Learning the Linear Dynamical System with ASOS

James Martens JMARTENS@CS.TORONTO.EDU
University of Toronto, Ontario, M5S 1A1, Canada

Abstract

We develop a new algorithm, based on EM, for learning the Linear Dynamical System model. Called the method of Approximated Second-Order Statistics (ASOS) our approach achieves dramatically superior computational performance over standard EM through its use of approximations, which we justify with both intuitive explanations and rigorous convergence results. In particular, after an inexpensive pre-computation phase, the iterations of ASOS can be performed in time *independent* of the length of the training dataset.

1. Introduction

1.1. The LDS Model

The time-invariant discrete Linear Dynamical System (LDS) is a classical and widely used model of real-valued multivariate time-series data $\{y_t \in \mathbb{R}^{N_y}\}_{t=1}^T$. Hidden states $\{x_t \in \mathbb{R}^{N_x}\}_{t=1}^T$ are generated via the time-evolution matrix $A \in \mathbb{R}^{N_x \times N_x}$ as:

$$x_{t+1} = Ax_t + \epsilon_t \tag{1}$$

where $\{\epsilon_t\}_{t=1}^T$ are i.i.d. multivariate normal with mean 0 and covariance matrix Q. Observations y_t are generated from x_t via the matrix $C \in \mathbb{R}^{N_y \times N_x}$ according to:

$$y_t = Cx_t + \delta_t \tag{2}$$

where $\{\delta_t\}_{t=1}^T$ are also i.i.d. multivariate normal with mean 0 and covariance matrix R. The initial state (x_1) distribution is multivariate normal with mean π_1 and covariance matrix Π_1.

The LDS is arguably the most commonly used time-series model for real-world engineering and financial applications. This is due to its relative simplicity, its mathematically predictable behavior, the existence of many physical systems that are known to be accurately modeled by it, and the fact that exact inference and prediction within the model can be done efficiently.

1.2. Learning the LDS

Learning the parameters of the LDS, sometimes called "system identification", is a well-studied problem. The available algorithms fall into three broad categories: the Prediction Error Method (PEM), Subspace Identification (4SID) and Expectation Maximization (EM). In the PEM approach (e.g. Ljung, 2002), a 1-step prediction-error objective is minimized via gradient-based optimization methods. Typical implementations use either gradient descent and thus require many iterations to converge, or use 2^{nd}-order optimization methods but then become impractical for large models.

EM, a broadly applied algorithm for maximum likelihood parameter estimation for hidden-variable models, can be applied to the LDS. The maximum likelihood objective can be seen as the special case of the prediction error objective (Ljung, 2002), but EM takes a different approach to PEM in optimizing this objective making it more practical for large models. Both PEM and EM are iterative optimization algorithms where each iteration requires a pass over the entire dataset. Since very many iterations can be required, neither of these algorithms scale well to very long time-series.

In the 4SID approach (Overschee & Moor, 1991), the LDS equations are re-written as large block matrix formulae, which are used to produce an estimate of the hidden states sequence via matrix projections (this boils down to computing a large singular value decomposition), which is then used to estimate the parameters. The block formulae generate predictions for future data-points using only the i previous ones, where i is a meta-parameter which controls the quality of the solution at the cost of computational performance. By contrast, statistically optimal state estimators (such as those used in the E-step of EM) use the entire time-series, including both past and future observations. 4SID is not an iterative optimization algorithm like PEM or EM and thus often tends to be faster than these methods while avoiding the problem of bad local minima. However, the solutions it produces, while of high quality, tend not to be locally optimal for any particular objective function

Appearing in *Proceedings of the 27^{th} International Conference on Machine Learning*, Haifa, Israel, 2010.

(such as the log-likelihood). The approach thus often advocated is to run EM initialized with the solution produced by 4SID, thus avoiding bad local minima while also achieving statistical optimality (Smith & Robinson, 2000). One disadvantage of 4SID is that standard implementations of it have considerable space requirements that prevent them from scaling nicely with the length of the training time-series.

1.3. Our Contribution

This paper develops a new method for dramatically increasing the efficiency of EM via an approximation scheme which we call the method of Approximate Second-Order Statistics (ASOS). The ASOS scheme approximates the E-step so that it can be computed in time *independent of* T, the length of the training time-series. This allows EM to be practical for time-series of nearly unbounded size, making it much more useful as an optimization tool to be used in conjunction with 4SID, or even as a replacement in cases where excessively long time-series make 4SID infeasible. Since the 4SID and EM algorithms have been analyzed and compared (Smith & Robinson, 2000; Smith et al., 1999), our goal in this paper will instead be to compare "ASOS-EM" with standard EM and show that the approximations have only a minimal effect on the solution quality while providing a huge computational performance gain, both in an theoretical and practical sense.

