Technical report from Automatic Control at Linköpings universitet

An Explanation of the Expectation Maximization Algorithm

Thomas B. Schön

Division of Automatic Control

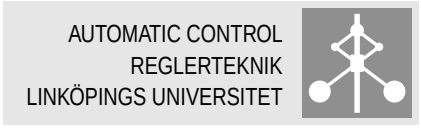
E-mail: schon@isy.liu.se

21st August 2009

Report no.: LiTH-ISY-R-2915

Address:
Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

WWW: http://www.control.isy.liu.se



Technical reports from the Automatic Control group in Linköping are available from http://www.control.isy.liu.se/publications.

Abstract

The expectation maximization (EM) algorithm computes maximum likelihood estimates of unknown parameters in probabilistic models involving latent variables. More pragmatically speaking, the EM algorithm is an iterative method that alternates between computing a conditional expectation and solving a maximization problem, hence the name expectation maximization. We will in this work derive the EM algorithm and show that it provides a maximum likelihood estimate. The aim of the work is to show how the EM algorithm can be used in the context of dynamic systems and we will provide a worked example showing how the EM algorithm can be used to solve a simple system identification problem.

Keywords: Expectation Maximization, system identification, Maximum likelihood, latent variables, probabilistic models.

An Explanation of the Expectation Maximization Algorithm

Thomas B. Schön

2009-08-21

Abstract

The expectation maximization (EM) algorithm computes maximum likelihood estimates of unknown parameters in probabilistic models involving latent variables. More pragmatically speaking, the EM algorithm is an iterative method that alternates between computing a conditional expectation and solving a maximization problem, hence the name expectation maximization. We will in this work derive the EM algorithm and show that it provides a maximum likelihood estimate. The aim of the work is to show how the EM algorithm can be used in the context of dynamic systems and we will provide a worked example showing how the EM algorithm can be used to solve a simple system identification problem.

1 Introduction

The expectation maximization (EM) algorithm computes maximum likelihood (ML) estimates of unknown parameters θ in probabilistic models involving latent variables Z[1]. An instructive way of thinking about EM is to think of it as a systematic way of separating one hard problem into two new closely linked problems, each of which is hopefully more tractable than the original problem. This problem separation forms the very heart of the EM algorithm.

More pragmatically speaking, the EM algorithm is an iterative method that alternates between computing a conditional expectation and solving a maximization problem, hence the name expectation maximization. To thoroughly appreciate the EM algorithm, it is important to understand why the above mentioned problem separation indeed results in an ML estimate. This will be explained in detail below.

The motivation for this work is to provide a basic introduction to the EM algorithm within the setting of dynamic systems. More specifically, the main focus is to explain how the EM algorithm can be used for estimating models of dynamic systems, i.e., *system identification*. That is, besides the general introduction and derivation of the EM algorithm given in Section 2, we will see how it can be used to identify parameters in a simple, but still very instructive

[1] The term *latent variable* is adopted from statistics and refers to a variable that is not directly observed. Hence, a latent variable has to be inferred (through a mathematical model) from other variables that are directly observed, i.e., measured. Latent variables are sometimes also referred to as hidden variables of unobserved variables and within the EM literature they are sometimes called the missing data or the incomplete data.

case in Section 3. In Section 4 we provide some brief insights into the, by now rather large, literature surrounding the EM algorithm and its applications and in Section 5 we provide the conclusions. Finally, in the Appendix we give some of the details skipped in the main text and the MATLAB code for the example discussed in Section 3.

2 A Formal Derivation of the EM Algorithm

In order to derive the EM algorithm in Section 2.2 we must first clearly define the ML problem, which is the topic of Section 2.1.

2.1 Maximum Likelihood Estimation

The *maximum likelihood* method, which was introduced by Fisher (1912, 1922), is based on the rather natural idea that the unknown parameters should be chosen in such a way that the observed measurements becomes as *likely as possible*. More specifically, the ML estimate is computed according to

$$\hat{\theta}^{\mathrm{ML}} \triangleq \arg\max_{\theta} p_\theta(y_1, \ldots, y_N), \tag{1}$$

where y_t denotes the measurement at time t. Furthermore, subindex θ indicates that the corresponding probability density function $p_\theta(y_1, \ldots, y_N)$ is parameterised by the (unknown) parameter θ. The joint density of the observations $p_\theta(y_1, \ldots, y_N)$ can, using the definition of conditional probabilities, be written as

$$p_\theta(y_1, \ldots, y_N) = p_\theta(y_1) \prod_{t=2}^{N} p_\theta(y_t | Y_{t-1}), \tag{2}$$

where $Y_{t-1} \triangleq \{y_1, \ldots, y_{t-1}\}$. It is often convenient to consider the so called log-likelihood function

$$L_\theta(Y) \triangleq \log p_\theta(y_1, \ldots, y_N) = \sum_{t=2}^{N} \log p_\theta(y_t \mid Y_{t-1}) + \log p_\theta(y_1), \tag{3}$$

rather than the likelihood function. In the interest of a more compact notation we have here introduced the notation $Y \triangleq \{y_1, \ldots, y_N\}$. The logarithm is a strictly increasing function, implying that the following problem is equivalent to (1)

$$\hat{\theta}^{\mathrm{ML}} = \arg\max_{\theta} \sum_{t=2}^{N} \log p_\theta(y_t \mid Y_{t-1}) + \log p_\theta(y_1). \tag{4}$$

This problem can of course be solved using standard methods such as Newton's method or one of its related variants, see e.g., Dennis and Schnabel (1983); Nocedal and Wright (2006) for details on these methods. However, the ML problem can also be solved using the expectation maximization algorithm, an approach that has steadily gained in popularity since its formal birth in 1977 (Dempster et al., 1977).

