposition 10.1.4 as well as item (i) above are indeed very mild in typical situations. Assumption (ii) however may be restrictive, even for models in which the EM algorithm is routinely used (such as the normal HMMs introduced in Section 1.3.2, for which this assumption does not hold if the variances υ_i are allowed to be arbitrarily small). The practical implication of (ii) being violated is that the EM algorithm may fail to converge to the stationary points of the likelihood for some particularly badly chosen initial points θ^0.

Most importantly, the fact that θ^{i+1} maximizes the intermediate quantity $\mathcal{Q}(\cdot\,;\theta^i)$ of EM does in no way imply that, ultimately, $\ell_\star$ is the global maximum of ℓ over Θ. There is even no guarantee that $\ell_\star$ is a local maximum of the log-likelihood: it may well only be a saddle point (Wu, 1983, Section 2.1). Also, the convergence of the sequence $\ell(\theta^i)$ to $\ell_\star$ does not automatically imply the convergence of $\{\theta^i\}$ to a point $\theta_\star$.

Pointwise convergence of the EM algorithm requires more stringent assumptions that are difficult to verify in practice. As an example, a simple corollary of the global convergence theorem states that if the solution set $\mathcal{S}$ in Theorem 10.5.2 is a single point, $\theta_\star$ say, then the sequence $\{\theta^i\}$ indeed converges to $\theta_\star$ (Luenberger, 1984, p. 188). The sketch of the proof of this corollary is that every subsequence of $\{\theta^i\}$ has a convergent further subsequence because of the compactness assumption (1), but such a subsequence admits s as an ascent function and thus converges to $\theta_\star$ by Theorem 10.5.2 itself. In cases where the solution set is composed of several points, further conditions are needed to ensure that the sequence of iterates indeed converges and does not cycle through different solution points.

In the case of EM, pointwise convergence of the EM sequence may be guaranteed under an additional condition given by Wu (1983) (see also Boyles, 1983, for an equivalent result), stated in the following theorem.

Theorem 10.5.4. *Under the hypotheses of Theorem 10.5.3, if*

(iii) $\|\theta^{i+1} - \theta^i\| \to 0$ *as* $i \to \infty$,

then all limit points of $\{\theta^i\}$ *are in a connected and compact subset of* $\mathcal{L}_\star = \{\theta \in \Theta : \ell(\theta) = \ell_\star\}$, *where* $\ell_\star$ *is the limit of the log-likelihood sequence* $\{\ell(\theta^i)\}$.

In particular, if the connected components of $\mathcal{L}_\star$ *are singletons, then* $\{\theta^i\}$ *converges to some* $\theta_\star$ *in* $\mathcal{L}_\star$.

Proof. The set of limit points of a bounded sequence $\{\theta^i\}$ with $\|\theta^{i+1} - \theta^i\| \to 0$ is connected and compact (Ostrowski, 1966, Theorem 28.1). The proof follows becuase under Theorem 10.5.2, the limit points of $\{\theta^i\}$ must belong to $\mathcal{L}_\star$. □

10.5.2 Rate of Convergence of EM

Even if one can guarantee that the EM sequence $\{\hat{\theta}^i\}$ converges to some point $\theta_\star$, this limiting point can be either a local maximum, a saddle point, or even a local minimum. The proposition below states conditions under which the stable stationary points of EM coincide with local maxima only (see also Lange, 1995, Proposition 1, for a similar statement). We here consider that the EM mapping M is a point-to-point map, that is, that the maximizer in the M-step is unique.

