Expectation Maximization Segmentation

Niclas Bergman

Department of Electrical Engineering
Linköping University, S-581 83 Linköping, Sweden
WWW: http://www.control.isy.liu.se
Email: niclas@isy.liu.se

October 21, 1998

Report no.: LiTH-ISY-R-2067
Submitted to (Nothing)

Technical reports from the Automatic Control group in Linköping are available by anonymous ftp at the address ftp.control.isy.liu.se. This report is contained in the compressed postscript file 2067.ps.Z.
Expectation Maximization Segmentation

Niclas Bergman

October 21, 1998

1 Introduction

This report reviews the Expectation Maximization (EM) algorithm and applies it to the data segmentation problem, yielding the Expectation Maximization Segmentation (EMS) algorithm. The EMS algorithm requires batch processing of the data and can be applied to mode switching or jumping linear dynamical state space models. The EMS algorithm consists of an optimal fusion of fixed interval Kalman smoothing and discrete optimization.

The next section gives a short introduction to the EM algorithm with some background and convergence results. In Section 3 the data segmentation problem is defined and in Section 4 the EM algorithm is applied to this problem. Section 5 contains simulation results and Section 6 some conclusive remarks.

2 The EM algorithm

The Expectation Maximization algorithm [5] is a multiple pass batch algorithm for parameter estimation where some part of the measurements are unknown. The algorithm can either be used for maximum likelihood estimation or in a Bayesian framework to obtain maximum a posteriori estimates. The parametric, maximum likelihood, formulation dates back to [2].

Consider the maximum likelihood framework where we seek the parameter vector \( \theta \in \Omega \) that maximizes the likelihood \( p(y \mid \theta) \) of the observed measurement vector \( y \). Often this function has a rather complicated structure and the maximization is hard to perform. The idea behind the EM algorithm is that there might be some hidden or unobserved data \( x \) that, if it were known, would yield a likelihood function \( p(y, x \mid \theta) \) which is much easier to maximize with respect to \( \theta \in \Omega \). Introduce the notation \( z = [y, x] \) for the complete data where \( y \) are the actual measurements and \( x \) are the unobserved measurements. Note that in some applications \( x \) is commonly referred to as parameters. Since

\[
p(z \mid \theta) = p(x \mid y, \theta) p(y \mid \theta)
\]

the log likelihood splits into two terms

\[
\log p(y \mid \theta) = \log p(z \mid \theta) - \log p(x \mid y, \theta).
\]

Integrating both sides with respect to a measure \( f(x) \) such that \( \int f(x) dx = 1 \) will not affect the left hand side,

\[
\log p(y \mid \theta) = \int \log p(z \mid \theta) f(x) dx - \int \log p(x \mid y, \theta) f(x) dx.
\]


Choosing \( f(x) \) as the conditional density of \( x \) given \( y \) and some candidate parameter vector \( \theta' \) yields

\[
\log p(y \mid \theta) = \mathbb{E}\left( \log p(z \mid \theta, y, \theta') \right) - \mathbb{E}\left( \log p(x \mid y, \theta) \right).
\]

(1)

The EM algorithm only considers the first term, it is defined as alternating between forming \( Q(\theta, \theta') \) and maximizing it with respect to its first argument. Initializing with some \( \theta_0 \), one pass of the algorithm is defined as

\[
Q(\theta, \theta) = \mathbb{E}\left( \log p(z \mid \theta, y, \theta) \mid y, \theta \right) \quad \text{(E-step)}
\]

\[
\theta_{p+1} = \arg \max_{\theta} Q(\theta, \theta) \quad \text{(M-step)}
\]

The algorithm keeps alternating between expectation and maximization until no significant improvement of \( Q(\theta_{p+1}, \theta_p) \) is observed in two consecutive iterations. A fundamental property of the EM algorithm is that for each pass through the algorithm the log likelihood (1) will monotonically increase. We show briefly below a derivation of this claim, the results have been borrowed from [5].