2.2 Expectation Maximization

The strategy underlying the EM algorithm is to separate the original ML problem (4) into two linked problems, each of which is hopefully easier to solve than the original problem. Abstractly speaking this separation is accomplished by exploiting the structure inherent in the probabilistic model. This will hopefully be made clear below.

The *key idea* is to consider the joint log-likelihood function of both the observed variables Y and the latent variables Z

$$L_\theta(Z, Y) = \log p_\theta(Z, Y), \tag{5}$$

and then assume that the latent variables Z were available to us. In order to understand why this makes sense, let us start by an example.

Example 2.1 (Identifying linear state-space models)
Consider the following linear state-space model

$$\begin{pmatrix} x_{t+1} \\ y_t \end{pmatrix} = \begin{pmatrix} A & B \\ B & D \end{pmatrix} \begin{pmatrix} x_t \\ u_t \end{pmatrix} + \begin{pmatrix} v_t \\ e_t \end{pmatrix}, \tag{6}$$

where the noise processes v_t and e_t are assumed to be i.i.d.

$$\begin{pmatrix} v_t \\ e_t \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \right). \tag{7}$$

The latent variables are for this problem given by the states, i.e., $Z = X \triangleq \{x_1, \ldots, x_{N+1}\}$. The problem is now to identify the parameters in the fully parameterized A, B, C and D matrices, based on the measured input u_t and output y_t signals. If we, according to the key idea mentioned above, consider the joint log-likelihood $\log p_\theta(X, Y)$ and assume that the latent variables X are known, the problem breaks down to a linear regression problem,

$$\hat{\theta} = \arg\max_\theta \sum_{t=1}^{N} \left\| \begin{pmatrix} x_{t+1} \\ y_t \end{pmatrix} - \begin{pmatrix} A(\theta) & B(\theta) \\ C(\theta) & D(\theta) \end{pmatrix} \begin{pmatrix} x_t \\ u_t \end{pmatrix} \right\|^2_{\Gamma^{-1}}, \quad \Gamma = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix}, \tag{8}$$

which straightforwardly allows for a closed form solution. The problem is of course that the latent variables X are not known. An idea, allowing us to still use the above idea, would then be to use the available observations in order to find the best possible estimate of the latent variables. This estimate can then be used in solving (8) and hopefully this results in something that is meaningful. As we will soon see the intuition used above is in fact in close agreement with theory.

Using the definition of conditional probability

$$p_\theta(Z \mid Y) \triangleq \frac{p_\theta(Z, Y)}{p_\theta(Y)}, \tag{9}$$

we can establish the following connection between (3) and (5),

$$\log p_\theta(Y) = \log p_\theta(Z, Y) - \log p_\theta(Z \mid Y). \tag{10}$$

Let θ_k denote the estimate of the parameter θ from the k^{th} iteration of the algorithm. The problem separation mentioned above is now obtained by integrating (10) w.r.t. $p_{\theta_k}(Z \mid Y)$, resulting in

$$\begin{aligned}\log p_\theta(Y) &= \int \log p_\theta(Z,Y) p_{\theta_k}(Z \mid Y) dZ - \int \log p_\theta(Z \mid Y) p_{\theta_k}(Z \mid Y) dZ \\ &= \underbrace{\mathrm{E}_{\theta_k}\left\{\log p_\theta(Z,Y) \mid Y\right\}}_{\triangleq \mathcal{Q}(\theta,\theta_k)} - \underbrace{\mathrm{E}_{\theta_k}\left\{\log p_\theta(Z \mid Y) \mid Y\right\}}_{\triangleq \mathcal{V}(\theta,\theta_k)}. \end{aligned} \tag{11}$$

In the above equation we have used the fact that

$$\int \log p_\theta(Y) p_{\theta_k}(Z \mid Y) dZ = \log p_\theta(Y). \tag{12}$$

It is worth noticing that the latent variables are here assumed to be continuous. However, there is nothing that prevents us from deriving the EM algorithm for discrete latent variables, the only difference is that the integrals in (11) will be replaced by summations.

Let us now study the difference between the log-likelihood function $L_\theta(Y)$ evaluated at two different values θ and θ_k,

$$L_\theta(Y) - L_{\theta_k}(Y) = \left(\mathcal{Q}(\theta,\theta_k) - \mathcal{Q}(\theta_k,\theta_k)\right) + \left(\mathcal{V}(\theta_k,\theta_k) - \mathcal{V}(\theta,\theta_k)\right), \tag{13}$$

where we have made use of the definitions in (11). It is now interesting to consider $\mathcal{V}(\theta_k,\theta_k) - \mathcal{V}(\theta,\theta_k)$ in more detail. Straightforward application of the definition of $\mathcal{V}(\theta,\theta_k)$ provided in (11) results in,

$$\begin{aligned}\mathcal{V}(\theta_k,\theta_k) - \mathcal{V}(\theta,\theta_k) &= \int \log\left(\frac{p_{\theta_k}(Z \mid Y)}{p_\theta(Z \mid Y)}\right) p_{\theta_k}(Z \mid Y) dZ \\ &= \mathrm{E}_{\theta_k}\left\{-\log\left(\frac{p_\theta(Z \mid Y)}{p_{\theta_k}(Z \mid Y)}\right) \mid Y\right\}. \end{aligned} \tag{14}$$

Note that this means that $\mathcal{V}(\theta_k,\theta_k) - \mathcal{V}(\theta,\theta_k)$ is the *Kullback-Leibler information distance* (Kullback and Leibler, 1951) between $p_{\theta_k}(Z \mid Y)$ and $p_\theta(Z \mid Y)$. Furthermore, the negative logarithm is a convex function, which implies that Jensen's inequality[2] can be used to establish

$$\begin{aligned}\mathrm{E}_{\theta_k}\left\{-\log\frac{p_\theta(Z \mid Y)}{p_{\theta_k}(Z \mid Y)} \mid Y\right\} &\geq -\log \mathrm{E}_{\theta_k}\left\{\frac{p_\theta(Z \mid Y)}{p_{\theta_k}(Z \mid Y)} \mid Y\right\} \\ &= -\log \int p_\theta(Z \mid Y) dZ = 0, \end{aligned} \tag{16}$$

which effectively proves that

$$\mathcal{V}(\theta_k,\theta_k) - \mathcal{V}(\theta,\theta_k) \geq 0. \tag{17}$$

[2] Jensen's inequality states that if f is a convex function then

$$\mathrm{E}\{f(x)\} \geq f(\mathrm{E}\{x\}), \tag{15}$$

provided that both expectations exist.