2. EM for LDS learning

The objective function maximized during maximum likelihood learning is the log probability of the observation sequence y given the model parameters (also known as the log likelihood function). This can be obtained from the previous joint probability by integrating out the hidden states: $\log p(y|\theta) = \log \int_x p(x, y|\theta)$. While the gradient and Hessian of this function are difficult to compute (indeed, implementations of PEM often resort to finite differences), it is relatively simple to compute the log joint probability and its expectation given y for a particular setting of the parameters. The EM algorithm, which was first applied to LDS parameter learning by Shumway & Stoffer (1982), can indirectly optimize the log-likelihood by iteratively maximizing this later quantity, which we denote $\mathcal{Q}_n(\theta)$.

$$\mathcal{Q}_n(\theta) \equiv E_{\theta_n}[\log p(x, y|\theta)|y] = \int p(x|y, \theta_n) \log p(x, y|\theta)\, dx$$

The EM algorithm iteratively alternates between two phases called the "E-step" and the "M-step". For a given estimate of the parameters θ_n, the E-step computes the expectations which appear in the expression for $\mathcal{Q}_n(\theta)$, allowing it to be easily evaluated and optimized with respect to θ in the M-step. The M-step then computes the new parameter estimate as $\theta_{n+1} = \arg\max_\theta \mathcal{Q}_n(\theta)$.

2.1. M-step for the LDS

As a consequence of the linear and Gaussian nature of the LDS model, the function $\mathcal{Q}_n(\theta)$ can be written in terms of θ and statistics that are first and second order in x and y (and are highly non-linear in θ_n). With these statistics computed, optimizing $\mathcal{Q}_n(\theta)$ with respect to θ reduces to a straightforward application of matrix calculus and is similar to linear regression with an unknown covariance. For a full specification and derivation of the M-step see Ghahramani & Hinton (1996).

For the sake of brevity we will use the following standard notation for the remainder of this paper:

$$x_t^k \equiv E_{\theta_n}[\, x_t \mid y_{\le k}\,] \qquad V_{t,s}^k \equiv \mathrm{Cov}_{\theta_n}[\, x_t, x_s \mid y_{\le k}\,]$$
$$\tilde{y}_t \equiv y_t - E_{\theta_n}[\, y_t \mid y_{\le t-1}\,] \qquad S_t \equiv \mathrm{Cov}_{\theta_n}[\, \tilde{y}_t \mid y_{\le t}\,]$$

Noting that $E_{\theta_n}[x_t x_s' \mid y_{\le k}\,] = x_t^k x_s^k + V_{t,s}^k$, and using the notation $(a, b)_k \equiv \sum_{t=1}^{T-k} a_{t+k} b_t'$ (where v' denotes the transpose of v) the complete list of statistics required to compute the M-step may be written as:

$$(y, x^T)_0, \quad (x^T, x^T)_0 + \sum_{t=1}^{T} V_{t,t}^T, \quad (x^T, x^T)_1 + \sum_{t=1}^{T-1} V_{t+1,t}^T$$

These expressions contain both sums of covariances and sums of products of means. For the remainder of this report we will refer to the later sums as the "M-statistics", which are particular examples of "2nd-order statistics".

2.2. E-Step for the LDS (Kalman recursions)

The Kalman recursions (see Ghahramani & Hinton, 1996) are a set of recursive relations that define a computational procedure for exact inference of the distribution over the hidden states x in the LDS model. The standard approach for computing the M-statistics is to apply this procedure to find x_t^T for each value of t in succession and then perform the required sums. Doing this has time complexity $O(N_x^3 T)$ which is the reason that the EM algorithm scales poorly to long time-series.

3. The ASOS Approach

3.1. Overview

The key observation that motivates the ASOS approach is that, for the purposes of the M-step, we don't actually care about the individual moments for each x_t, but rather certain sums over the products of these, the M-statistics, along with sums over covariance matrices. Thus we can focus on the M-statistics directly and come up with approximations that will make them more efficiently computable

without having to concern ourselves with computing the individual x_t^T. To this end we will derive a set of "2nd-order recursions" from the Kalman recursions which define a recursive scheme to compute various 2nd-order statistics over y_t, x_t^t, and x_t^T, culminating in the computation of the M-statistics. Whereas the Kalman recursions relate 1st-order moments across different time-steps, these 2nd-order recursions will relate 2nd-order statistics across different "time-lags", where by "time-lag" we mean the value of k in $(a,b)_k \equiv \sum_{t=1}^{T-k} a_{t+k}b_t$, i.e. the difference in the temporal-indices as they appear in the sum.

The number of distinct time-lags is equal to T, the numbers of time-steps (statistics with time-lag larger than T are equal to the 0 matrix), and so solving them exactly would entail just as much or more computation than simply running the original Kalman recursions. Fortunately it turns out that, at the cost of introducing some fairly liberal approximations which have some favorable statistical and asymptotic properties, we can evaluate the 2nd-order recursions *much* more efficiently than the Kalman recursions. In particular, by approximating statistics of "large" time-lag by carefully chosen unbiased estimators we can derive a compact system of linear equations that can be solved very efficiently. The size of this system, and thus the cost of solving it, turns out to be a function of the cut-off time-lag "k_{lim}" at which we choose to approximate the 2nd-order statistics. The resulting algorithm only depends on the values of $(y,y)_k \equiv \sum_{t=1}^{T-k} y_{t+k}y_t'$ for $0 \leq k \leq k_{kim}$. And while computing these clearly requires time proportional to T, they only need to be pre-computed *once* before the EM-iterations begin.

