Predictive Linear-Gaussian Models of Controlled Stochastic Dynamical Systems

Matthew Rudary MRUDARY@UMICH.EDU
Satinder Singh BAVEJA@UMICH.EDU
Computer Science and Engineering, University of Michigan, Ann Arbor, MI, USA

Abstract

We introduce the controlled predictive linear-Gaussian model (cPLG), a model that uses predictive state to model discrete-time dynamical systems with real-valued observations and vector-valued actions. This extends the PLG, an uncontrolled model recently introduced by Rudary et al. (2005). We show that the cPLG subsumes controlled linear dynamical systems (LDS, also called Kalman filter models) of equal dimension, but requires fewer parameters. We also introduce the predictive linear-quadratic Gaussian problem, a cost-minimization problem based on the cPLG that we show is equivalent to linear-quadratic Gaussian problems (LQG, sometimes called LQR). We present an algorithm to estimate cPLG parameters from data, and show that our algorithm is a consistent estimation procedure. Finally, we present empirical results suggesting that our algorithm performs favorably compared to expectation maximization on controlled LDS models.

1. Introduction

Linear dynamical system models (LDSs), also known as Kalman filter models or state-space models, are widely used in control and prediction tasks in many applications from a variety of fields. These models are very useful when their parameters are known in advance. However, their parameters are not easily learned. This learning is typically achieved using expectation maximization (EM). EM finds parameters that *locally* optimize the expected likelihood of the data, so it can learn an inaccurate model.

Recently, Rudary et al. (2005) introduced the predictive linear-Gaussian model (PLG). Any *uncontrolled* LDS with real-valued observations has an equivalent PLG. Rudary et al. also introduced a parameter estimation algorithm that obtains a consistent estimate of the parameters—that is, as the dataset increases in size, the estimate converges in probability to the true parameter values. The state of the PLG is based on predictions about the future outcomes in the system—that is, it is a predictive state representation (PSR). In domains with discrete observations, PSRs have advantages over some traditional models. For example, they have more expressive power than hidden Markov models and partially observable Markov decision processes (POMDPs) (Jaeger, 1997; Singh et al., 2004), and learning one type of PSRs from data outperforms learning POMDPs using EM on many small problems (Wolfe et al., 2005). One of the strengths of PSRs, including PLGs, is their lack of reference to latent state variables.

In this work, we introduce a new model of stochastic dynamical systems with real-valued observations and vector-valued actions, the *controlled* predictive linear-Gaussian model (cPLG). We show that the cPLG subsumes controlled LDSs, and that optimal actions can be computed as with LDSs, even without estimating the hidden state variables. In addition, we introduce a consistent parameter estimation algorithm for cPLGs, and experimentally compare it to parameter estimation of LDSs using EM.

2. The cPLG Model

The cPLG model, like all models of dynamical systems, computes the probability of future outcomes given the history of past interactions with the system; these interactions consist of taking actions (each action is a vector from $\mathcal{R}^l$) and receiving observations (from $\mathcal{R}$). The cPLG is as powerful as the controlled LDS, and its modeling power rests on a few properties of the system. The future observations in the system are jointly Gaussian random variables, and the distribution of all of them can be computed from the distribution of just the next n; n is thus the dimension of the system. The distribution is extended beyond those n observations by a linear function—that is, the $(n+1)$st ob-

Appearing in *Proceedings of the 23^{rd} International Conference on Machine Learning*, Pittsburgh, PA, 2006.

servation in the future is a noisy linear function of the next n observations. This noise is *not* i.i.d. The actions also have a linear effect on future observations. The dynamics of the system are illustrated by the following equation:

$$Y_{t+n+1} = g' Z_t + \sum_{i=0}^{n-1} \Gamma_i' u_{t+i+1} + \eta_{t+n+1}. \qquad (1)$$

We will explain the meaning of (1) as we develop the mathematics of the model and describe the above properties of the system in more detail.

First, this equation should be viewed from the standpoint of extending the distribution of observations further into the future, starting from t. The random variable Y_t models the observation at time t (a realization of that observation will be denoted y_t). As mentioned above, the observation $n+1$ timesteps in the future is a function of the next n observations; these next n are collected into the vector Z_t (i.e., $Z_t = [Y_{t+1}\ Y_{t+2}\ \cdots\ Y_{t+n}]'$, where A' is the transpose of the vector/matrix A). In fact, it is a linear function, as g is a vector describing the linear effect of the next n observations on the $(n+1)$st.