Hence, if we make use of this fact in (13) and choose a new parameter θ such that $\mathcal{Q}(\theta, \theta_k) \geq \mathcal{Q}(\theta_k, \theta_k)$, we have in fact also increased the likelihood, or left it unchanged,

$$\mathcal{Q}(\theta, \theta_k) \geq \mathcal{Q}(\theta_k, \theta_k) \quad \Rightarrow \quad L_\theta(Y) \geq L_{\theta_k}(Y). \tag{18}$$

The EM algorithm now suggests itself in that if we start by computing $\mathcal{Q}(\theta, \theta_k)$ according to its definition in (11) this function can then be maximized with respect to θ in order to obtain a new estimate θ_{k+1}. According to the above analysis, this new estimate will indeed produce a higher or at least the same likelihood as the previous estimate θ_k. This procedure is then repeated until convergence, which is summarised in the algorithm below. It is important to note that the convergence is only guaranteed to be to a local minima.

Algorithm 2.1 (*Expectation Maximization*)

1. *Set $k = 0$ and initialize θ_0 such that $L_{\theta_k}(Y)$ is finite.*
2. ***Expectation (E) step:*** *Compute*

$$\mathcal{Q}(\theta, \theta_k) = \mathrm{E}_{\theta_k}\left\{\log p_\theta(Z, Y) \mid Y\right\} = \int \log p_\theta(Z, Y) p_{\theta_k}(Z \mid Y) dZ. \tag{19}$$

3. ***Maximization (M) step:*** *Compute*

$$\theta_{k+1} = \arg\max_\theta \mathcal{Q}(\theta, \theta_k). \tag{20}$$

4. *If not converged, update $k := k + 1$ and return to step 2.*

There are several ways in which the convergence check in step 4 of the above algorithm can be performed. One common way is to simply monitor the value of the log-likelihood and say that the algorithm has converged whenever the increase falls below a certain threshold $\varepsilon_{\mathrm{L}} > 0$ (a typical default value is $\varepsilon_{\mathrm{L}} = 10^{-6}$), i.e.,

$$|L_{\theta_{k+1}}(Y) - L_{\theta_k}(Y)| \leq \varepsilon_{\mathrm{L}}. \tag{21}$$

Another way to check for convergence is to monitor the change in the parameter value between two consecutive iterations and state that the algorithm has converged when

$$\|\theta_{k+1} - \theta_k\|^2 \leq \varepsilon_{\mathrm{P}}, \tag{22}$$

where $\varepsilon_{\mathrm{P}} > 0$ is some suitably chosen threshold.

3 An Illustrative Example

Rather than providing a general solution to the problem of identifying linear state-space models, we will solve the simplest problem we can think of. The motivation for this is simply that it facilitates understanding. The general case is then a generalisation of this, see (Gibson and Ninness, 2005) for details concerning the fully parameterised case.

Let us consider the following scalar linear state-space model,

$$x_{t+1} = \theta x_t + v_t, \tag{23a}$$

$$y_t = \frac{1}{2} x_t + e_t, \tag{23b}$$

where the noise processes are Gaussian according to

$$\begin{pmatrix} v_t \\ e_t \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix} \right). \tag{23c}$$

For simplicity we assume that initial state is fully known, according to $x_1 = 0$. Finally, the true θ-parameter is given by

$$\theta^\star = 0.9. \tag{23d}$$

The identification problem is now to determine the parameter θ on the basis of the observations $Y = \{y_1, \ldots, y_N\}$, using the EM algorithm introduced in the previous section. The first thing to do is to identify the latent variables Z. Inspired by Example 2.1 we realise that the states $X \triangleq \{x_1, \ldots, x_{N+1}\}$ plays the role of latent variables in this problem, due to the fact that if the states were known we could find the parameter simply by solving a linear regression problem.

In the rest of this section we will now provide all the details necessary to device a working EM algorithm to identify the parameter θ in (23). More specifically, the E and the M steps are presented in Section 3.1 and 3.2, respectively. Together, this results in the EM algorithm detailed in Section 3.3. Finally, numerical experiments are provided in Section 3.4.

3.1 The Expectation (E) Step

The expectation (E) step of the algorithm consists in computing the following quantity

$$\mathcal{Q}(\theta, \theta_k) \triangleq \mathrm{E}_{\theta_k} \left\{ \log p_\theta(X, Y) \mid Y \right\} = \int \log p_\theta(X, Y) p_{\theta_k}(X \mid Y) dX. \tag{24}$$

Let us start by finding an expression for $\log p_\theta(X, Y)$, when the data is generated by the probabilistic model given in (23). We have,

$$\begin{aligned} p_\theta(X, Y) &= p_\theta(x_{N+1}, X_N, y_N, Y_{N-1}) \\ &= p_\theta(x_{N+1}, y_N \mid X_N, Y_{N-1}) p_\theta(X_N, Y_{N-1}), \end{aligned} \tag{25}$$

where we have used the definition of conditional probabilities in the second equality. According to the Markov property[3] inherent in (23) we have

$$p_\theta(x_{N+1}, y_N \mid X_N, Y_{N-1}) = p_\theta(x_{N+1}, y_N \mid x_N), \tag{27}$$