To understand the meaning of the term "stable", consider the following approximation to the limit behavior of the EM sequence: it is sensible to expect that if the EM mapping M is sufficiently regular in a neighborhood of the limiting fixed point $\theta_\star$, the asymptotic behavior of the EM sequence $\{\theta^i\}$ follows the tangent linear dynamical system

$$(\theta^{i+1} - \theta_\star) = M(\theta^i) - M(\theta_\star) \approx \nabla_\theta M(\theta_\star)(\theta^i - \theta_\star) . \tag{10.70}$$

Here $\nabla_\theta M(\theta_\star)$ is called the *rate matrix* (see for instance Meng and Rubin, 1991). A fixed point $\theta_\star$ is said to be *stable* if the spectral radius of $\nabla_\theta M(\theta_\star)$ is less than 1. In this case, the tangent linear system is asymptotically stable in the sense that the sequence $\{\zeta^i\}$ defined recursively by $\zeta^{i+1} = \nabla_\theta M(\theta_\star)\zeta^i$ tends to zero as n tends to infinity (for any choice of ζ^0). The linear *rate of convergence* of EM is defined as the largest moduli of the eigenvalues of $\nabla_\theta M(\theta_\star)$. This rate is an upper bound on the factors ρ_k that appear in (10.17).

Proposition 10.5.5. *Under the assumptions of Theorem 10.1.6, assume that* $\mathcal{Q}(\cdot\,;\theta)$ *has a unique maximizer for all* $\theta \in \Theta$ *and that, in addition,*

$$H(\theta_\star) = -\int \nabla^2_\theta \log f(x\,;\theta)\big|_{\theta=\theta_\star} p(x\,;\theta_\star)\,\lambda(dx) \tag{10.71}$$

and

$$G(\theta_\star) = -\int \nabla^2_\theta \log p(x\,;\theta)\big|_{\theta=\theta_\star} p(x\,;\theta_\star)\,\lambda(dx) \tag{10.72}$$

are positive definite matrices for all stationary points of EM (i.e., such that $M(\theta_\star) = \theta_\star$*). Then for all such points, the following hold true.*

(i) $\nabla_\theta M(\theta_\star)$ *is diagonalizable and its eigenvalues are positive real numbers.*
(ii) The point $\theta_\star$ *is stable for the mapping* M *if and only if it is a proper maximizer of* $\ell(\theta)$ *in the sense that all eigenvalues of* $\nabla^2_\theta \ell(\theta_\star)$ *are negative.*

Proof. The EM mapping is defined implicitly through the fact that $M(\theta')$ maximizes $\mathcal{Q}(\cdot\,;\theta')$, which implies that

$$\int \nabla_\theta \, \log f(x\,;\theta)|_{\theta=M(\theta')}\, p(x\,;\theta')\,\lambda(dx) = 0 \ ,$$

using assumption (b) of Theorem 10.1.6. Careful differentiation of this relation at a point $\theta' = \theta_\star$, which is such that $M(\theta_\star) = \theta_\star$ and hence $\nabla_\theta \, \ell(\theta)|_{\theta=\theta_\star} = 0$, gives (Dempster *et al.*, 1977; Lange, 1995, see also)

$$\nabla_\theta M(\theta_\star) = [H(\theta_\star)]^{-1}\left[H(\theta_\star) + \nabla^2_\theta \ell(\theta_\star)\right] \ ,$$

where $H(\theta_\star)$ is defined in (10.71). The missing information principle—or Louis' formula (see Proposition 10.1.6)—implies that $G(\theta_\star) = H(\theta_\star) + \nabla^2_\theta \ell(\theta_\star)$ is positive definite under our assumptions.

Thus $\nabla_\theta M(\theta_\star)$ is diagonalizable with positive eigenvalues that are the same (counting multiplicities) as those of the matrix $A_\star = I + B_\star$, where $B_\star = [H(\theta_\star)]^{-1/2}\nabla^2_\theta \ell(\theta_\star)[H(\theta_\star)]^{-1/2}$. Thus $\nabla_\theta M(\theta_\star)$ is stable if and only if $B_\star$ has negative eigenvalues only. The Sylvester law of inertia (see for instance Horn and Johnson, 1985) shows that $B_\star$ has the same inertia (number of positive, negative, and zero eigenvalues) as $\nabla^2_\theta \ell(\theta_\star)$. Thus all of $B_\star$'s eigenvalues are negative if and only if the same is true for $\nabla^2_\theta \ell(\theta_\star)$, that is, if $\theta_\star$ is a proper maximizer of ℓ. □

The proof above implies that when $\theta_\star$ is stable, the eigenvalues of $M(\theta_\star)$ lie in the interval $(0, 1)$.

