PARTICLE FILTERING APPROACH TO STATE ESTIMATION IN BOOLEAN DYNAMICAL SYSTEMS

Ulisses Braga-Neto

Department of Electrical and Computer Engineering
Texas A&M University
College Station, Texas 77843
E-mail: ub@ieee.org

ABSTRACT

Exact optimal state estimation for discrete-time Boolean dynamical systems may become impractical computationally if system dimensionality is large. In this paper, we consider a particle filtering approach to address this problem. The methodology is illustrated through application to state tracking in high-dimensional Boolean network models. The results show that the particle filter can be very accurate under a moderate number of particles. The impact of resampling on performance is also investigated.

Index Terms— Boolean Dynamical Systems, Boolean Networks, Optimal State Estimation, Particle Filtering, Sequential Monte-Carlo Methods.

1. INTRODUCTION

We investigate in this paper the application of particle filtering [1–3] to the problem of optimal estimation in dynamical systems consisting of Boolean state variables governed by networks of logical gates updated and observed through noise at discrete time intervals. Such models are of great interest in Genomic Signal Processing and other application areas.

In [4,5], a signal model for Boolean dynamic systems was introduced. It was shown that this model includes as special cases the Boolean Network (BN), Boolean Network with perturbation (BNp), and Probabilistic Boolean Network (PBN) models. The optimal recursive MMSE estimator for the proposed signal model was called the Boolean Kalman Filter (BKF) and an algorithm for its computation was provided.

If full knowledge about the model is available, the BKF provides the exact optimal state estimator. However, for large numbers of state variables, the computation of the BKF becomes impractical. Here we propose a Particle Filtering algorithm that successfully addresses the high-dimensionality problem. Numerical state-tracking experiments with high-dimensional models demonstrate the accuracy of the proposed method.

This paper is organized as follows. Section 2 reviews the Boolean signal model and its MMSE estimator, the Boolean Kalman Filter. Section 3 introduces the particle filter algorithm for state estimation under this model. Section 4 presents numerical results. Finally, Section 5 provides a summary and conclusions.

2. BOOLEAN SIGNAL MODEL

Assume that the system is described by a state process \( \{ X_k; k = 0, 1, \ldots \} \), where \( X_k \in \{ 0, 1 \}^d \) is a Boolean vector of size \( d \). The state is observed indirectly through the observation process \( \{ Y_k; k = 1, 2, \ldots \} \), where \( Y_k \) is a general vector of any number of continuous or discrete measurements. The state and observation processes satisfy the signal model specified by:

\[
X_k = f(X_{k-1}) \oplus n_k \quad \text{(state model)}
\]

\[
Y_k = h(X_k, v_k) \quad \text{(observation model)}
\]

for \( k = 1, 2, \ldots \). Here, “\( \oplus \)” indicates component-wise modulo-2 addition, \( f: \{ 0, 1 \}^d \rightarrow \{ 0, 1 \}^d \) is an arbitrary network function, which expresses a logical relationship between the state vectors at consecutive time points, \( h: \{ 0, 1 \}^d \rightarrow O \) is a general function mapping the current state into the observation space \( O \), whereas \( \{ n_k, v_k; k = 1, 2, \ldots \} \) are white noise processes, with \( n_k \in \{ 0, 1 \}^d \) and \( v_k \in O \). The noise processes are “white” in the sense that the noises at distinct time points are uncorrelated random variables. It is also assumed that the noise processes are uncorrelated with each other and with the state process.

The optimal filtering problem consists of finding an estimator \( \hat{X}_k = h(Y_1, \ldots, Y_k) \) of the state \( X_k \) that optimizes a given performance criterion among all possible functions of \( Y_1, \ldots, Y_k \). The two criteria considered here are the conditional mean-square error (MSE):

\[
\text{MSE}(Y_1, \ldots, Y_k) = E[||\hat{X}_k - X_k||^2 | Y_k, \ldots, Y_1] \quad (2)
\]

and the (unconditional) mean-square error

\[
\text{MSE} = E[||\hat{X}_k - X_k||^2] = E[\text{MSE}(Y_1, \ldots, Y_k)] \quad (3)
\]
The BKF provides the MMSE state estimator, according to both criteria above, and may be computed exactly in a recursive fashion, as shown in [4]. Briefly, let \((x_1, \ldots, x_{2^d})\) be an arbitrary enumeration of the possible state vectors. For each time \(k = 1, 2, \ldots\) define the posterior distribution vectors (PDV) \(\Pi_{k|k}\) and \(\Pi_{k|k-1}\) of length \(2^d\) by means of

\[
\Pi_{k|k}(i) = P(X_k = x^i | Y_1, \ldots, Y_k),
\]

\[
\Pi_{k|k-1}(i) = P(X_k = x^i | Y_1, \ldots, Y_k),
\]

for \(i = 1, \ldots, 2^d\). Let the prediction matrix \(M_k\) of size \(2^d \times 2^d\) be the transition matrix of the Markov chain defined by the state model: \((M_k)_{ij} = P(X_k = x^i | X_{k-1} = x^j) = P(n_k = x^i \oplus f(x^j))\), for \(i, j = 1, \ldots, 2^d\). Additionally, given a value of the observation vector \(y\), let the update matrix \(T_k(y)\), also of size \(2^d \times 2^d\), be a diagonal matrix defined by the observation model: \((T_k(y))_{jj} = p_{y_k}(h(X_k, v_k) = y | X_k = x^j)\), for \(j = 1, \ldots, 2^d\). Finally, define the matrix \(A\) of size \(d \times 2^d\) via \(A = [x^1 \cdots x^{2^d}]\).