**Lemma 1.**

\[ \mathcal{H}(\theta, \theta') \leq \mathcal{H}(\theta', \theta') \]

**Proof.** Using Jensen’s inequality [4] we have that

\[ \mathcal{H}(\theta', \theta') - \mathcal{H}(\theta, \theta) = \mathbb{E} \left( -\log \frac{p(x \mid y, \theta)}{p(x \mid y, \theta')} \mid y, \theta \right) \geq -\log \mathbb{E} \left( \frac{p(x \mid y, \theta)}{p(x \mid y, \theta')} \mid y, \theta \right) \]

\[ = -\log \int p(x \mid y, \theta) p(x \mid y, \theta') p(x \mid y, \theta') dx = -\log \int p(x \mid y, \theta) dx = 0 \]

since \(-\log(x)\) is convex. Equality is obtained above whenever \( p(x \mid y, \theta') \propto p(x \mid y, \theta) \) where the proportionality constant must be one in since both densities must integrate to unity.

**Theorem 1.** For any \( \theta_0 \in \Omega \)

\[ p(y \mid \theta_{p+1}) \geq p(y \mid \theta_p) \quad p = 0, 1, \ldots \]

with equality if and only if both

\[ Q(\theta_{p+1}, \theta_p) = Q(\theta_p, \theta_p) \]

and

\[ p(x \mid y, \theta_{p+1}) = p(x \mid y, \theta_p) \]

**Proof.** From (1) we have that

\[ p(y \mid \theta_{p+1}) - p(y \mid \theta_p) = Q(\theta_{p+1}, \theta_p) - Q(\theta_p, \theta_p) + \mathcal{H}(\theta_p, \theta_p) - \mathcal{H}(\theta_{p+1}, \theta_p). \]

Since \( \theta_{p+1} \) in the EM algorithm is the maximizing argument of \( Q(\theta, \theta_p) \) the difference of \( Q \) functions is non-negative. By Lemma 1 the difference of \( \mathcal{H} \) functions is positive with equality if and only if \( p(x \mid y, \theta_{p+1}) = p(x \mid y, \theta_p) \).
The theorem proves that the log likelihood function will increase at each iteration of the EM algorithm. Assuming that the likelihood function is bounded for all $\theta \in \Omega$ the algorithm yields a bounded monotonically increasing sequence of likelihood values and thus it must converge to a fixed point where the conditions for quality given in Theorem 1 are met. Let $\theta^*$ be a maximum likelihood (ML) estimate of $\theta$ such that $p(y | \theta^*) \geq p(y | \theta)$ for all $\theta \in \Omega$, then it follows that $\theta^*$ is a fixed point of the EM algorithm. Adding some regulatory conditions one can also prove the converse statement, that the fixed points of the EM algorithm are in fact ML estimates at least in a local sense. One can also derive expressions for the rate of convergence of the EM algorithm, the details can be found in [5].

In general the statement about the convergence of the EM algorithm is that it converges to a local maxima of the likelihood function $p(y | \theta)$.

The discussion above hold equally true under a Bayesian framework where the maximum a posterior (MAP) estimate is considered instead of the ML estimate. We simply replace the log likelihood by the posterior density

$$p(\theta | z) = \frac{p(z, \theta)}{p(z)} = \frac{p(z | \theta)p(\theta)}{p(z)}$$

and since the maximization is with respect to $\theta$ both

$$Q(\theta, \theta_p) = \mathbb{E}(\log p(z | \theta) + \log p(\theta) | y, \theta_p)$$

and

$$Q(\theta, \theta_p) = \mathbb{E}(\log p(z, \theta) | y, \theta_p)$$

yields an EM algorithm for which MAP estimates are fixed points.

Finally we note that the class of likelihoods used in [5] is more general than the one used here. In the original article the measured data $y$ is seen through a many-to-one mapping from the complete data set $z$. Each measured value $y$ then defines a set $Z(y)$ in the complete data sample space. In order not to clutter the presentation in this text we have considered the simplified case when the complete data set can be divided into two disjoint parts, one observed and one unobserved.