[3] A discrete-time stochastic process $\{x_t\}$ is said to possess the *Markov property* if

$$p(x_{t+1} | x_1, \ldots, x_t) = p(x_{t+1} | x_t). \tag{26}$$

implying that (25) can be written as

$$p_\theta(X, Y) = p_\theta(x_{N+1}, y_N \mid x_N) p_\theta(X_N, Y_{N-1}). \tag{28}$$

Repeated use of the above ideas straightforwardly yields

$$p_\theta(X, Y) = p_\theta(x_1) \prod_{t=1}^{N} p_\theta(x_{t+1}, y_t \mid x_t). \tag{29}$$

From (23) we have

$$p_\theta\left(\begin{pmatrix} x_{t+1} \\ y_t \end{pmatrix} \mid x_t\right) = \mathcal{N}\left(\begin{pmatrix} x_{t+1} \\ y_t \end{pmatrix}; \begin{pmatrix} \theta \\ 1/2 \end{pmatrix} x_t, \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}\right), \tag{30}$$

which inserted in (29) results in the following expression for $\log p_\theta(X, Y)$

$$\log p_\theta(X, Y) = p_\theta(x_1) \sum_{t=1}^{N} \log \mathcal{N}\left(\begin{pmatrix} x_{t+1} \\ y_t \end{pmatrix}; \begin{pmatrix} \theta \\ 1/2 \end{pmatrix} x_t, \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}\right). \tag{31}$$

Inserting the expression for the normal density function and using the fact that in the current example we, for simplicity, assumed that the initial state was known, we obtain,

$$\log p_\theta(X, Y) \propto \sum_{t=1}^{N} \log\left(\frac{1}{0.01\sqrt{2\pi}} e^{-\frac{1}{2} w_t}\right) \propto -\frac{1}{2} \sum_{t=1}^{N} w_t \tag{32a}$$

where the exponent w_t is given by

$$w_t = \begin{pmatrix} x_{t+1} - \theta x_t \\ y_t - \frac{1}{2} x_t \end{pmatrix}^T \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}^{-1} \begin{pmatrix} x_{t+1} - \theta x_t \\ y_t - \frac{1}{2} x_t \end{pmatrix}. \tag{32b}$$

The expression (32) can be simplified, resulting in

$$\log p_\theta(X, Y) \propto -\sum_{t=1}^{N} x_t^2 \theta^2 + 2 \sum_{t=1}^{N} x_t x_{t+1} \theta, \tag{33}$$

where terms independent of θ have been neglected, since they, similar to x_1, will not affect the resulting optimization problem. Recall the we are interested in the $\mathcal{Q}$-function, which was defined in (24). Now, inserting (33) into (24) results in

$$\begin{aligned} \mathcal{Q}(\theta, \theta_k) &\propto -\operatorname{E}_{\theta_k}\left\{\sum_{t=1}^{N} x_t^2 \mid Y\right\} \theta^2 + 2 \operatorname{E}_{\theta_k}\left\{\sum_{t=1}^{N} x_t x_{t+1} \mid Y\right\} \theta \\ &= -\varphi \theta^2 + 2\psi\theta, \end{aligned} \tag{34}$$

where we have defined

$$\varphi \triangleq \sum_{t=1}^{N} \operatorname{E}_{\theta_k}\left\{x_t^2 \mid Y\right\}, \qquad \psi \triangleq \sum_{t=1}^{N} \operatorname{E}_{\theta_k}\left\{x_t x_{t+1} \mid Y\right\}. \tag{35}$$

The expected values used to compute φ in (35) are explicitly provided by the Rauch-Tung-Striebel (RTS) smoother (Rauch et al., 1965). Furthermore, the expected values used to compute ψ in (35) are provided by an extension to the RTS formula. All the details are provided in Theorem 1 below.

Theorem 1 *Consider the following linear state-space model*

$$x_{t+1} = Ax_t + Bu_t + v_t, \tag{36a}$$
$$y_t = Cx_t + Du_t + e_t, \tag{36b}$$

where the noise processes are Gaussian according to

$$\begin{pmatrix} v_t \\ e_t \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \right) \tag{36c}$$

and the initial state is distributed according to $x_1 \sim \mathcal{N}(\mu, P_1)$. *Let the parameter vector* θ *be composed of* A, B, C, D, Q, R, S, P_1 *and* μ. *Then*

$$\mathrm{E}_{\theta_k}\left\{ x_t x_t^T \mid Y \right\} = \widehat{x}_{t|N}\widehat{x}_{t|N}^T + P_{t|N}, \tag{37a}$$
$$\mathrm{E}_{\theta_k}\left\{ x_{t+1} x_t^T \mid Y \right\} = \widehat{x}_{t+1|N}\widehat{x}_{t|N}^T + M_{t+1|N}, \tag{37b}$$
$$\mathrm{E}_{\theta_k}\left\{ y_t x_t^T \mid Y \right\} = y_t\widehat{x}_{t|N}^T, \tag{37c}$$

where

$$\widehat{x}_{t|N} = \widehat{x}_{t|t} + J_t\left(\widehat{x}_{t+1|N} - \bar{A}\widehat{x}_{t|t} - \bar{B}u_t - SR^{-1}y_t\right), \tag{38a}$$
$$P_{t|N} = P_{t|t} + J_t\left(P_{t+1|N} - P_{t+1|t}\right)J_t^T, \tag{38b}$$
$$J_t = P_{t|t}\bar{A}^T P_{t+1|t}^{-1}, \tag{38c}$$

for $t = N, \ldots, 1$ *and*

$$M_{t|N} = P_{t|t}J_{t-1}^T + J_t(M_{t+1|N} - \bar{A}P_{t|t})J_{t-1}^T, \tag{39a}$$

for $t = N-1, \ldots, 1$ *and* $M_{N|N}$ *is initialized according to*

$$M_{N|N} = (I - K_N C)\bar{A}P_{N-1|N-1}. \tag{39b}$$

Finally, the quantities $\widehat{x}_{t|t}, P_{t|t}$ *and* $P_{t+1|t}$ *are provided by the Kalman filter for the system given by*

$$\bar{A} = A - SR^{-1}C, \tag{40a}$$
$$\bar{B} = B - SR^{-1}D, \tag{40b}$$
$$\bar{Q} = Q - SR^{-1}S^T. \tag{40c}$$