To realize this approach we need to simplify the Kalman recursions by using a tool from LDS theory known as "steady-state", which we discuss next.

3.2. The steady state assumption

The Kalman recursions, in addition to computing the conditional means x_t^T for each state, also compute the covariance matrices (e.g. $V_{t,t}^T$) between hidden state vectors, along with the filtering and smoothing matrices, K_t and J_t (for the precise definitions of these, we defer again to Ghahramani & Hinton (1996)). Notably, the recursions for these quantities do *not* involve the actual time-series data. Moreover, a well-known result is that under certain control-theoretic conditions for the model parameters these matrices rapidly approach constant matrices as t grows, a phenomenon known as "steady state" (e.g. Goodwin & Sin, 1984).

In particular, as t grows K_t converges to a constant matrix which we denote K (without a subscript). And similarly, $V_{t,t}^T$, $V_{t,t-1}^T$ and J_t converge to Λ_0, Λ_1 and J respectively as $\min(t, T-t)$ grows. Computing these matrices reduces to solving discrete algebraic Riccati equations and simpler Lyapunov equations, for which there are efficient algorithms.

For simplicity we will assume that the steady-state condition applies over the entire time-series. Later we will see how this strong assumption can be replaced with a more realistic one that will approximate the truth to an arbitrary precision. Under steady state the Kalman recursions for the state means can be written compactly as:

$$x_t^* = Hx_{t-1}^* + Ky_t \qquad\qquad x_t^T = Jx_{t+1}^T + Px_t^*$$

where we have defined $x_t^* \equiv x_t^t$, $H \equiv A - KCA$ and $P \equiv I - JA$. The usefulness of the switch in notation from x_t^t to x_t^* is that it allows us to use our special notation for 2nd-order statistics $(a,b)_k \equiv \sum_{t=1}^{T-k} a_{t+k}b_t'$ with the vector-list x_t^t as an argument, e.g. $a = x^*$.

In addition to simplifying the Kalman recursions, assuming steady state makes it much easier to compute the covariance-matrix sums required by the M-step; we just multiply the corresponding steady-state value by T or $T-1$, as the case may be. Thus to complete the E-step it remains only to compute the M-statistics.

3.3. Recursions for the 2nd-order statistics

In this section we will give the general approach for deriving the 2nd-order recursions and then provide the complete list. To find the equation that computes the statistic $(a,b)_k$ we right-multiply the Kalman recursion for a_{t+k} (or if a = y, just the trivial equation $y_{t+k} = y_{t+k}$) by the transpose of the one for b_t and then sum both sides from $t = 1$ to $T-k$. As a simple example, suppose we wish to find the recursion for $(x^*, y)_k$. We simply right-multiply the simplified Kalman recursion for x_{t+k}^* by y_t', sum both sides over t, and then re-write everything using our special notation for 2nd-order statistics:

$$\begin{aligned}\sum_{t=1}^{T-k} x_{t+k}^* y_t' &= \sum_{t=1}^{T-k} (Hx_{t+k-1}^* y_t' + Ky_{t+k}y_t') \\ &= H\sum_{t=1}^{T-k} x_{t+k-1}^* y_t' + K\sum_{t=1}^{T-k} y_{t+k}y_t' \\ &= (x^*, y)_k = H((x^*, y)_{k-1} - x_T^* y_{T-k+1}') + K(y,y)_k\end{aligned}$$

Complicating this idea somewhat is the fact that the Kalman recursions for x_t^t are not defined for the special cases x_t^t for $t = 1$ and thus we must add in an additional nuisance term $a_{1+k}b_1'$ to compensate. Similarly, we must sometimes subtract an additional term from a statistic before using it in the equation for a statistic of a higher time lag since the latter is summed over a smaller range ($1...T-k$ instead of $1...T-k+1$). The complete list

of 2nd-order recursions, which we will call the "2nd-order recursions", is:

$$(y, x^*)_k = (y, x^*)_{k+1} H' + ((y, y)_k - y_{1+k} y_1') K' + y_{1+k} x_1^{*\prime}$$
$$(x^*, y)_k = H((x^*, y)_{k-1} - x_T^* y_{T-k+1}') + K(y, y)_k$$
$$(x^*, x^*)_k = (x^*, x^*)_{k+1} H' + ((x^*, y)_k - x_{1+k}^* y_1') K' + x_{1+k}^* x_1^{*\prime}$$
$$(x^*, x^*)_k = H((x^*, x^*)_{k-1} - x_T^* x_{T-k+1}^{*\prime}) + K(y, x^*)_k$$
$$(x^T, y)_k = J(x^T, y)_{k+1} + P((x^*, y)_k - x_T^* y_{T-k}') + x_T^T y_{T-k}'$$
$$(x^T, x^*)_k = J(x^T, x^*)_{k+1} + P((x^*, x^*)_k - x_T^* x_{T-k}^{*\prime}) + x_T^T x_{T-k}^{*\prime}$$
$$(x^T, x^T)_k = ((x^T, x^T)_{k-1} - x_k^T x_1^T) J' + (x^T, x^*)_k P'$$
$$(x^T, x^T)_k = J(x^T, x^T)_{k+1} + P((x^*, x^T)_k - x_T^* x_{T-k}^{T\prime}) + x_T^T x_{T-k}^T$$