3 The Data Segmentation Problem

Consider a linear Markovian state space model for the sought $n_x$-dimensional parameter vector $x_k$ where the dynamics depend on the unknown binary segmentation sequence $\delta_k \in \{0, 1\}$,

$$x_{k+1} = F_{k, \delta} x_k + G_{k, \delta} w_{k, \delta}$$

$$y_k = H_{k, \delta} x_k + e_{k, \delta}$$

$k = 1, 2, \ldots, N$

(2)

where $y_k \in \mathbb{R}^{n_y}$ and $w_{k, \delta} \in \mathbb{R}^{n_w}$. In general $n_x \geq n_y$ and $n_x \geq n_u$. The noises $\{w_{k, \delta}\}$ and $\{e_{k, \delta}\}$ are two independent i.i.d. sequences of zero mean Gaussian random variables with known full rank covariances $Q_{k, \delta}$ and $R_{k, \delta}$, respectively. The notation above indicates that the system matrices have a known time dependency as well as that they depend on the current segmentation parameter $\delta_k$. In order to obtain segments of uncorrelated states one can use

$$F_{k, \delta} = (1 - \delta_k) F_k.$$
Often the segmentation only affects the process noise covariance, e.g., increasing it with a scalar fudge factor $\gamma \gg 0$ at the change instant

$$Q_{k,\delta} = (1 - \delta_k)Q_k + \gamma \delta_k Q_k.$$  \hfill (3)

The explicit time dependent system matrices in (2) may, e.g., be due to local linearizations along the estimated state trajectory. The algorithm described in this work apply to general $r$-valued segmentations, i.e., for $\delta_k \in \{0, 1, \ldots, r\}$, and hence the model (2) can be seen as a mode jumping or switching model.

The initial state $x_1$ in (2) is Gaussian with known mean $\hat{x}_1$ and full rank covariance matrix $P_1$, and the noise sequences $\{w_{k,\delta}\}$, $\{e_{k,\delta}\}$ and the initial state $x_1$ are mutually uncorrelated.

In the data segmentation problem a length $N$ batch of measurement data is available for off-line processing in order to determine the corresponding length $N$ segmentation sequence,

$$Y = [y_1^T, y_2^T, \ldots, y_N^T]^T \quad \Delta = [\delta_1, \delta_2, \ldots, \delta_N]^T.$$  

In a parametric framework this involves maximizing the likelihood $p(Y \mid \Delta)$, and in a stochastic framework the parameter vector that maximizes the posterior probability density $p(\Delta \mid Y)$ is sought. The recursive counterpart of segmentation is usually referred to as detection. One way to introduce recursive processing is to extend the computational horizon and use fixed lag batch processing instead.

If a pure Bayesian framework is considered the prior distribution of the segmentation sequence need also be specified. Two convenient choices for prior distribution are to model the sequence $\{\delta_k\}$ as independent Bernoulli variables,

$$\delta_k = \begin{cases} 0 & \text{with probability } q \\ 1 & \text{with probability } 1 - q \end{cases} \quad k = 1, 2, \ldots, N$$  \hfill (4)

or to assume that the total number of jump instants $n$ in a length $N$ data set is Poisson distributed with intensity $\lambda$,

$$p_n(m) = \frac{\lambda^m}{m!} \exp(-\lambda) \quad m = 0, 1, 2, \ldots$$  \hfill (5)

The key feature with both these priors is that they have the advantage of simplifying the maximization step of the EM algorithm. With a general prior the computational complexity grows exponentially with the data set size.

4 Applying Expectation Maximization to Data Segmentation

As described in Section 1, the EM algorithm defines an iterative scheme for finding ML or MAP estimates of a parameter $\theta$ given some measurements $y$. In the data segmentation problem this corresponds to determining the set of segmentation parameters $\Delta$ using the gathered data set $Y$. Bold capital letters are used to denote stacked vectors of $N$ random variables from (2)

$$X = [x_1^T, x_2^T, \ldots, x_N^T]^T \quad U = [w_1^T, w_{1,\delta}^T, \ldots, w_{N-1,\delta}^T]^T \quad E = [e_1^T, e_{2,\delta}^T, \ldots, e_{N,\delta}^T]^T$$

4
where the noise vectors $U$ and $E$ explicitly depend on the segmentation sequence $\Delta$. Using this notation, the estimation model (2) can be written compactly as

$$
\begin{align*}
X &= A_{\Delta} U \\
Y &= B_{\Delta} X + E
\end{align*}
$$

where matrices $A_{\Delta}$ and $B_{\Delta}$ are block matrices built up by the state space matrices of the model (2). Direct inspection yields that

$$
A_{\Delta} = \begin{bmatrix}
I & 0 & \cdots & 0 \\
W_1^1 & I & 0 & \cdots & 0 \\
W_2^1 & W_1^2 & I & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
W_N^{N-1} & \cdots & \cdots & \cdots & W_N^{N-1} & I
\end{bmatrix}
$$