Proof 1 *We refrain from giving the proof here, instead we refer to Shumway and Stoffer (2006).*

3.2 The Maximization (M) Step

According to (20), the maximization (M) step amounts to solving the following problem

$$\theta_{k+1} = \arg\max_{\theta} \mathcal{Q}(\theta, \theta_k). \tag{41}$$

In the previous section we derived the following expression for $\mathcal{Q}(\theta, \theta_k)$

$$\mathcal{Q}(\theta, \theta_k) = -\varphi\theta^2 + 2\psi\theta, \tag{42}$$

where φ and ψ are defined in (35). Hence, the M step simply amounts to solving the following quadratic problem,

$$\theta_{k+1} = \arg\max_{\theta} \; -\varphi\theta^2 + 2\psi\theta, \tag{43}$$

which results in

$$\theta_{k+1} = \frac{\psi}{\varphi}. \tag{44}$$

3.3 Explicit EM Algorithm

The final algorithm is now obtained simply by inserting the results derived in Section 3.1 and Section 3.2 into the general EM algorithm provided in Algorithm 2.1. This results in the algorithm provided below.

Algorithm 3.1 (*Expectation Maximization for** (23)**)

1. *Set $k = 0$ and initialize θ_0.*
2. ***Expectation (E) step:** Compute*

$$\mathcal{Q}(\theta, \theta_k) = -\operatorname{E}_{\theta_k}\left\{\sum_{t=1}^{N} x_t^2 \mid Y\right\}\theta^2 + 2\operatorname{E}_{\theta_k}\left\{\sum_{t=1}^{N} x_t x_{t+1} \mid Y\right\}\theta, \tag{45}$$

   where the involved expected values are computed according to Theorem 1.

3. ***Maximization (M) step:** Find the next iterate according to*

$$\theta_{k+1} = \frac{\operatorname{E}_{\theta_k}\left\{\sum_{t=1}^{N} x_t x_{t+1} \mid Y\right\}}{\operatorname{E}_{\theta_k}\left\{\sum_{t=1}^{N} x_t^2 \mid Y\right\}} \tag{46}$$

4. *If $|L_{\theta_k}(Y) - L_{\theta_{k-1}}(Y)| \geq 10^{-6}$, update $k := k + 1$ and return to step 2, otherwise terminate.*

All the details of the above algorithm are now accounted for, save for how to compute the log-likelihood function $L_\theta(Y)$, which is needed for the convergence check. In the interest of a self-contained presentation we provide a derivation of an explicit expression for $L_\theta(Y)$ in Appendix A and in Appendix B, the MATLAB code is available.

3.4 Numerical Experiments

In the previous sections we derived an EM algorithm for estimating the unknown parameter θ in (23a). We know that the maximum likelihood method is asymptotically consistent and since (23) is a linear model we expect the estimate to converge to the true value as the number of samples N tends to infinity. Before we show that this is indeed the case for Algorithm 3.1, let us explain the experimental conditions. The experiment is made by conducting seven Monte Carlo studies, each using 1000 realisations of data Y. The only difference between the studies is that the number of samples N used is different for each study. All

Table 1: *The table shows the EM estimate provided by Algorithm 3.1 for seven different N. More specifically, $\hat{\theta}$ is the result of Monte Carlo studies, each using* 1000 *realisations of data. Recall that the true parameter value is $\theta^\star = 0.9$.*

N	100	200	500	1000	2000	5000	10000
$\hat{\theta}$	0.8716	0.8852	0.8952	0.8978	0.8988	0.8996	0.8998

information concerning the model is provided in (23) and the initial guess used for θ is chosen as $\theta_0 = 0.1$ for all realisations.

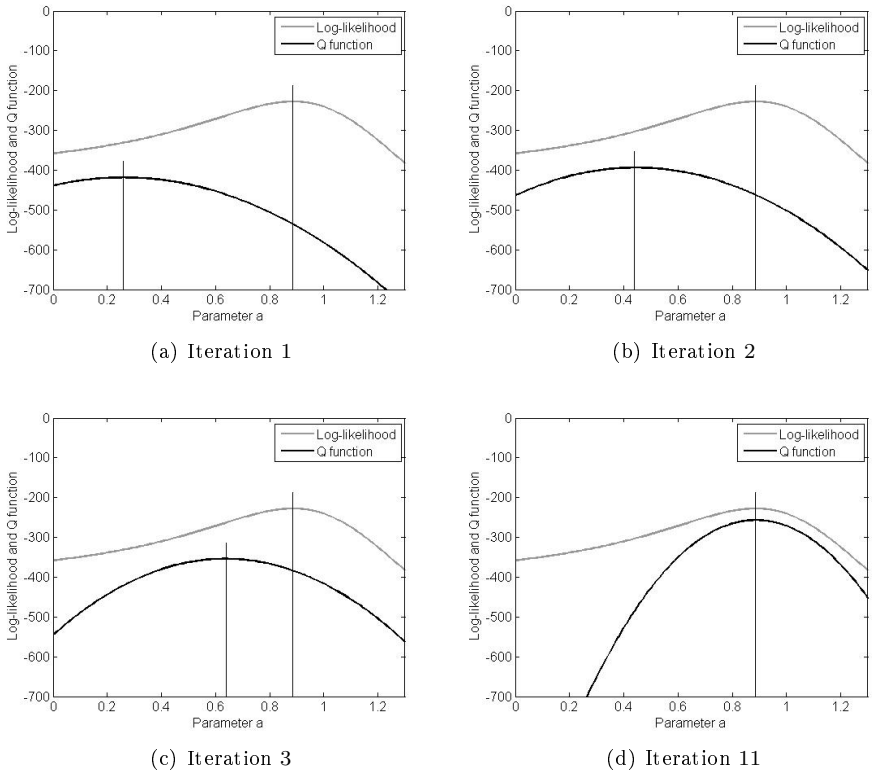
The result is provided in Table 1, where the EM estimate $\hat{\theta}$ is shown for the different number of samples N used. As expected, the EM estimate approaches the true value $\theta^\star$ as the number of samples N increase. In order to further illustrate how the EM algorithm works we have provided plots of the log-likelihood $L_\theta(Y)$ and the $\mathcal{Q}$-function in Figure 1. In the expectation (E) step, that is step 2, an expression for the $\mathcal{Q}$-function is computed according to the details provided in Section 3.1 and Appendix A.2. The result of this computation is provided by the black curve in Figure 1. In the subsequent step, the maximization (M) step, of the EM algorithm, the task is to find the θ that maximizes the $\mathcal{Q}$-function. The result of this is illustrated by a vertical line in the figure. The grey curve in Figure 1 is the corresponding log-likelihood, computing according to Appendix A.1.