The following result, which appears in [4], gives a procedure to compute the MMSE state estimator.

**Theorem 1.** (Boolean Kalman Filter.) The optimal minimum MSE estimator \(\hat{X}_k\) of the state \(X_k\) given the observations \(Y_1, \ldots, Y_k\) up to time \(k\), according to either criterion (2) or (3), is given by

\[
\hat{X}_k = \mathbb{E}[X_k | Y_1, \ldots, Y_k],
\]

where \(\mathbb{E}(\cdot)\) is the expectation of \(X_k\) conditioned on \(Y_1, \ldots, Y_k\). This estimator and its optimal conditional MSE can be computed by the following procedure.

1. **Initialization Step:** The initial PDV is given by \(\Pi_{0|0}(i) = P(X_0 = x^i),\) for \(i = 1, \ldots, 2^d\).

2. **Prediction Step:** Given the previous PDV \(\Pi_{k-1|k-1}\), the predicted PDV \(\Pi_{k|k-1}\) is given by \(\Pi_{k|k-1} = M_k \Pi_{k-1|k-1}\).

3. **Update Step:** Given the current observation \(Y_k = y_k\), let \(\beta_k = T_k(y_k) \Pi_{k|k-1}\). The updated PDV \(\Pi_{k|k}\) is obtained by normalizing \(\beta_k\) to obtain a probability measure: \(\Pi_{k|k} = \beta_k / \|\beta_k\|_1\).

4. **MMSE Estimator Computation Step:** The MMSE estimator is given by

\[
\hat{X}_k = \frac{\alpha}{\beta_k} \Pi_{k|k}
\]

with optimal conditional MSE

\[
\text{MSE}(Y_1, \ldots, Y_k) = \|\min\{\alpha \Pi_{k|k}, (\Pi_{k|k})^c\}\|_1,
\]

where the minimum is applied component-wise, and \(X_k^c(i) = 1 - X_k(i)\), for \(i = 1, \ldots, d\).

**3. PARTICLE FILTERING ALGORITHM**

Application of the BKF may become impractical computationally if the dimensionality of the state vector is too high. For example, with 16 boolean variables in the state vector, the total size of the state space is \(2^{16} = 65,536\), and the prediction matrix \(M\) is of size \(65,536 \times 65,536\), requiring \(2^{16} \times 2^{16} \times 8 = 2^{35} \approx 34\) GB of storage space.

We consider here a sequential Monte-Carlo approach, popularly known as a particle filter [3], in order to obtain an approximate solution to the optimal state estimation problem. We employ \(P(X_k | X_{k-1})\) as the proposal distribution for generating particles \(x_{k,i}\), and compute the particle weights \(Q_{k,i}\) via the likelihood \(p_{y_k}(h(x_k, v_k) = y_k | X_k = x_k)\). Resampling of the particles is done via multinomial sampling whenever the effective sample size \(N_{\text{eff}} = (\sum_{i=1}^{N} Q_{k,i}^2)^{-1}\) is smaller than a minimum size. The algorithm can be summarized as follows.

1. Initialization Step: Set \(N\) (number of particles), and \(N_{\text{eff}}\) (minimum effective sample size that triggers resampling).

   Sample \(N\) particles from the initial state distribution:

   \(x_{0,i} \sim \text{Multinomial}(N, \Pi_{0|0})\), for \(i = 1, \ldots, N\).

   For \(k = 1, 2, \ldots,\), do:

2. Prediction Step: Generate new particles from the old particles using the state process: \(x_{k,i} = f(x_{k-1,i}) \oplus n_k\), for \(i = 1, \ldots, N\).

3. Update Step: Given the current observation \(Y_k\), calculate the weights \(Q_{k,i} = p_{y_k}(h(x_{k,i}, v_k) = y_k | X_k = x_{k,i})\), for \(i = 1, \ldots, N\). The updated weights are given by

   \(Q_{k,i} = \frac{Q_{k,i}}{\sum_{j=1}^{N} Q_{j,i}}\), for \(i = 1, \ldots, N\).

4. Resampling Step: If \(N_{\text{eff}} = (\sum_{i=1}^{N} Q_{k,i}^2)^{-1} < N_{\text{eff}}\) then obtain new particles \(x_{k,i} \sim \text{Multinomial}(N, Q_{k,1}, \ldots, Q_{k,N})\), for \(i = 1, \ldots, N\). Set \(Q_{k,i} = 1/N\), for \(i = 1, \ldots, N\).