where zeros denote zero matrices of appropriate dimension, and the state transition matrix is defined as

$$
W_k^l = F_{k,\delta} F_{l-1,\delta} \cdots F_{k,\delta}.
$$

Furthermore,

$$
B_{\Delta} = \text{diag}(H_1, H_2, \ldots, H_N)
$$

where $\text{diag}(A_1, A_2, \ldots, A_M)$ denotes a matrix with blocks $A_i$ along the diagonal. The statistical properties of the noises in (6) are compactly given in the expression

$$
E\left\{ \begin{bmatrix} U \\ E \end{bmatrix} \begin{bmatrix} U^T & E^T & 1 \end{bmatrix} \right\} = \begin{bmatrix} Q_{\Delta} & 0 & \bar{\epsilon} \\
0 & R_{\Delta} & 0 \end{bmatrix}
$$

where both $Q_{\Delta}$ and $R_{\Delta}$ are block diagonal,

$$
Q_{\Delta} = \text{diag}(P_1, Q_{1,\delta}, Q_{2,\delta}, \ldots, Q_{N-1,\delta})
$$

$$
R_{\Delta} = \text{diag}(R_{1,\delta}, R_{2,\delta}, \ldots, R_{N,\delta}).
$$

Eliminating the states $X$ in (6) yields

$$
Y = B_{\Delta} A_{\Delta} U + E,
$$

the Expectation Maximization Segmentation (EMS) algorithm is obtained by treating the segmentation sequence $\Delta$ as the parameters and the vector $U$ as the unobserved data in the model (8). With a Bayesian framework each EM pass $p$ is defined by

$$
\Delta_{p+1} = \arg\max_\Delta E (\log p(Y, U, \Delta) | Y, \Delta_p)
$$

where the joint density can be divided into three factors

$$
p(Y, U, \Delta) = p(Y | U, \Delta) p(U | \Delta) p(\Delta).
$$
The first and second factors are given by (8) and (7) above,

\[ p(Y \mid U, \Delta) = N(Y; B_\Delta A_\Delta U, R_\Delta) \]
\[ p(U \mid \Delta) = N(U; \begin{bmatrix} \hat{\theta}_0 \\ \theta \end{bmatrix}, Q_\Delta), \]

the last factor is the Bayesian prior for the segmentation sequence and can be ignored if the ML framework is used.

Denoting the quadratic norm \( x^T A x \) by \( \|x\|_A^2 \), and the determinant by \( |A| \), the logarithm of (10) is

\[
\log p(Y, U, \Delta) = \log p(\Delta) - \frac{1}{2}\|Y - B_\Delta A_\Delta U\|_{R_\Delta}^2 - \frac{1}{2} \log |R_\Delta| -
\frac{1}{2}\|U - \begin{bmatrix} \hat{\theta}_0 \\ \theta \end{bmatrix}\|_{Q_\Delta}^2 - \frac{1}{2} \log |Q_\Delta| - \frac{(N_\Delta - 1)n + Nn_\theta + n_\Delta}{2} \log(2\pi). \tag{11}
\]

Since the maximization in (9) is performed over the segmentation sequence \( \Delta \) only terms involving the segmentation parameters need to be retained from (11) when the conditional expectation is evaluated. Hence, removing terms and positive factors independent of the segmentation sequence we define the return function

\[
J(Y, U, \Delta) = 2 \log p(\Delta) - \|Y - B_\Delta A_\Delta U\|_{R_\Delta}^2 - \log |R_\Delta| -
\|U - \begin{bmatrix} \hat{\theta}_0 \\ \theta \end{bmatrix}\|_{Q_\Delta}^2 - \log |Q_\Delta| \tag{12}
\]

and use

\[
Q(\Delta, \Delta_p) = E_U \left( J(Y, U, \Delta) \mid Y, \Delta_p \right)
\]

in the EM algorithm:

\[
\Delta_{p+1} = \arg \max_\Delta Q(\Delta, \Delta_p)
\]

The conditional mean and covariance of the vector \( U \)

\[
\bar{U}_p = E(U \mid Y, \Delta_p)
\]
\[
S_p = E \left((U - \bar{U}_p)(U - \bar{U}_p)^T \mid Y, \Delta_p\right) \tag{13}
\]

are obtained from standard linear estimation theory using the model (8) and inserting the segmentation sequence \( \Delta_p \). In the actual implementation used in the simulation evaluation of Section 5 we utilize the square root array algorithms of Morf and Kailath [7, 1, 6]. Utilizing the sparse matrix format in Matlab this implementation yields maximum numerical stability with a minimum of coding complexity.