3.4. The ASOS Approximations

Examining the 2nd-order recursion for $(y, x^*)_k$ we see that it depends on $(y, x^*)_{k+1}$. If we had access to its exact value of $(y, x^*)_{k+1}$ we could use the recursion to compute $(y, x^*)_k$ exactly. But since we don't we will have to rely on an approximation. In particular we will approximate $(y, x^*)_k$ for some sufficiently large value of k, which we will denote k_{lim}, and then use the recursion to recursively compute approximate versions of each $(y, x^*)_k$ from $k = k_{lim}$ down to $k = 0$.

There are several reasons why we might expect this could be a reasonable thing to do. Firstly, for large time-lags these statistics express relationships between variables that are far apart in time in the LDS model and thus likely less important than relationships between variables that are close. Later we will show how this intuition can be made formal by quantifying the approximation error and identifying sufficient conditions under which it is negligible. Another reason that this approximation is appealing is that it is reminiscent (although not equivalent) of one of the approximations implicitly made by the 4SID algorithm, namely that state vectors at each time-step are estimated via a non-steady state Kalman filter starting i time-steps in the past and initialized from 0, where i is 4SID's "block-size" meta-parameter. Finally, by using estimators that are unbiased under the model we expect that the quality of the approximation will become better as the model parameters converge to a setting that fits the data and/or the amount of data increases. In a later section we will give a formal result which quantifies the relative error of the approximations and establishes that it goes to zero as the amount of data grows, under the condition that the data is generated from the model.

The approximation we will use for $(y, x^*)_{k_{lim}+1}$ is $CA\left((x^*, x^*)_{k_{lim}} - x_T^* x_{T-k_{lim}}^{*\prime}\right)$. This seemingly arbitrary choice is justified by the following result:

Claim 1. *If the data is generated from the model's distribution then this approximation is unbiased.*

Proof. For any $k > 1$ we have:

$$\begin{aligned}\mathrm{E}_{\theta_n}[\, y_{t+1} x_{t-k}^{*\prime} \mid y_{\le t}\,] &= \mathrm{E}_{\theta_n}[\, y_{t+1} \mid y_{\le t}\,] x_{t-k}^{*\prime} \\ &= C x_{t+1}^t x_{t-k}^{*\prime} = CA x_t^* x_{t-k}^{*\prime}\end{aligned}$$

Then taking the expectation of both sides and using the law of iterated expectations we get:

$$\mathrm{E}_{\theta_n}[\, y_{t+1} x_{t-k}^{*\prime}\,] = \mathrm{E}_{\theta_n}[\, CA x_t^* x_{t-k}^{*\prime}\,]$$

Taking $k = k_{lim}$ and summing both sides from $t = 0$ to $t = T - k_{lim}$ we have:

$$\mathrm{E}_{\theta_n}[\, (y, x^*)_{k_{lim}+1}\,] = \mathrm{E}_{\theta_n}[\, CA\left((x^*, x^*)_{k_{lim}} - x_T^* x_{T-k_{lim}}^{*\prime}\right)\,]$$

which is the claim. □

In order to completely evaluate the 2nd-order recursions we will also need similar approximations to start the recursions for $(x^*, x^*)_k$, $(x^T, y)_k$, $(x^T, x^*)_k$ and $(x^T, x^T)_k$ (note that the recursion for $(x^*, y)_k$ can be started from $(x^*, y)_0 = (y, x^*)_0'$).

The following two approximations can be shown to be unbiased using a proof similar to the one given above:

$$(x^T, x^*)_{k_{lim}} \approx (x^*, x^*)_{k_{lim}} \qquad (x^T, y)_{k_{lim}} \approx (x^*, y)_{k_{lim}}$$

Together with the approximation for $(y, x^*)_{k_{lim}+1}$ we will call these the "ASOS approximations".

Unfortunately there are no obvious candidates for unbiased approximations of either $(x^*, x^*)_{k_{lim}}$ or $(x^T, x^T)_{k_{lim}}$ that could be used to start the corresponding 2nd-order recursions. In the next section we will show how this problem can be circumvented by deriving two additional equations from the Kalman recursions that will sufficiently constrain the solution.

Finally, we need to approximate the "first-order statistics" x_t^* and x_t^T for the first and last $k_{lim} + 1$ time-steps since these appear as "nuisance terms" in the 2nd-order equations. This can be done easily by running the steady-state Kalman recursions on the first and last "k_{lag}" time-steps, where k_{lag} is some constant $\ge k_{lim} + 1$. For the first k_{lag} time-steps the Kalman recursions can be initialized from π_1. In our experiments we used $k_{lag} = 2k_{lim}$.

