Learning Arbitrary State Transition Kernels for Prediction

Aaron P. Shon  Chris L. Baker  David B. Grimes  Rajesh P. N. Rao
{aaron,clbaker,grimes,rao}@cs.washington.edu
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195
March 25, 2004

Abstract
Predicting future states of noisy dynamical systems requires efficient inference algorithms that are robust to uncertainty. Markovian state-space models are especially effective at such tasks, allowing real-time inference and efficient training from data. Most prior work focuses on learning state-space models where state densities and dynamics have analytically tractable forms. We present a model that learns Markov processes with arbitrary state transition kernels and non-Gaussian, multi-modal state densities to predict the temporal evolution of nonlinear Markovian systems. The model uses radial basis function approximators to represent predictive state transition kernels. Conjugate gradient forms the basis of a generalized expectation-maximization algorithm for unsupervised learning of the function approximators. A particle-based state representation admits non-Gaussian state distributions. Particle smoothing estimates maximum a posteriori state sequences for model inference. After learning, the model can be used for prediction, filtering, or fixed-lag smoothing. We demonstrate the model’s robustness by predicting nonlinear, non-Gaussian state sequences from artificially generated noisy time-series data.

1 Introduction
Nonlinear dynamical systems are ubiquitous in nature. Such systems generate continuous-valued state sequences over time that describe, for instance, the kinematics of human and robotic limb movements, electroencephalogram signals, or waveforms of human speech. Given an initial probability distribution over state space locations, the states of such systems change over time according to some set of noisy differential equations. In modeling dynamical systems, a particularly effective method is to assume that states satisfy the Markov property, i.e. that future states only depend on one previous state. This approximation has allowed the development and tractable implementation of many algorithms for continuous and discrete state space models.

For linear dynamical systems with continuous, Gaussian-distributed state and observations that are a linear function of the state, corrupted by additive Gaussian noise, the Kalman filter predicts future states optimally. However, for nonlinear dynamical systems, no parametric models exist to estimate state values optimally. Several algorithms have been proposed for approximate state estimation, including the extended Kalman Filter (EKF) and the Unscented Kalman Filter [8, 14], but these models still assume Gaussian state, and can be sub-optimal in the case of multimodal distributions or highly nonlinear state transition kernels. Recently, particle filters were proposed as a nonparametric alternative for performing state estimation. Particle filters can accommodate nonlinear state transitions with non-Gaussian noise models and converge when the number of samples is sufficiently large [7].
Previous approaches to learning the dynamics for particle filters have focused on parameter estimation in parametric models. However, in many cases, the dynamical model of a particular system is completely unknown, and state densities may not assume a convenient parametric form. In this paper, we propose the use of radial basis function (RBF) approximators to approximate particle filter transition kernels. We introduce a generalized expectation-maximization (GEM) algorithm [2, 5] for maximum-likelihood model learning.

2 Generalized State Space Model

Many interesting problems in real-time machine prediction and control exhibit locally Markovian temporal structure. In principle, Markov state space models can be used in any situation where states approximately follow the Markov assumption—that future states only depend on one or a few previous state(s). In this paper, we attempt to learn partially observable Markov processes, making only weak assumptions about the nature of state transitions and state and observation noise.

Let the state transitions of a generalized Markovian state space model be governed by an arbitrary function $f$. Let $v(t)$ be additive noise with stationary distribution $p_v(.)$. Let $x(t)$ be an $m$-dimensional state vector, and let future states depend on previous states following:

$$ x(t+1) = f(x(t)) + u(t) + v(t). \quad (1) $$

Let $g$ be an observation function for $x(t)$. Let $r(t)$ be additive noise with stationary distribution $p_r(.)$. Let the $n$-dimensional observation vector $y(t)$ depend on the state according to

$$ y(t) = g(x(t)) + r(t). \quad (2) $$

In Markovian state space models, the expression for the predicted next state is

$$ p(x(t) | x(t-1)) = p_v\left(x(t) - f(x(t-1))\right). \quad (3) $$

The quantity in Eq. (3) is analytically tractable only for Gaussian-distributed $v(t)$ and for linear $f$. As mentioned previously, our model uses particle filtering to compute $p(x(t)|x(t-1))$ and thus permits any distribution for $v(t)$. In a particle filter, $p(x(t)|x(t-1))$ is approximated as a set of $N$ particles, each representing a weighted Dirac delta function. Let $X^i(t)$ be the $i$th particle in the prediction density, for $i \in \{1 \ldots N\}$. Let $v^i(t-1) \sim p_v(.)$ be a sample from the stationary state transition noise distribution. Next-state predictions are computed by:

$$ X^i(t) = f(x^i(t-1)) + v^i(t-1). \quad (4) $$

This approximation converges to Eq. (3) as $N \to \infty$.