Taking the conditional expectation of (12), inserting (13) and completing the squares yields

\[
Q(\Delta, \Delta_p) = 2 \log p(\Delta) - \|Y - B_\Delta A_\Delta \bar{U}_p\|_{R_\Delta}^2 - \log |R_\Delta| - \log |Q_\Delta| -
\|\bar{U}_p - \begin{bmatrix} \hat{\theta}_0 \\ \theta \end{bmatrix}\|_{Q_\Delta}^2 - \text{tr} \left( (A_\Delta^T B_\Delta^T R_\Delta^{-1} B_\Delta A_\Delta + Q_\Delta^{-1}) S_p \right) \tag{14}
\]
where the linearity of the trace operator $\text{tr}(\cdot)$ and the relation $\|x\|_A^2 = \text{tr}(Ax x^T)$ has been used repeatedly. Since $\Delta$ is discrete, the maximization step in the EM algorithm follows by evaluating all combinations of $\Delta$ and choosing the one yielding the highest return value of (14). In the case of a binary segmentation sequence this means that $2^N$ values of $Q(\Delta, \Delta_p)$ must be evaluated, a more detailed inspection of the block matrices entering (14) shows that this exponential growth can be replaced by a linear one if the prior distribution $p(\Delta)$ is chosen in the right way. We will show this shortly but first we summarize the EMS algorithm in the following steps.

1. (Initialize, $p = 1$)
   Assume no jumps by setting $\Delta_1 = 0$.

2. (Expectation)
   Compute the estimate $\hat{U}_p$ and the error covariance $S_p$ based on the segmentation sequence $\Delta_p$.

3. (Maximization)
   Compute the next segmentation sequence $\Delta_{p+1}$ as the maximizing argument of (14) by choosing the alternative with highest return.

4. Set $p := p + 1$ and return to item 2.

The iterations are halted when no significant improvement of (14) is obtained in two consecutive passes through the algorithm.

If $N$ is very large the stacked vectors and block matrices in (14) enforce requirements of large memory in the implementation. By more closely investigating the block matrices we will see that indeed there is no need to form these large matrices. Instead, standard fixed interval Kalman smoothing can be used to compute the smoothed state estimate and covariance and the return (14) can be expressed in these quantities. First we define a notation for the conditional mean and covariance of the complete state sequence

$$
\hat{X}_p = \mathbb{E}(X \mid Y, \Delta_p)
$$

$$
P_p = \mathbb{E}\left((X - \hat{X}_p)(X - \hat{X}_p)^T \mid Y, \Delta_p\right),
$$

from (6) we have that

$$
X = A_\Delta U \quad U = A_\Delta^T X
$$

where $M^\dagger = (M^T M)^{-1} M^T$ is the Moore-Penrose pseudo inverse. Inserting this in the return (12) yields

$$
J(Y, X, \Delta) = 2 \log p(\Delta) - \|Y - B_\Delta \hat{X}_p\|_{R_\Delta^{-1}}^2 - \log |R_\Delta| - \|A_\Delta^\dagger X - \left[\begin{array}{c} \hat{x} \\ \hat{v} \end{array}\right]\|_{Q_\Delta^{-1}}^2 - \log |Q_\Delta|
$$

taking conditional expectation, inserting (15) and completing the squares we have that

$$
Q(\Delta, \Delta_p) = 2 \log p(\Delta) - \|Y - B_\Delta \hat{X}_p\|_{R_\Delta^{-1}}^2 - \log |R_\Delta| - \log |Q_\Delta| - \|A_\Delta^\dagger \hat{X}_p - \left[\begin{array}{c} \hat{x} \\ \hat{v} \end{array}\right]\|_{Q_\Delta^{-1}}^2 - \text{tr}\left((B_\Delta^T R_\Delta^{-1} B_\Delta + A_\Delta^T Q_\Delta^{-1} A_\Delta) P_p\right).
$$