3.5. Solving the approximated system

Solving the 2nd-order recursions subject to the ASOS approximations is a non-trivial task. One key difficulty is that we have no way of starting either the recursions for $(x^*, x^*)_k$ and $(y, x^*)_k$ without first obtaining some kind of approximation for $(x^*, x^*)_{k_{lim}}$. In this section we will show how this difficulty can be overcome, and derive a complete method for solving the 2nd-order recursions subject to the ASOS approximations. We will assume that the 1st-order nuisance terms have already been approximated.

To overcome the aforementioned difficulty, we can use the fact that there are two 2nd-order recursions for $(x^*, x^*)_k$, one which "increments" k and one which "decrements" so as to derive a new equation that keep k constant, thus relating $(x^*, x^*)_k$ to *itself*. In particular we can plug in the fourth 2nd-order recursion for $(x^*, x^*)_{k+1}$ into the third recursion and simplify:

$$(x^*, x^*)_k = H(x^*, x^*)_k H' + ((x^*, y)_k - x^*_{1+k} y_1')K' - H x^*_T x^*_{T-k}{}' H' + K(y, x^*)_{k+1} H' + x^*_{1+k} x_1^{*\prime}$$

Then using the same basic method we can derive a similar equation for $(x^T, x^T)_k$:

$$(x^T, x^T)_k = J(x^T, x^T)_k J' + P((x^*, x^T)_k - x^*_T x^T_{T-k}{}') - J x^T_{k+1} x_1^{T\prime} J' + J(x^T, x^*)_{k+1} P' + x^T_T x^T_{T-k}{}'$$

We will call these two equations the "ASOS equations".

Our basic strategy will be to exploit the self-referential nature of the first ASOS equation in order to find a solution for $(x^*, x^*)_{k_{lim}}$ (taking $k = k_{lim}$). Complicating this idea is the presence of additional unknown matrix quantities in the equation and so before we can proceed we must find a way to express these in terms of $(x^*, x^*)_{k_{lim}}$.

By repeated application of the first and second 2nd-order recursions, followed by an application of the first ASOS approximation, $(x^*, y)_{k_{lim}}$ can be expressed as:

$$(x^*, y)_{k_{lim}} = (x^*, y)_{k_{lim}}{}^\dagger + H^{2k_{lim}+1}(x^*, x^*)_{k_{lim}}{}' A' C'$$

where $(x^*, y)_{k_{lim}}{}^\dagger$ is the value of $(x^*, y)_{k_{lim}}$ as computed by solving these recursions starting from $(x^*, x^*)_{k_{lim}} = 0$. This formula can be easily verified by following the dependency on $(x^*, x^*)_{k_{lim}}$ through said recursions.

Substituting this expression for $(x^*, y)_{k_{lim}}$ and the first ASOS approximation $(y, x^*)_{k_{lim}+1}$ into the first ASOS equation, then simplifying, gives:

$$(x^*, x^*)_k = A(x^*, x^*)_k H' + H^{2k+1}(x^*, x^*)_k{}' A' C' K' + G$$
$$G \equiv -A x^*_T x^*_{T-k}{}' H' + \left((x^*, y)_k{}^\dagger - x^*_{1+k} y_1'\right) K' + x^*_{1+k} x_1^{*\prime}$$

where $k = k_{lim}$ for brevity.

Since we can compute $(x^*, y)_{k_{lim}}{}^\dagger$ by just running the recursions under $(x^*, x^*)_{k_{lim}} = 0$, the only unknown quantity in this equation, which we will call the "primary equation", is $(x^*, x^*)_{k_{lim}}$. Moreover this equation is linear in $(x^*, x^*)_{k_{lim}}$ which gives us some hope that we can solve it. Unfortunately, it is not clear at first glance how we can do this efficiently. This equation almost has the form of a Sylvester equation, for which there are well-known efficient algorithms (Bartels & Stewart, 1972), but is slightly more complicated due to the presence of the term $H^{2k_{lim}+1}(x^*, x^*)_{k_{lim}}{}' A' C' K'$. The good news is that we have developed an iterative algorithm for solving this equation which seems to converge very quickly in practice (for details, see the supplement available at http://www.cs.toronto.edu/~jmartens/ASOS).

With the solution of the primary equation we can utilize the ASOS approximation for $(y, x^*)_{k_{lim}+1}$ and recursively compute $(y, x^*)_k$ for $k = k_{lim}$ down to 0 using the first 2nd-order recursion. Then using the fact that $(x^*, y)_0 = (y, x^*)_0'$ we may recursively compute $(x^*, y)_k$ for $k = 0$ to k_{lim} using the second 2nd-order recursions. With $(x^*, y)_k$ computed we may then use the third 2nd-order recursion to recursively compute $(x^*, x^*)_k$ for $k = k_{lim}$ down to 0. Having computed $(x^*, y)_k$ also allows us to recursively compute $(x^T, y)_k$ via the fifth 2nd-order recursion, starting the recursion with the ASOS approximation for $(x^T, y)_{k_{lim}}$.

Next, with $(x^*, x^*)_k$ computed for $k = 0$ to k_{lim} we may use the sixth 2nd-order recursion to recursively compute $(x^T, x^*)_k$, starting the recursion with the second ASOS approximation (i.e. the one for $(x^T, x^*)_{k_{lim}}$). Finally, we may compute $(x^T, x^T)_0$ by solving the second ASOS equation (which can be done efficiently since it has the form of a Lyapanov equation) and use the seventh 2nd-order recursion to compute $(x^T, x^T)_1$ from this.