Since Z_t models observations in the future, knowing that Y_{t+n+1} has a linear dependence on Z_t is useless without knowing the distribution of Z_t. In fact, Z_t is a Gaussian random vector. The initial n observations, Z_0, are normally distributed with mean μ_0 and covariance Σ_0 (which we write as $Z_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$); μ_0 and Σ_0 are the initial state of the system, and are parameters of the cPLG. As actions are taken and observations are seen, new state variables μ_t and Σ_t can be computed. Let h_t represent the history of interaction with the system through time t, viz. $u_1, y_1, u_2, y_2, \ldots, u_t, y_t$ (where u_t is the action at time t). Then the distribution of the next n observations are given by the state at t: $Z_t | h_t \sim \mathcal{N}(\mu_t, \Sigma_t)$.

The second term of (1) describes the effect of the actions on future observations. Each Γ_i is an l-vector that defines the linear effect that the action u_{t+i+1} has on the future observation Y_{t+n+1}. Because observations in the future are affected by future actions, and because we have no distributional information about those actions, we must assume something about future actions to make statements about the future. So we assume that future actions will be zero (i.e. will have no effect). This means the statement above about the state variables was incomplete; the full semantics of μ_t and Σ_t are that

$$Z_t | h_t, u_{t+1} = \mathbf{0}, u_{t+2} = \mathbf{0}, \ldots \sim \mathcal{N}(\mu_t, \Sigma_t).$$

The final term in (1) models the uncertainty in the future observation. That is, η_{t+n+1} is the noise term. Clearly, in order for Y_{t+n+1} to be Gaussian, η_{t+n+1} must be a Gaussian random variable as well. But it need not be independent of Z_t. In fact, we allow the noise term for the observation $n+1$ steps in the future to covary with the intervening n future observations. That is,

$$\eta_{t+n+1} | h_t \sim \mathcal{N}(0, \sigma^2) \quad \text{and} \quad \mathrm{Cov}[Z_t, \eta_{t+n+1} | h_t] = C,$$

where σ^2 and C are independent of time. This is contrary to many models, including the LDS, that assume i.i.d. noise.

With these fundamentals laid out, we can define a cPLG. The parameters of the model are the initial state variables, μ_0 and Σ_0; the linear trend in the observations, g; the additive linear functions of the actions, $\Gamma_0, \ldots, \Gamma_{n-1}$; the variance of the noise term, σ^2; and the covariance of the noise term with the observations, C. The state at time t is given by the mean vector and covariance matrix of the n observations immediately in the future, μ_t and Σ_t. After fixing u_{t+1} and observing $Y_{t+1} = y_{t+1}$, we can update the state:

$$\mu_{t+1} = G\mu_t + L u_{t+1} + \frac{f_t}{e_1' \Sigma_t e_1}(y_{t+1} - e_1' \mu_t), \text{ and } (2)$$

$$\Sigma_{t+1} = G\Sigma_t G' + F - \frac{f_t f_t'}{e_1' \Sigma_t e_1}, \qquad (3)$$

where e_i is the ith column of the $n \times n$ identity matrix,

$$G = \begin{pmatrix} \mathbf{0} & I_{n-1} \\ \hline g' & \end{pmatrix},$$

$f_t = G\Sigma_t e_1 + (e_1' C) e_n$, $F = \sigma^2 e_n e_n' + G C e_n' + e_n C' G'$, and I_{n-1} is the $(n-1) \times (n-1)$ identity matrix. L is the linear effect of the next action on the next n observations. Its ith row is $L_i = \Gamma_{n-1}' + \sum_{j=1}^{i-1} (e_{n-i+j}' g) L_{j-i}$ for $i = 1, \ldots, n$. See Appendix A.1 for the derivation of (2) and (3). We can now state one of our main results:

Theorem 1. *Any controlled LDS with n-dimensional state and scalar observations has an equivalent n-dimensional cPLG.*

That is, any system (with scalar observations) that can be modeled by a controlled LDS can be modeled as compactly by a cPLG.[1] Indeed, it can be modeled even *more* compactly; the cPLG achieves equivalent representative power to the LDS with $(3n^2 - n)/2$ fewer parameters. We defer the proof of the theorem until Appendix A.3.

3. Optimal Control of the cPLG

One of the principle purposes of a controlled model is to use it to maximize a reward function, or, equivalently, to

[1] This is despite the superficial resemblance of (1), the core dynamics of the cPLG, to the ARX model, which is considerably weaker in representative power than the LDS (Ljung, 1987). The major differences are the non-i.i.d. noise and that (1) looks forward where the ARX looks back.

minimize a cost function. Here we can look to the traditional LDS model for inspiration—one of its attractive features is that, under a quadratic cost function, the optimal action is the certainty-equivalent (CE) action. The CE action is the one that would be optimal if the model were noise-free and the state equal to the least-squares estimate (i.e., the Kalman filter estimate). Moreover, the optimal CE action is a linear function of the state estimate.