(16)
It is straightforward to verify that the pseudo inverse of $A_{\Delta}$ has a strong diagonal structure,

$$
A_{\Delta}^{-1} = \begin{bmatrix}
I & 0 & \cdots & 0 \\
0 & G_{1,\delta}^\dagger F_{1,\delta} & \cdots & 0 \\
0 & 0 & \cdots & G_{2,\delta}^\dagger \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & G_{N-1,\delta}^\dagger F_{N-1,\delta} & G_{N,\delta}^\dagger
\end{bmatrix}
$$

Furthermore, $B_{\Delta}, R_{\Delta}^{-1}$ and $Q_{\Delta}^{-1}$ are all block diagonal which means that each norm and matrix trace in (16) can be written as a sum of norms and matrix traces of smaller matrices. Introducing a notation for the smoothed state estimate and (cross) covariance

$$
\tilde{x}_k^P = \mathbb{E}(x_k | Y, \Delta_p) \\
P_{t,k}^P = \mathbb{E}((x_t - \tilde{x}_t^P)(x_k - \tilde{x}_k^P)^T | Y, \Delta_p)
$$

it follows that the return function (16) can be written

$$
Q(\Delta, \Delta_p) = \log p(\Delta) - \sum_{k=1}^{N} \left( \|y_k - H_{k,\delta} \tilde{x}_k^P \|_{R_{k,\delta}^{-1}}^2 + \text{tr} \left( H_{k,\delta}^T R_{k,\delta}^{-1} H_{k,\delta} P_{t,k}^P \right) + \right.
+ \log |R_{k,\delta}| - \sum_{k=1}^{N-1} \left( \|G_{k,\delta}^\dagger (\tilde{x}_{k+1}^P - F_{k,\delta} \tilde{x}_k^P) \|_{Q_{k,\delta}^{-1}}^2 + \log |Q_{k,\delta}| + \right.
\text{tr} \left( G_{k,\delta}^\dagger Q_{k,\delta}^{-1} G_{k,\delta}^\dagger (P_{k+1,k+1}^P - 2F_{k,\delta} P_{k,\delta}^P F_{k,\delta}^T) \right) \right)
$$

(17)

where once again the terms independent of $\Delta$ have been removed from the expression. If the Bernoulli prior (4) is used the first term is

$$
\log p(\Delta) = \sum_{k=1}^{N} \delta_k \log \left( \frac{1-\epsilon}{\epsilon} \right).
$$

Note that by this choice of prior the conditional expectation (17) can be maximized independently for each $k$. Hence, given the state estimate and covariance (15) the return (17) is computed for each term in the sum and the option with highest return is chosen yielding a computational complexity of $O(N)$. To summarize, the EMS algorithm using standard fixed interval Kalman smoothing and a Bernoulli prior (4) for the segmentation sequence is given in the steps below.

1. (Initialize, $p = 1$)
   Assume no jumps by setting $\Delta_1 = 0$.

2. (Expectation)
   Compute the estimates $\tilde{x}_k^P$ and the error covariances $P_{t,k}^P$ using fixed interval Kalman smoothing based on the segmentation sequence $\Delta_p$. 

8
3. (Maximization)
Compute the next segmentation sequence $\Delta_{p+1}$ as the maximizing argument of (17) by choosing the alternative with highest return for each $k$.

4. Set $p := p + 1$ and return to item 2.
The iterations are halted when no significant improvement of (17) is obtained.

If the prior (5) is used instead a different kind of maximization step must be performed and the term entering (17) is

$$ \log p(\Delta) = \log p_n(m) = m \log(\lambda) - \log(m!) - \lambda $$

where $m = \sum_{k=1}^{N} \delta_k$. Since only the number of jump instants is penalized by this prior the maximization of $Q$ is performed by sequentially introducing $\delta_k = 1$ at the most rewarding position in the batch and comparing this with the cost of increasing $m$. Formally, the maximization step above is replaced by the following sequential scheme.

3. (Maximization)
For each $k$ compute the increase in $Q(\Delta, \Delta_p)$ obtained with $\delta_k = 1$ compared with $\delta_k = 0$. Sort these relative rewards in decreasing order. Set $\Delta_{p+1} = 0$.

3.1 Set $m = 0$

3.2 If $2 \log p_n(m) - 2 \log p_n(m + 1)$ is smaller than the increase in $Q$ obtained from introducing a jump $\delta_k$ at the most rewarding position add this jump to $\Delta_{p+1}$ otherwise continue at item 4.