5. PDV Calculation: The updated PDV \(\Pi_{k|k}\) is obtained by

   \(\Pi_{k|k} = \sum_{i=1}^{N} Q_{k,i} \Pi_{k,i}\), for \(i = 1, \ldots, 2^d\).

6. Approximate MMSE Estimator Computation Step: The approximate MMSE estimator is given by

\[
\hat{X}_k = \frac{\alpha}{\beta_k} \Pi_{k|k}
\]

with optimal conditional MSE

\[
\text{MSE}(Y_1, \ldots, Y_k) = \|\min\{\alpha \Pi_{k|k}, (\Pi_{k|k})^c\}\|_1.
\]

Note that the MMSE estimator computation step is the same as in the BKF, except that the result here is approximate due to the Monte-Carlo computation.
4. NUMERICAL EXPERIMENTS

Here we illustrate the particle filtering approach described in the previous section by means of the following simple Boolean signal model, where both state and observation vectors are Boolean and of the same dimensionality.

\[
\begin{align*}
X_k &= f(X_{k-1}) \oplus n_k \quad \text{(state model)} \\
Y_k &= X_k \oplus v_k \quad \text{(observation model)}
\end{align*}
\]

for \( k = 1, 2, \ldots \). The network function \( f \) is selected randomly in our experiments, and the noise is i.i.d, \( n_k(i) \sim \text{Bernoulli}(p) \) and \( v_k(i) \sim \text{Bernoulli}(q) \), for \( i = 1, \ldots, d \), where \( 0 < p, q < 0.5 \). The noise parameter \( p \) gives the amount of “perturbation” to the Boolean state process; the closer it is to \( p = 0.5 \), the more chaotic the system will be, while a value of \( p \) close to zero means that the state trajectories are nearly deterministic, being governed tightly by the network function. Here, we set \( p = 0.05 \). On the other hand, \( q \) gives the intensity of the observation noise, being related to its variance \( q(1 - q) \). The noise is maximal at \( q = 0.5 \), while a value of \( q \) close to zero means that the observations are tightly correlated with the clean state signal. Estimation performance should be monotone with \( q \), being always better the smaller \( q \) is. Here, we consider a few values of \( q \), to examine the effect of observation noise on performance.

We employ \( N = 1000 \) particles and a threshold for efficient sample size \( N_T = 0.1N = 100 \). The simulation is carried out by sampling the initial state from the a uniform initial state distribution, and running the signal model in (11) to generate sample state and observation sequences \( X_k \) and \( Y_k \), respectively, for \( k = 1, \ldots, 100 \). The observation sequence \( Y_k \) is fed to the particle filter to obtain an estimated state sequence \( \hat{X}_k \). We also calculate the estimated error rate \( \epsilon = \frac{1}{100} \sum_{k=1}^{100} I(X_k \neq \hat{X}_k) \), where \( I \) is the indicator function. Values of \( \epsilon \) close to zero indicate accurate estimation performance.

Figure 1(a) and (b) display results for \( d = 8 \), the and \( q = 0.01 \) (clean observations) and \( q = 0.1 \) (noisy observations), both run on the same randomly generated network function. Displayed are the true (solid line) and estimated (dashed line) state paths, as well as the conditional MSE. We can see that the clean observations produce a much better result, with smaller MSE and error rate \( \epsilon \). It appears that the number of particles \( N = 1000 \) is insufficient to handle the noisy observation case.

Interestingly, running the filters again with the network function and all parameters fixed, but without a resampling step, produces a smaller error rate for the clean observation case, \( \epsilon = 6.06\% \), and a larger error rate for the noisy observation case, \( \epsilon = 58.58\% \), which seems to indicate that resampling helps in the noisy case, but is actually detrimental in the low-noise case. To verify this, we ran the particle filter on a system with \( d = 10 \) variables, \( p = 0.05 \), and \( q = 0.01 \) (clean observations), and plotted the error rate as a function of the number of particles used, both with and without the resampling step. In the former case, \( N_T = 0.1N \), as before. Figure 2 displays the plots. We can see that the error rate generally decreases with an increasing number of particles, which is expected, but we also see that the error rates are lower or the same without introducing a resampling step. With a small number of particles compared to the dimensionality of the problem, resampling appears to be detrimental to performance.

5. CONCLUSION

We presented in this paper a particle filtering approach to optimal state estimation in Boolean dynamical systems. Such an approach is necessary in the case of high-dimensional systems, where exact computation of the estimator may not be feasible. Numerical examples demonstrated the performance of the method under different noise conditions, number of particles, dimensionality, in addition to the presence or absence of a resampling step.

The use of a resampling step appears to be detrimental under low-noise conditions and smaller number of particles.
More work is necessary to confirm such observations. In addition, future work will investigate the use of the particle filtering approach in the system identification problem, where unknown parameters of the network function are estimated from the observations.

6. REFERENCES