3.6. Relaxing the Steady-state Assumption

To derive the 2nd-order equations in their simple form it is critical that the filtering and smoother matrices K, H and P do not vary with the time-step t. Otherwise they can't be factored out of the sums, making it impossible to write the recursions only in terms of 2nd-order statistics and nuisance terms.

We know that the LDS rapidly obtains steady-state (up to an arbitrary precision) everywhere except for some leading and trailing i time-steps, where i is not a function of T and generally $i \ll T$. Thus we can apply the ASOS method to approximate the statistics over this middle interval and use the non-steady-state Kalman recursions to (approximately) compute the statistics associated with the first and last i time-steps. i can be determined by monitoring convergence of K_t to K while running the Kalman-filter, or just set at some reasonably large fixed value.

4. Error analysis

4.1. The relationship between k_{lim} and the approximation error

In this section we will derive a set of formulae which quantify the error in the M-statistics as computed via the 2nd-order recursions in terms of the error introducted due to the

approximations. This ends up being a linear relationship relationship because the 2nd-order recursions are linear in the 2nd-order statistics. The notable feature of this relationship is that its 'strength' decays exponentially as k_{lim} grows, thus providing one justification for idea that the quality of the approximation increases with the value of k_{lim}.

Consider each of the three 2nd-order statistics approximated directly by the ASOS approximations, adding $(x^*, x^*)_{k_{lim}}$ to this list. We will call these the "Directly Approximated Statistics" or the DAS. The following result helps quantify the error in the M-statistics in terms of error in the DAS. Note that error due to any approximation in the 1st-order nuisance terms will be ignored in this analysis for the purpose of simplicity. We will briefly address this problem at the end of the section.

Claim 2. *Given a fixed setting of the parameters θ there exists some $0 \leq \lambda < 1$ such that for each M-statistic the difference between the true value and the value as approximated by the ASOS procedure can be expressed as a linear function of the approximation error in the DAS whose operator norm is bounded above by $c{k_{lim}}^2 \lambda^{k_{lim}-1}$ for some constant c that doesn't depend on k_{lim}.*

Proof. The proof is straightforward. λ turns out to be the spectral radius of H and J (they are equal). For a detailed proof see the supplement available at http://www.cs.toronto.edu/~jmartens/ASOS. □

Note that the above result does not assume anything about the particular approximations being used for the DAS. So unless the approximation error of one of the DAS grows extremely quickly with k_{lim} we can conclude that the error in the M-statistics will decay exponentially as k_{lim} grows. And since the expected size of any 2nd-order statistic can be bounded, even a naive approximation of 0 for each DAS will ensure that the associated expected error is bounded.

In practice we have found that λ can often be significantly less than 1, even when the spectral radius of A is relatively close to 1. However, as the EM algorithm progresses and the model (as determined by the evolving parameters) becomes more "confident", λ may occasionally grow large enough that $\lambda^{k_{lim}-1}$ won't be very close to 0. Fortunately, there is another result which we present in the next section that allows us to bound the error in a manner that doesn't depend on λ but is instead related to the value of T.

Having ignored the issue of approximating the the 1st-order nuisance terms in the above analysis we will now briefly address it. If these terms are approximated by applying the steady-state Kalman recursions to the leading and trailing $k_{lim}+1$ time-steps, which is the approach we advocate, and if we add x^T_{klim+1} and x^*_{T-klim} to the DAS list, then the above claim still holds and our proof can be easily extended.

4.2. Asymptotic Behavior

In this section we will characterize the asymptotic behavior of the ASOS approximations as $T \to \infty$ under the condition that the data is in fact generated from the model. While this scenario is artificial, it can nevertheless inform us about how the approximations will behave in practice. In particular, if the model is close to describing the data, and T is sufficiently large, then this characterization should at least describe the real situation approximately.

We have already established in section 3.4 that the ASOS approximations are unbiased in this setting. The first objective of this section is to establish a deeper result, that the error in the ASOS approximations converges to 0 in the expected squared $\|\cdot\|_2$-norm (viewing the matrices as vectors) as long as we scale everything by $\frac{1}{T}$. This rescaling is a natural thing to do because the 2nd-order statistics are all sums over $\sim T$ elements, and thus their expected size grows with T. Then having established this result we will outline the proof of an important consequence, namely that the M-step updates which use the ASOS-approximated M-statistics will converge in probability to the exact updates as $T \to \infty$.

Note that since we are scaling all of the equations and statistics by $\frac{1}{T}$ the effect of the nuisance terms in each equation will go to zero as $T \to \infty$ and so we can ignore them in the analysis.