4 Bibliography

To the best of the author's knowledge, the earliest account of an EM type algorithm is Newcomb (1886), who used it for estimating a mixture model. Since then there have been quite a few papers on EM type algorithms. However, the algorithm is usually accredited to Dempster et al. (1977), where the first systematic treatment of the method is provided and the name expectation maximization was coined. Currently, the most complete account of the EM algorithm is provided by McLachlan and Krishnan (2008), which also provides a good historical account of the algorithm.

As already mentioned in the introduction, the main focus in this work has been the use of the EM algorithm for estimation problems arising in dynamic systems, i.e., *system identification*. A solid overview of the use of the EM algorithm for system identification was provided by Ninness (2009) in his plenary talk at the 15th IFAC Symposium on System Identification held in Saint-Malo, France in July 2009. Since the EM algorithm offers a solution to the general problem of maximum likelihood estimation it is widely used within many disciplines of science, see McLachlan and Krishnan (2008) for many examples of this. Before we provide a brief overview of what has been done when it comes to system identification it is interesting to note that the algorithm is very popular in the neighbouring areas of machine learning, see e.g., (Bishop, 2006), robotics, see e.g., (Thrun et al., 2005) and signal processing, see e.g., (Moon, 1996).

Let us now briefly review what has been done within the system identification community when it comes to using the EM algorithm. When it comes to identifying linear state-space models the work of Gibson and Ninness (2005)



(a) Iteration 1 (b) Iteration 2

(c) Iteration 3 (d) Iteration 11

Figure 1: *Four instances of the log-likelihood $L_\theta(Y)$ and the $\mathcal{Q}$-function for estimating the parameter θ in* (23) *using Algorithm 3.1. More specifically, the different plots show the log-likelihood and the $\mathcal{Q}$-function as a function of the parameter θ at iterations $k = 1, k = 2, k = 3$ and $k = 11$ of Algorithm 3.1. The vertical lines corresponds to the estimate computed at iteration k and the true parameter value, respectively.*

provides most of the necessary details. The earlier work of Shumway and Stoffer (1982) is also worth mentioning here. An early application of the EM algorithm for solving system identification problems is provided by Isaksson (1993). Bilinear systems are discussed by Gibson et al. (2005). There is also a very useful toolbox available for using EM to identify dynamic systems (Ninness and Wills, 2009a,b). There are by now also quite a few approaches for nonlinear system identification using the EM algorithm. In order to handle the nonlinear problem more approximations are needed. There are several suboptimal solutions available, where the nonlinear smoothing problem is approximated using an extended Kalman smoother, see e.g., (Ghaharamani and Roweis, 1999; Roweis and Ghaharamani, 2001; Duncan and Gyöngy, 2006). There is also a recent approach, where the theoretically appealing particle smoothers are employed (Schön et al., 2009, 2006; Wills et al., 2008). An interesting extension, handling the case of missing observations is discussed by Gopaluni (2008). Finally, there are recent discussions on possible embellishments based on variational inference, as discussed by Tzikas et al. (2008) and Bishop (2006).

5 Conclusion

The aim of this work has been to show how the expectation maximization algorithm can be used to estimate unknown parameters in dynamic systems, that is how it can be used to solve certain system identification problems. We did this by first showing that the expectation maximization algorithm is an iterative method for computing maximum likelihood estimates of unknown parameters in probabilistic models involving latent variables. In order to make the material as accessible as possible we provided all the details (in terms of the necessary derivations and the final MATLAB code) for a simple system identification problem.

6 Acknowledgements

I would like to thank the machine learning reading group here at the Division of Automatic Control for inspiring me to write these notes. Furthermore, I would like to thank for all the constructive comments and ideas kindly provided during the process of writing these notes.

A Explicit Expressions for L and $\mathcal{Q}$

This appendix provides derivations of explicit expressions for the log-likelihood function and the $\mathcal{Q}$-function used in the example in Section 3, repeated below for convenience,

$$\begin{aligned} x_{t+1} &= \theta x_t + v_t, \\ y_t &= \tfrac{1}{2} x_t + e_t, \end{aligned} \qquad \begin{pmatrix} v_t \\ e_t \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix} \right), \tag{47}$$

where the initial state is assumed known according to $x_1 = 0$.