We will introduce the predictive linear-quadratic-Gaussian (PLQG) framework, which applies a quadratic cost function to a cPLG model, and show that the optimal action in this framework is a linear function of the mean vector μ_t. We will also show that, under reasonable conditions, a cost function can be selected that yields the same optimal controls for an LDS and its equivalent cPLG.

Quadratic Cost Function Much work has focused on the linear-quadratic-Gaussian (LQG) control problem; that is, the problem of minimizing a quadratic cost function on an LDS model.[2] In the PLQG problem, on the other hand, the goal is to minimize a quadratic cost function on a *cPLG* model. At each timestep t (up to a horizon of T), the PLQG framework assesses a cost that is quadratic in the mean vector μ_{t-1} and the control u_t.[3] At the end of the horizon, a terminal cost is added. That is, the PLQG quadratic cost function J^μ can be written as

$$J^\mu = \sum_{\tau=1}^{T} j^\mu(\mu_{\tau-1}, u_\tau) + \mu_T' W_{\mu,f} \mu_T$$

where $j^\mu(\mu, u) = \mu' W_\mu \mu + 2u' W_{\mu,u} \mu + u' W_u u$ is the per-timestep cost. We restrict W_μ and $W_{\mu,f}$ to be symmetric positive semidefinite and W_u to be symmetric positive definite (SPD).

This is quite similar to the cost function in the traditional LQG framework. In that framework, a cost quadratic in the latent state and action is assessed at each timestep up through the cost horizon, T; finally, a terminal cost quadratic in the final state is added. The LQG quadratic cost function is J^x (we use the superscript x to refer to the LQG cost function, which is a function of the latent state X_t; the superscript μ refers to the PLQG cost function, a function of the mean vector μ_t). This can be written as

$$J^x = \sum_{\tau=1}^{T} j^x(X_\tau, u_\tau) + X_{T+1}' W_{x,f} X_{T+1},$$

where $j^x(X, u) = X' W_x X + 2u' W_{x,u} X + u' W_u u$ is the per-timestep cost. We restrict W_x and $W_{x,f}$ to be symmetric positive semidefinite and W_u to be SPD.

[2] See Appendix A.2 for a definition of the LDS.

[3] The time indices of μ_{t-1} and u_t differ by one because the initial mean vector is μ_0; this is the information used by the PLQG to select the first action u_1.

Quadratic cost functions of this type are quite flexible; they can be used to minimize the sum-of-squares of the states or actions, to minimize energy expenditures (which may depend on the product of state and action), etc. The LQG is used in a wide variety of control applications, including chemical plant control, aircraft control, and vibration cancellation. The PLQG's cost function is just as flexible; we will show that, under reasonable conditions, the PLQG can represent any problem the LQG can, and in particular that a cost function can be selected such that the optimal action in both frameworks is the same.

Minimizing the Cost Function At a given time t, after observing a sequence h_{t-1} of observations and controls through time $t-1$, the system must select a sequence of actions $u_t(h_{t-1}), \ldots, u_T(h_{t-1})$ that minimizes the expected cost. However, this can be cast as a dynamic programming problem, so that $u_t(h_{t-1})$ can be selected independently from future actions.

We will now show that the optimal action $u_t(h_{t-1})$ can be computed as a linear function of the mean vector μ_{t-1}.

Lemma 2. *The optimal action $u_t(h_{t-1})$ to be taken at time t after observing history h_{t-1} is the CE optimal action and a linear function of the mean vector μ_{t-1}.*

Proof. Let $J_t^\mu(h_{t-1})$ be the optimal expected cost-to-go at time t after observing h_{t-1}:

$$\begin{aligned} J_t^\mu(h_{t-1}) &= \\ &= \min_{u_t,\ldots,u_T} \mathrm{E}\left[\sum_{\tau=t}^{T} j^\mu(\mu_{\tau-1}, u_\tau) + \mu_T' W_{\mu,f} \mu_T \,\middle|\, h_{t-1} \right] \\ &= \min_{u_t} \mathrm{E}[j^\mu(\mu_{t-1}, u_t) + J_{t+1}^\mu(h_{t-1}, u_t, Y_t) | h_{t-1}]. \end{aligned}$$