10.5.3 Generalized EM Algorithms

As discussed above, the type of convergence guaranteed by Theorem 10.5.3 is rather weak but, on the other hand, this result is remarkable as it indeed covers not only the original EM algorithm proposed by Dempster *et al.* (1977) but a whole class of variants of the EM approach. One of the most useful extensions of EM is the ECM (for expectation conditional maximization) by Meng and Rubin (1993), which addresses situations where direct maximization of the intermediate quantity of EM is intractable. Assume for instance that the parameter vector θ consists of two sub-components θ_1 and θ_2, which are such that maximization of $\mathcal{Q}((\theta_1, \theta_2)\,; \theta')$ with respect to θ_1 or θ_2 only (the other sub-component being fixed) is easy, whereas joint maximization with respect to $\theta = (\theta_1, \theta_2)$ is problematic. One may then use the following algorithm for updating the parameter estimate at iteration i.

E-step: Compute $\mathcal{Q}((\theta_1, \theta_2)\,; (\theta_1^i, \theta_2^i))$;
CM-step: Determine

$$\theta_1^{i+1} = \arg\max_{\theta_1} \mathcal{Q}((\theta_1, \theta_2^i)\,; (\theta_1^i, \theta_2^i)) \ ,$$

and then

$$\theta_2^{i+1} = \arg\max_{\theta_2} \mathcal{Q}((\theta_1^{i+1}, \theta_2)\,; (\theta_1^i, \theta_2^i)) \ .$$

It is easily checked that for this algorithm, (10.8) is still verified and thus ℓ is an ascent function; this implies that Theorem 10.5.3 holds under the same set of assumptions.

The example above is only the simplest case where the ECM approach may be applied, and further extensions are discussed by Meng and Rubin (1993) as well as by Fessler and Hero (1995) and Meng and Dyk (1997).

10.5.4 Bibliographic Notes

The EM algorithm was popularized by the celebrated article of Dempster *et al.* (1977). It is generally admitted however that several published works predated this landmark paper by describing applications of the EM principle to some specific cases (Meng and Dyk, 1997). Interestingly, the earliest example of a complete EM strategy, which also includes convergence proofs (in addition to describing the forward-backward smoothing algorithm discussed in Chapter 3), is indeed the work by Baum *et al.* (1970) on finite state space HMMs, generalizing the idea put forward by Baum and Eagon (1967). This pioneering contribution has been extended by authors such as Liporace (1982), who showed that the same procedure could be applied to other types of HMMs. The generality of the approach however was not fully recognized until Dempster *et al.* (1977) and Wu (1983) (who made the connection with the theory of global convergence) showed that the convergence of the EM approach (and its generalizations) is guaranteed in great generality.

The fact that the EM algorithm may also be used, with minor modifications, for MAP estimation was first mentioned by Dempster *et al.* (1977). Green (1990) illustrates a number of practical applications where this option plays an important role. Perhaps the most significant of these is speech processing where MAP estimation, as first described by Gauvain and Lee (1994), is commonly used for the model adaptation task (that is, re-retraining from sparse data of some previously trained models).

The ECM algorithm of Meng and Rubin (1993) (discussed Section 10.5.3) was also studied independently by Fessler and Hero (1995) under the name SAGE (space-alternating generalized EM). Fessler and Hero (1995) also introduced the idea that in some settings it is advantageous to use different ways of augmenting the data, that is, different ways of writing the likelihood as in (10.1) depending on the parameter subset that one is trying to re-estimate; see also Meng and Dyk (1997) for further developments of this idea.