3.3 Set $m := m + 1$ and repeat at item 3.2

Still, the complexity of the maximization is only $O(N)$ which should be compared with the general case of $O(2^N)$ if a binary segmentation sequence is used.

5 Simulation Evaluations
We evaluate the algorithm on to examples.

5.1 Scalar process with jumping average
The first example is a scalar random walk process with two jumps. The model used in the EMS filter had the Bernoulli prior and a fudge factor of 9,

$$ F_{k,\delta} = 1 \quad G_{k,\delta} = 1 \quad Q_{k,\delta} = (1 - \delta_0)0.09 + \delta_0 0.81 \\
H_{k,\delta} = 1 \quad R_{k,\delta} = 0.25 \quad \hat{x}_1 = 1 \quad P_1 = 1 \quad q = 0.933. $$

A sequence of $N = 30$ samples from this model with jumps of magnitude 3 introduced at two instants are shown in Figure 1. Running the EMS algorithm on this data yields convergence in only two iterations. The return (17) is shown in Figure 2 where the contribution at each different time instants is shown separately. The algorithm manage to find both jump instants correctly in this case. The algorithm seem rather sensitive to the noise levels since by increasing the measurement noise by a factor 2 the jumps are undetected. However, with the higher noise level it is hard to manually distinguish the jump instants just by looking at the measurement data.
Figure 1: Scalar process with changing mean value.

Figure 2: First and second iteration of the EMS algorithm.

5.2 Manoeuvre detection in target tracking

A simulation example with a five state nonlinear model of an aircraft making four abrupt changes in turn rate is depicted in Figure 3(a). The aircraft starts to the right in the figure and flies towards the left. The position measurements are generated using a sampling period of 5 seconds and independent additive Gaussian noise with standard deviation 300 m in each channel. The target is turning during sample 15–18 and sample 35–38. Figure 3(b) presents the result of applying the EMS algorithm to the batch of data depicted in Figure 3(a).

In the filter, a four dimensional linear model of the aircraft movement was used together with the manoeuvring model (3). Since the simulation model is nonlinear the distinct jumps in the fifth turn rate state are hard to detect using the linear model. There is a trade-off between detection sensitivity and non-manoeuvre performance since the first turn is less distinct with smaller turn rate than the second. The trade-off is controlled by choosing the three filter parameters process noise covariance $Q_k$, the fudge factor $\gamma$, and the probability of not manoeuvring $q$. After some fine tuning of the filter parameters the filter
detected manoeuvres at sample 15 and during samples 32–40. This simulation case is also presented in [3].

6 Conclusions and Extensions

The EM algorithm is a general method for ML or MAP estimation. Applied to the segmentation problem it yields the EMS algorithm which consists of alternating between fixed interval Kalman smoothing and discrete optimization. The experience gained during the development of the short simulation study presented in Section 5 is that the algorithm converges in only a few iterations. This could of course be a positive fact since it means low computational burden. However, it might also indicate that the posterior density has a lot of local minima and the risk to end up in one of those is rather high.

Several extensions of the work presented herein are possible. One already mentioned is to consider general $r$-valued segmentations. Another is to apply the algorithm to recursive problems, using, e.g., fixed lag smoothing instead of fixed interval smoothing. In order to model the behavior that mode jumps seldom come close to each other a dynamic prior for the segmentation sequence can be used. An initial probability

$$
\delta_1 = \begin{cases} 
0 & \text{with probability } p_0 \\
1 & \text{with probability } 1 - p_0 
\end{cases}
$$

and a Markovian dependency of the segmentation parameters

$$
\delta_k = \begin{cases} 
\delta_{k-1} & \text{with probability } p \\
1 - \delta_{k-1} & \text{with probability } 1 - p 
\end{cases} \quad k = 1, 2, \ldots, N
$$

leads to dynamical programming in the maximization step of the EM algorithm. This dynamical programming problem and can be solved using, e.g., the Viterbi
algorithm. The simple simulation study shows but one actual application of the algorithm, several others are imaginable and could be tested. As mentioned above it is always important to choose a good initial parameter value $\Delta_1$ close to the global maximum of the posterior density. A thorough investigation of the sensitivity to wrong initialization of the EMS algorithm should also be performed.

References