Let ϕ_i be the (true) value of the error in the ith ASOS approximation, i.e. the value of the left side minus the right. So for example, $\phi_2 = (x^T, x^*)_{k_{lim}} - (x^*, x^*)_{k_{lim}}$. Then we have the following claim which characterizes the expected size of each ϕ_i:

Claim 3. *For $i = 1, 2, 3$:*

$$\lim_{T \to \infty} \mathrm{E}_\theta[\ \| \frac{1}{T} \mathrm{vec}(\phi_i) \|_2^2\ ] = 0$$

Proof. The ASOS approximations were derived by finding unbiased estimators at each time-step and then summing over time. It turns out that the approximation errors also have zero correlation across time which is the critical property required to prove this result. See the supplement for details. □

Claim 4. *The approximation error in $\frac{1}{T}$-scaled 2nd-order statistics as estimated by the ASOS procedure converges to 0 in expected squared $\|\cdot\|_2$-norm as $T \to \infty$.*

Proof. This follows from the fact that the ASOS procedure is just an efficient method solving a large linear system

Table 1. Per-iteration computational complexity

EM	SS-EM	ASOS-EM
$O(N_x^3 T)$	$O(N_x^2 T + N_x^3 i)$	$O(N_x^3 k_{lim})$

whose coefficients are not a function of T, and thus the procedure can only "amplify" the errors due to the three ASOS approximations by a constant factor. For a detailed proof, see the supplement. □

Claim 5. *The parameter updates produced by the M-step using the approximated M-statistics will converge to those produced using the true M-statistics as $T \to \infty$.*

Proof. For the covariance parameters R and Q the update formula are linear in the $\frac{1}{T}$-scaled M-statistics (for A it's actually a $\frac{1}{T-1}$ scaling, but this is equivalent in the limit) and thus converge in the expected squared $\|\cdot\|_2$-norm, which implies convergence in probability.

For the parameters A and C we cannot prove convergence in the expected squared $\|\cdot\|_2$-norm but we can still prove convergence in probability. First, note that we may replace the M-statistics in the update formula with their $\frac{1}{T}$-scaled counterparts since the scaling factor $\frac{1}{T}$ will be canceled due to the matrix inversion. Second, note that convergence in expected squared $\|\cdot\|_2$-norm of the approximate M-statistics to the true ones implies their convergence in probability. Finally, note that the exact value of the M-statistic which gets inverted is non-singular (it must be, since otherwise the update formula is undefined) and thus the formula is continuous at this point. Convergence in probability of the update formula then follows by the Continuous Mapping Theorem for random variables. □

5. Computational complexity

The per-iteration computational complexity for EM, ASOS-EM (EM approximated via ASOS), and SS-EM (EM via direct evaluation of the steady-state approximated Kalman recursions) is given in Table 1. Note that we are assuming that T is the dominant term, followed by N_x, then k_{lim}, and finally i (i is defined as in section 3.6). The key difference between the per-iteration running time of ASOS-EM and that of EM or SS-EM is that there is no dependency on T. The only T-dependent computation required for ASOS-EM is the pre-computation of $(y, y)_k$ for $0 \leq k \leq k_{lim} + 1$ which only needs to be performed once, before the EM iterations begin. The y statistics can even be computed online so that the complete time-series never even needs to be stored in memory.

6. Relationship to 1st-order approximations and 4SID

A natural question to ask is if there is some approximation for the individual mean terms (i.e. x_t^* and $x_t^T \ \forall t$), that when multiplied and summed appropriately, gives the same estimates for the 2nd-order statistics that ASOS does. If such an equivalence did exist then the approximated statistic $(x^T, x^T)_0$ would always be positive definite, which isn't true in general (although it *will* always be symmetric).

Comparisons to 4SID can be made in terms of the approximation being used. As mentioned in our previous discussion of 4SID, the state estimates it (implicitly) computes for each time-step are equivalent to the estimates which would be produced by a non-steady-state Kalman filter that starts i time-steps in the past. The estimates produced by ASOS are of a different character in the sense that when they include information from the future as well as they are derived from both the filtering and *smoothing* Kalman recursions. Note however that the ASOS approximations require the model parameters to be available (i.e. the estimate produced by the EM iteration) while the 4SID estimates do not require any pre-existing parameter estimate, which is why the algorithm is non-iterative.

7. Experimental Setup

Our experiments were designed to examine the trade-off between the solution quality and speed of ASOS-EM as a function of the meta-parameter k_{lim}, while using standard EM and SS-EM as baselines. All algorithms were implemented using carefully vectorized MATLAB code and run on an Intel 3.2GHz quad-core machine. Exact log-likelihoods were computed every 10 iterations as each algorithm ran. The runs were all initialized with the same random initial parameters. Our implementations of ASOS-EM and SS-EM both used the "relaxed" steady-state approximation with i fixed to 25.