A.1 Log-Likelihood L

Let us start with the log-likelihood, which according to (3) is given by

$$L_\theta(Y) = \sum_{t=2}^{N} \log p_\theta(y_t \mid Y_{t-1}) + \log p_\theta(y_1). \tag{48}$$

We have that

$$p_\theta(y_t \mid Y_{t-1}) = \mathcal{N}\left(y_t \,;\, \frac{1}{2}\hat{x}_{t|t-1}, \frac{1}{4}P_{t|t-1} + 0.1 \right), \tag{49a}$$

$$p_\theta(y_1) = \mathcal{N}\left(y_1 \,;\, \frac{1}{2}\hat{x}_{1|0}, \frac{1}{4}P_{1|0} + 0.1 \right), \tag{49b}$$

where $\hat{x}_{t|t-1}$ and $P_{t|t-1}$ are both provided by the Kalman filter and $\hat{x}_{1|0}$ and $P_{1|0}$ are the initial guesses for the state variable and its covariance. Now inserting (49) into (48) results in

$$L_\theta(Y) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{N} \log\left(\frac{1}{4}P_{t|t-1} + 0.1 \right) - \sum_{t=1}^{N} \frac{\left(y_t - \frac{1}{2}\hat{x}_{t|t-1} \right)^2}{\frac{1}{2}P_{t|t-1} + 0.2} \tag{50}$$

A.2 $\mathcal{Q}$-Function

According to (24) and (29) the $\mathcal{Q}$-function is given by

$$\mathcal{Q}(\theta, \theta_k) = \int \log p_\theta(x_1) \prod_{t=1}^{N} p_\theta(x_{t+1}, y_t \mid x_t) p_{\theta_k}(X \mid Y) dX, \tag{51}$$

where

$$p_\theta(x_{t+1}, y_t \mid x_t) = p_\theta(x_{t+1} \mid x_t) p_\theta(y_t \mid x_t). \tag{52}$$

Inserting (52) into (51) together with the fact that x_1 is fully known results in

$$\begin{aligned}
\mathcal{Q}(\theta, \theta_k) &= \int \log \left(\prod_{t=1}^{N} p_\theta(x_{t+1} \mid x_t) p_\theta(y_t \mid x_t) \right) p_{\theta_k}(X \mid Y) dX \\
&= \int \sum_{t=1}^{N} \log \left(\frac{1}{(2\pi)^{1/2}\sqrt{0.1}} e^{-\frac{1}{0.2}(x_{t+1} - \theta x_t)^2} \right) p_{\theta_k}(X \mid Y) dX \\
&+ \int \sum_{t=1}^{N} \log \left(\frac{1}{(2\pi)^{1/2}\sqrt{0.1}} e^{-\frac{1}{0.2}(y_t - \frac{1}{2} x_t)^2} \right) p_{\theta_k}(X \mid Y) dX \\
&= \int \sum_{t=1}^{N} \left(-\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(0.1) - 5(x_{t+1} - \theta x_t)^2 \right) p_{\theta_k}(X \mid Y) dX \\
&+ \int \sum_{t=1}^{N} \left(-\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(0.1) - 5(y_t - \frac{1}{2} x_t)^2 \right) p_{\theta_k}(X \mid Y) dX \\
&= -N\log(0.2\pi) - 5\sum_{t=1}^{N} \int (x_{t+1}^2 - 2\theta x_{t+1} x_t + \theta^2 x_t^2) p_{\theta_k}(X \mid Y) dX \\
&- 5\sum_{t=1}^{N} \int (y_t^2 - y_t x_t + \frac{1}{4} x_t^2) p_{\theta_k}(X \mid Y) dX \\
&= -N\log(0.2\pi) - 5\alpha + 10\theta\psi - 5\theta^2\sigma - 5\sum_{t=1}^{N} y_t^2 + 5\beta - \frac{5}{4}\sigma,
\end{aligned} \tag{53}$$

where we have defined

$$\alpha \triangleq \sum_{t=1}^{N} \int x_{t+1}^2 p_{\theta_k}(X \mid Y) dX, \tag{54a}$$

$$\beta \triangleq \sum_{t=1}^{N} \int x_t y_t p_{\theta_k}(X \mid Y) dX, \tag{54b}$$

$$\psi \triangleq \sum_{t=1}^{N} \int x_{t+1} x_t p_{\theta_k}(X \mid Y) dX, \tag{54c}$$

$$\sigma \triangleq \sum_{t=1}^{N} \int x_t^2 p_{\theta_k}(X \mid Y) dX. \tag{54d}$$

B MATLAB Code for the Numerical Example

This appendix provides the MATLAB code for solving the problem studied in Section 3. In order to run the code below the UNIT system identification toolbox, available from `http://sigpromu.org/idtoolbox/` has to be installed first. The toolbox is used for computing the necessary expected values (35).