The next distribution of interest is the filtering density, which is a probability distribution over states given the set of observations $y(1:t)$. The expression for the filtering density is

$$ p\left(x(t) \mid y(1:t)\right) = \int p(y_t \mid x_t) p\left(x(t) \mid x(t-1)\right) dx(t). \quad (5) $$

Like the prediction density, the filtering density is analytically tractable only for linear observation functions and Gaussian distributed $r(t)$. Through particle filtering, our model can handle non-Gaussian observation noise models and non-linear observation functions $g(x(t))$. The filtering density is approximated
by resampling from the set \( \{ \mathbf{x}^1 \ldots \mathbf{x}^N \} \) as follows. Let \( \mathbf{x}^i(t) \) be the \( i \)th particle in the filtering density, for \( i, j \in \{ 1 \ldots N \} \).

\[
P(\mathbf{x}^i(t) \leftarrow \mathbf{x}^j(t)) = \frac{w^j(t)}{\sum_{k=1}^{N} w^k(t)},
\]

where sampling weights are calculated by

\[
w^j(t) \propto p(y(t)|\mathbf{x}^i(t)) = p_r \left( y(t) - g(\mathbf{x}^i(t)) \right).
\]

The last quantity required for learning a state space model is the smoothing density, given by

\[
p(x(t) \mid y(1:T)) = \int \frac{p(x(t) \mid y(1:t))p(x(t+1) \mid x(t))}{p(x(t+1) \mid y(1:t))} \cdot p(x(t+1) \mid y(1:T)) \, dx(t+1).
\]

This density is analytically tractable only for linear \( f, g \) and Gaussian \( \mathbf{u}(t), \mathbf{v}(t), \mathbf{r}(t) \). However, particle smoothing allows this density to be approximated efficiently and accurately, and ultimately allows our to overcome the challenges associated with nonlinear smoothing in previous approaches.

Several other smoothing methods have been proposed previously. The algorithm given in [4] uses a first-order Taylor series approximation to invert nonlinear \( f \) to obtain the backward density. To obtain the smoothing density, Kalman smoothing is used. The approach in this paper suffers from significant bias and inaccuracy, especially in high-dimensional state spaces [9]. The MCHMM calculates a backward density using particle filtering. However, the convolution of the forward and backward densities to obtain a smoothing density relies on density trees [13], which can introduce significant approximation error.

Our model does not suffer these limitations, and uses the method given in [3] to calculate a smoothing density. The smoothing density is calculated backwards in time using particles obtained from particle filtering. Let \( \mathbf{x}^i(t) \) be the \( i \)th particle in the smoothing density, for \( i = 1 \ldots N \). The smoothing particle sets are sampled as follows:

\[
P(\mathbf{x}^i(t) \leftarrow \mathbf{x}^j(t)) = \frac{w^{i,j}(t, t+1)}{\sum_{k=1}^{N} w^{i,k}(t, t+1)},
\]

where sampling weights are calculated by

\[
w^{i,j}(t, t+1) \propto w^i(t) p(\mathbf{x}^j(t+1) \mid \mathbf{x}^i(t)) \]

\[= w^i(t) p_r \left( \mathbf{x}^j(t+1) - f(\mathbf{x}^i(t)) \right).
\]

Particle smoothing also allows a MAP state sequence to be calculated using the Viterbi algorithm. We denote the MAP state sequence as \( \mathbf{x}(1:T) \). The details of this algorithm are given in [3].

### 3 Model Learning

Learning arbitrary Markovian transition kernels for state space models is an important step toward real time processing of time series data. In cases where the form underlying dynamical system is almost completely unknown, rich transition kernels are required to perform filtering and prediction.

Learning a transition kernel for the particle filter uses an iterative generalized expectation-maximization algorithm. By the Markov property, the joint likelihood of the model factorizes as

\[
p(x(1:T), y(1:T)) = p(x(1)) \prod_{t=2}^{T} p(x(t) \mid x(t-1)) \prod_{t=1}^{T} p(y(t) \mid x(t)).
\]
This expression can be maximized using EM by iteratively computing the expected log likelihood and maximizing it with respect to the parameters of the RBF approximators. The expected log likelihood is given by

$$Q = E[p(x(1:T), y(1:T)) | y(1:T)].$$  \hspace{1cm} (12)

The expectation step of the algorithm computes the MAP state sequence given the observations and the current model parameters. This is essentially a sample-based version of forward-backward algorithm given in [11], and is most similar to the approach in [13].

The maximization step optimizes the parameters of the RBF approximators to maximize the likelihood of the model for the MAP state sequences. All parameters of the RBF approximators (i.e., means and covariances of each hidden unit, as well as linear weights) are changed during this step. For efficiency, our implementation uses diagonal hidden unit covariance matrices. The input-output training point pairs $\langle x(t-1), x(t) \rangle$, $t = 1 \ldots T$, for training the RBF approximator come from the MAP state sequences computed in the E step.

Unlike the model presented in [4], where the entire state space is gridded with evenly-spaced RBF hidden units, each with fixed covariance, our model modifies both hidden unit parameters and weights to learn transition kernels. Modifying hidden unit parameters precludes an analytical solution to model learning. Because the maximization step cannot be computed exactly, the model employs conjugate gradient to optimize the RBF hidden unit centers and covariances (see [12] for derivations of gradients and Hessians). After each conjugate gradient step, the algorithm applies an exact weight update step. Because our algorithm does not analytically solve for optimal parameter settings with respect to the data, the algorithm constitutes an example of generalized expectation maximization (GEM). Like EM, GEM is still guaranteed to converge to a local maximum of the likelihood function [10].