We used 3 datasets in our experiments. The first was a 3-dimensional time-series of length 6305 which consisted of sensor readings from an industrial milk evaporator. This is a standard dataset used in system identification and is available on-line from the Database for the Identification of Systems (DaISy). The second dataset consisted of the first 10 dimensions of a 49-dimensional time-series of length 15300 consisting of transformed sensor readings from a motion capture experiment. This dataset is available on-line from the Carnegie Mellon University Motion Capture Database and was preprocessed as in Taylor et al. (2007). The third dataset was from the Signal Processing Information Base (SPIB) based at Rice University and consisted of the first 750,000 time-steps (38 seconds) of an audio recording taken in the noisy 'Operations Room" of a de-

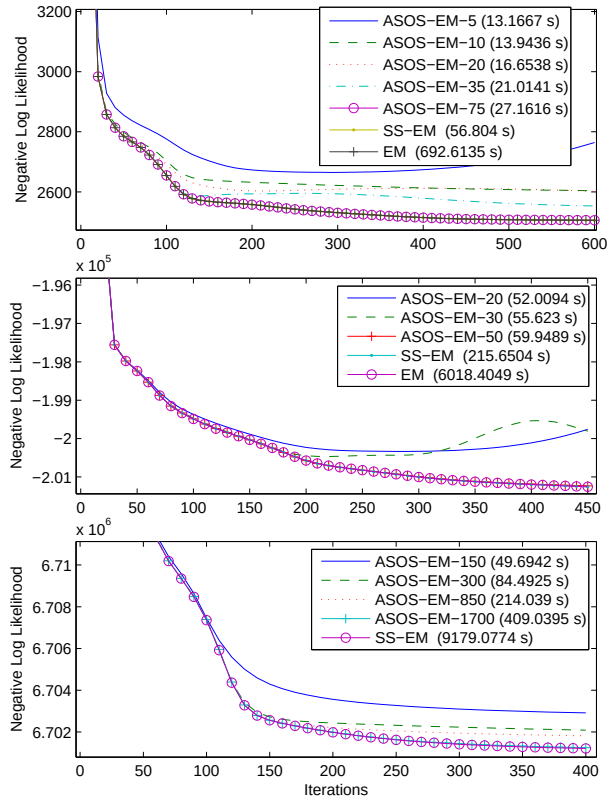

Figure 1. **NOTE:** *Running times are included in the figure legends in brackets.* **Top**: *Results from the evaporator dataset with* $N_x = 15$. **Middle**: *Results from the mo-cap data with* $N_x = 40$. **Bottom**: *Results from the destroyer operations room audio dataset with* $N_x = 20$. **NOTE:** *Graphs are highly zoomed so that the differences between the algorithms may appear more significant than they actually are.*

stroyer warship.

We will present our results as a series of graphs of log-likelihood versus iteration number, with the meta-parameters and other important details of the experiment given in the captions. 'ASOS-EM-n' is a run of ASOS-EM with $k_{lim} = n$.

8. Discussion of Results

Our experiments demonstrate that ASOS-EM converges in roughly the same number of iterations as standard EM with the solution quality approaching that achieved by standard EM as k_{lim} is raised, as predicted by the theory. Moreover, the computational performance advantages of ASOS-EM over EM and SS-EM are clearly evident in these experiments, even for values of k_{lim} where the solution quality is virtually identical to standard EM.

We included run-times with our results only to demonstrate that ASOS-EM can achieve real performance improvements in a reasonable implementation setting (carefully vectorized MATLAB code). Whether or not the reader accepts them as reasonable indicators of relative performance, the fact remains that ASOS-EM is *asymptotically* faster than either SS-EM and EM per iteration since its iteration cost is independent of T.

Where ASOS-EM seems to diverge from standard EM (when it does at all) is in the later stages of convergence. This is likely explained by the fact that, up until the end of the optimization, the parameter estimates reflect a shorter-term temporal dependency (as indicated by the value of λ), and thus the ASOS approximation is close to exact. It is also apparent from the non-monotonic log-likelihood trend observed in the results for ASOS-EM-5 in the first graph that ASOS-EM cannot guarantee, in general, a decrease in the log likelihood for each iteration.

Overall these results are very encouraging and motivate further exploration of the ASOS method and its applications. It remains to be seen if this approach can be extended to a continuous-time version of the LDS, or one that uses control signal inputs.

Acknowledgments

The author would like to thank Geoffrey Hinton and Richard Zemel for their helpful advice. This work was supported by NSERC and the University of Toronto.

References

Bartels, R. H. and Stewart, G. W. Solution of the matrix equation AX + XB = C. *Commun. ACM*, 15(9), 1972.

Ghahramani, Z. and Hinton, G.E. Parameter estimation for linear dynamical systems. Technical report, 1996.

Goodwin, G.C. and Sin, K.S. *Adaptive Filtering Prediction and Control*. Prentice-Hall, 1984.

Ljung, L. Prediction error estimation methods. *Circuits, Systems, and Signal Processing*, 2002.

Overschee, P. Van and Moor, B. De. Subspace algorithms for the stochastic identification problem. In *30th IEEE Conference on Decision and Control*, 1991.

Shumway, R.H. and Stoffer, D.S. An approach to time series smoothing and forecasting using the em algorithm. *Journal of Time Series Analysis*, 1982.

Smith, G., Freitas, J.F. De, Niranjan, M., and Robinson, T. Speech modelling using subspace and em techniques. In *NIPS*, 1999.

Smith, G.A. and Robinson, A.J. A comparison between the em and subspace identification algorithms for time-invariant linear dynamical systems. Technical report, Cambridge University, 2000.

Taylor, G.W., Hinton, G.E., and Roweis, S. Modeling human motion using binary latent variables. In *NIPS*, 2007.