Listing 1: *EM code for the example*

```
addpath('H:\Matlab\unit');      % Add path to UNIT toolbox

opt.miter = 100;     % Max number of EM iterations
opt.LLdec = 1e-6;    % Min decrease of log-likelihood
theta0 = 0.1;        % Initial guess for the parameter

%=================================================
%===   Simulate the one dimensional state-space model   ===
%=================================================
N = 500;             % Number of data
m.ss.A = 0.9;        m.ss.B = 0;
m.ss.C = 0.5;        m.ss.D = 0;
m.ss.Q = 0.1;        m.ss.S = 0;
m.ss.R = 0.1;

m.ss.X1 = 0;         m.ss.P1 = 0;  % Initial state (fully known)
x = zeros(1,N+1); y = zeros(1,N);
x(1) = m.ss.X1;
v = sqrt(m.ss.Q)*randn(1,N);      % Process noise sequence
e = sqrt(m.ss.R)*randn(1,N);      % Measurement noise sequence
for t=1:N
  x(t+1) = m.ss.A*x(t) + v(t);
  y(t)   = m.ss.C*x(t) + e(t);
end
z.y = y;
mEst = m;
mEst.ss.A = theta0;      % Initial parameter guess
%======================
%===   EM algorithm   ===
%======================
for k = 1:opt.miter
  % E step
  g      = ks(z,mEst,opt);      % Smoother
  LL(k) = -0.5*g.LL;            % Store the log-likelihood
  phi    = 0;    psi = 0;    sigma = 0;
  for t = 1:N
    phi   = phi + g.ss.xs(t+1)*g.ss.xs(t+1) + g.ss.Ps(1,1,t+1)^2;
    sigma = sigma + g.ss.xs(t).^2 + g.ss.Ps(1,1,t)^2;
    psi   = psi + g.ss.xs(t)*g.ss.xs(t+1) + g.ss.Ms(1,1,t);
  end;
  % M step
  a(k)       = psi/sigma;       % Store result
  mEst.ss.A = a(k);             % Update model

  % Compute the Q-function for the current estimate
  alpha = phi;
  beta  = 0;
  for t=1:N
    beta = beta + z.y(t)*g.ss.xs(t);
  end;
  aNow = [0:0.01:0.8  0.8:0.001:1  1.01:0.01:1.3];
  for i = 1:length(aNow)
    Q1       = -N*log(0.2*pi) - 5*sum(z.y.^2) - 5*alpha;
```

```
        Q2      = 5*beta + 10*psi*aNow(i) - 5*sigma*(aNow(i)^2+1/4);
        Q{k}(i) = Q1 + Q2;
    end;

    if k>1          % Check termination condition
      if LL(k)-LL(k-1) < opt.LLdec
        break;
      end;
    end;
  end;
  a = [theta0 a];

% Compute the log-likelihood function
for i = 1:length(aNow)
  m.ss.A = aNow(i);
  g      = kf(z,m,opt);       % Kalman filter
  Ltmp   = -(N/2)*log(2*pi);
  for t = 2:N
    CovInn   = g.ss.Ri(1,1,t)^2;
    Lpart1   = -(1/2)*((z.y(t) - m.ss.C*g.ss.xp(t))^2)/CovInn;
    Lpart2   = -(1/2)*log(det(CovInn));
    Ltmp     = Ltmp + Lpart1 + Lpart2;
  end;
  LL(i) = Ltmp;
end;
```

References

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, USA.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Dennis, J. E. and Schnabel, R. B. (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice Hall, Englewood Cliffs, NJ, USA.

Duncan, S. and Gyöngy, M. (2006). Using the EM algorithm to estimate the disease parameters for smallpox in 17th century London. In *Proceedings of the IEEE international conference on control applications*, pages 3312–3317, Munich, Germany.

Fisher, R. A. (1912). On an absolute criterion for fitting frequency curves. *Messenger of Mathematics*, 41:155–160.

Fisher, R. A. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society Series A*, 222:309–368.

Ghaharamani, Z. and Roweis, S. T. (1999). Learning nonlinear dynamical systems using an EM algorithm. In *Advances in Neural Information Processing Systems*, volume 11, pages 599–605. MIT Press.

Gibson, S. and Ninness, B. (2005). Robust maximum-likelihood estimation of multivariable dynamic systems. *Automatica*, 41(10):1667–1682.

Gibson, S., Wills, A., and Ninness, B. (2005). Maximum-likelihood parameter estimation of bilinear systems. *IEEE Transactions on Automatic Control*, 50(10):1581–1596.

Gopaluni, R. B. (2008). Identification of nonlinear processes with known model structure using missing observations. In *Proceedings of the 17th IFAC World Congress*, Seoul, South Korea.

Isaksson, A. (1993). Identification of ARX-models subject to missing data. *IEEE Transactions on Automatic Control*, 38(5):813–819.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

McLachlan, G. and Krishnan, T. (2008). *The EM Algorithm and Extensions*. Whiley Series in Probability and Statistics. John Wiley & Sons, New York, USA, second edition.

Moon, T. K. (1996). The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60.

Newcomb, S. (1886). A generalized theory of the combination of observations so as to obtain the best result. *American Journal of Mathematics*, 8(4):343–366.

Ninness, B. (2009). Some system identification challenges and approaches. In *Plenary talk at the 15th IFAC Symposium on System Identification (SYSID)*, Saint-Malo, France.

Ninness, B. and Wills, A. (2009a). An identification toolbox for profiling novel techniques. *Submitted to Automatica*.

Ninness, B. and Wills, A. (2009b). System identification toolbox. Available from `http://sigpromu.org/idtoolbox/`.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, USA, 2 edition.

Rauch, H. E., Tung, F., and Striebel, C. T. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450.

Roweis, S. T. and Ghaharamani, Z. (2001). *Kalman filtering and neural networks*, chapter 6. Learning nonlinear dynamical systems using the expectation maximization algorithm, Haykin, S. (ed), pages 175–216. John Wiley & Sons.

Schön, T. B., Wills, A., and Ninness, B. (2006). Maximum likelihood nonlinear system estimation. In *Proceedings of the 14th IFAC Symposium on System Identification*, pages 1003–1008, Newcastle, Australia.

Schön, T. B., Wills, A., and Ninness, B. (2009). System identification of nonlinear state-space models. *Submitted to Automatica*.

Shumway, R. H. and Stoffer, D. S. (1982). An approach to time series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis*, 3(4):253–264.

Shumway, R. H. and Stoffer, D. S. (2006). *Time Series Analysis and Its Applications*. Springer Texts in Statistics. Springer, New York, USA, 2 edition.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents. The MIT Press, Cambridge, MA, USA.

Tzikas, D. G., Likas, A. C., and Galatsanos, N. P. (2008). The variational approximation for Bayesian inference. *IEEE Signal Processing Magazine*, 25(6):131–146.

Wills, A., Schön, T. B., and Ninness, B. (2008). Parameter estimation for discrete-time nonlinear systems using EM. In *Proceedings of the 17th IFAC World Congress*, Seoul, South Korea.