4 Results

The following sections present preliminary results on using the model to form predictive state representations. The first deals with synthetic time series in a 2-dimensional state space. The second deals with predicting optical flow data in the context of simple assembly tasks.

4.1 Predicting synthetic time series

Fig. 1 shows synthetic time series learned by the model. Hidden states are chosen on each time step from a Gaussian distribution with arbitrary means and noise covariance matrix $\Sigma_v = 0.001I$. Fig. 1(a-d) show a sequence of hidden states $x$ (circles), with blue through red respectively denoting the beginning through the end of each sequence. The model observes each hidden state plus i.i.d. zero-mean Gaussian noise (covariance $\Sigma_v = 0.01I$ for the examples shown here). MAP particles $\bar{x}$ are plotted as “x” signs, again with colors denoting the time step on which the model predicted each MAP particle. Note that the colors of the MAP particles match those of the hidden states, indicating that the model has correctly learned to predict what the next state will be. Fig. 2(a,b) show the corresponding vector fields representing the predictive dynamics learned for the sequence in Fig. 1(a,b).

Fig. 3 shows the model’s prediction for the path through state space on a single sample trajectory. This trajectory is nondeterministic; a bifurcation point exists. Thus, the particle filter propagates samples out to both arms of the bifurcation based on the learned dynamics encoded in the RBF network. The true trajectory, uncorrupted by additive noise, is plotted as black arrows.
Figure 1: **Learned particle representations of state densities:** (a,b) Particle clouds (colored points) show particle filter densities for two separate trajectories through a 2D synthetic state space, plotted over the actual hidden states (circles). Colors denote time steps, from blue (beginning of the sequence) to red (end of the sequence). (c,d) MAP particles from the above sample sets, plotted over the actual hidden states (circles). Colors are the same as in (a,b). Note that colors for the true hidden states approximately match those of the MAP particles, indicating that the model has learned the correct temporal ordering for the points in the filtering density.

4.2 Predicting hand gestures in a simple assembly task

An ongoing, more practically relevant task is to predictively model actions in a simple assembly task. We briefly outline the problem and offer preliminary results. The idea here is to observe images containing a person’s hands acting on some object(s) of interest, beginning in a well-known configuration and ending at some defined point in time. This could represent, e.g., an assembly task conducted on simple, block-like objects.

To isolate the hands, we perform some preprocessing steps. First, a skin detection algorithm (based on work by Collins and Liu [1]) isolates skin regions from background pixels. After filtering out candidate regions to find hands based on size and aspect ratio, we perform optic flow on the resulting grayscale image using the Horn-Schunck algorithm [6]. Principal components analysis further reduces the optic flow data to 10-dimensional vectors. The matrix of principal components thus represents the model’s observation process.

After performing dimensionality reduction to infer state values, the model learns consecutive pairs of 10-dimensional vectors to form a predictive representation of the assembly task. Fig. 4(a,b) show two (grayscale) frames of a hand performing a movement against a background. Fig. 4(c,d) show corresponding
principal component vectors recovered for the horizontal and vertical optical flow, respectively, plotted against grayscale images of the raw optical flow (light pixels represent high-magnitude positive optical flow, dark pixels represent high-magnitude negative flow, and gray pixels represent low-magnitude flow).

Preliminary results suggest the model is indeed able to learn this task. We continue to refine our implementation to improve speed and predictive accuracy. Possible future applications include a general algorithm for action recognition and prediction, and learning predictive models of how the environment responds to actions for use in active robotics platforms.

5 Conclusion

This paper proposed a new model for learning state transition kernels for particle filters. To our knowledge, the proposed approach is the first to combine learning an efficient function approximator capable of interpolation with a predictive, particle-based state representation. The model is versatile enough to admit any implementation of particle filtering and function approximation, and should be most applicable in situations where efficient, non-parametric learning of times series is desired. We presented preliminary results illustrating the model’s performance learning synthetic mixtures of nonlinear dynamical systems, and described ongoing work on a computer vision learning task. We plan to investigate applications of the model to speech processing, EEG-based brain-computer interfaces, and motor control in an active vision stereo head.

References

Figure 3: Prediction of synthetic data: The model is able to predict even in the presence of nondeterministic trajectories. Because we use a sample-based representation, the model maintains a multimodal state density, allowing it to accommodate bifurcation points. Colored points represent predicted state estimates, color coded from blue (early in the trajectory) to red (late in the trajectory). Black arrows show how the true hidden state evolves over time, uncorrupted by noise. Although most particles follow a blurred trajectory between the two arms of the bifurcation, a significant number follow each arm of the bifurcation, representing correct state predictions.

Figure 4: **Raw observations and hidden state representations for simple assembly task:** (a,b) Raw images (observations) for temporally consecutive images. (c,d) Horizontal and vertical optical flow images. Green arrows represent principal component vectors (with the highest associated eigenvalues) extracted from each optical flow image pair. The top ten such vectors from each image pair are presented to the RBF network to learn a predictive kernel.