This can be divided into a history-independent constant and a history-dependent part that is quadratic in the state: $J_t^\mu(h_{t-1}) = v_t^\mu + \mu_{t-1}' V_t^\mu \mu_{t-1}$, where v_t^μ and V_t^μ are constants. Now we can compute the expectation:

$$\begin{aligned} J_t^\mu(h_{t-1}) = \min_{u_t} [&(G\mu_{t-1} + Lu_t)' V_{t+1}^\mu (G\mu_{t-1} + Lu_t) \\ &+ j^\mu(\mu_{t-1}, u_t) + v_t^\mu], \end{aligned}$$

where $v_t^\mu = \mathrm{tr}(V_{t+1}^\mu \frac{f_{t-1} f_{t-1}'}{e_1' \Sigma_{t-1} e_1}) + v_{t+1}^\mu$ ($\mathrm{tr}(A)$ denotes the trace of the matrix A, i.e. the sum of the elements on its diagonal). Since f_{t-1} and Σ_{t-1} are independent of the system's history, v_t^μ is not affected by the actions and can be ignored in the minimization; hereafter we will focus only on the quadratic, history-dependent part of J_t^μ.

The optimal action at t can be computed by taking the derivative of J_t^μ with respect to u_t, and then solving for the zero. We thus get a linear function of the mean vector μ_{t-1}:

$$u_t(h_{t-1}) = -(W_u + L'V_t^\mu L)^{-1}(W_{\mu,u} + L'V_t^\mu G)\mu_{t-1} \triangleq \Pi_t^\mu \mu_{t-1}.$$

It can be shown that $v_t^\mu = 0$ for all t when there is no uncertainty, but that V_t^μ remains unchanged. Thus this is also the CE optimal action. □

Having computed the optimal action, this result can now be used to compute the quadratic part of J_t^μ:

$$V_t^\mu = W_z + G'V_{t+1}^\mu G - (W'_{\mu,u} + G'V_{t+1}^\mu L) \times (W_u + L'V_{t+1}^\mu L)^{-1}(W_{\mu,u} + L'V_{t+1}^\mu G).$$

The recursion is initialized by $V_{T+1}^\mu = W_{\mu,f}$.

We have already seen that for any n-dimensional controlled LDS, there is an equivalent n-dimensional cPLG. It is also the case that for any LQG based on an n-dimensional controlled LDS with full rank,[4] there is an equivalent PLQG based on an n-dimensional cPLG. It is equivalent in the following sense: Given a sequence h_{t-1} of actions and observations, both formalisms will select the same optimal action u_t, and the optimal expected cost-to-go computed by each will differ by a constant independent of h_{t-1}.

Theorem 3. *For any n-dimensional LQG with full rank, an equivalent n-dimensional PLQG exists that, given any history of interaction with the system, computes the same optimal action as the LQG.*

In other words, the PLQG can be used to specify and solve the same control problems as full-rank LQGs. The proof of Theorem 3 can be found in Appendix A.4.

4. Parameter Estimation in cPLGs

The cPLG model has several potential advantages over the LDS with respect to parameter estimation. First, LDS models have more parameters than cPLGs of the same dimension, and that parameter space has inherent symmetries not present in cPLG parameter space. For example, two distinct LDS models describe the same system if any two elements of the X_t state vector (and corresponding rows and columns of the parameters) are swapped. This symmetry can cause difficulties in learning LDS models.

Another important advantage is that the cPLG parameters have a definite meaning in relation to the data. For instance, μ_0 is the expected value of the first n observations. LDS parameters do not have such an interpretation; A, for example, is the linear trend in the latent variables, which has no direct connection to the data. We exploit the meaning of the parameters in the cPLG to estimate the parameters through the Consistent Estimation algorithm (CE).

[4] We define a full-rank LDS as one for which M has full rank (see (8) for a definition of M). If M is rank-deficient, for any state x_t there are infinitely many distinct states that place the same distribution over future trajectories.

4.1. The CE Algorithm

The CE algorithm takes two inputs: the dimension of the system (n) and a dataset consisting of multiple trajectories collected through interaction with the system. The dataset contains K such trajectories, each of which is N timesteps long. We label the tth action and observation from the kth trajectory as u_t^k and y_t^k, respectively. Thus, the kth trajectory is the sequence $u_1^k, y_1^k, u_2^k, y_2^k, \ldots, u_N^k, y_N^k$—note that this follows our convention that the realization of a random variable (in this case, the observation Y_t) is set in lowercase. We assume that each trajectory starts with the system in its initial configuration. Because of the importance of the variable Z_t, the random vector representing the n observations following t, we also collect the data into vectors $z_t^k = (y_{t+1}^k \ \cdots \ y_{t+n}^k)'$.

As we describe the CE algorithm, it will be convenient to consider subgroups of the parameters separately.

Linear Trends The first parameters to be estimated are the trend parameters g and Γ_i, $i = 0, \ldots, n-1$. Note that Y_{t+n+1} is a linear function of the data plus a noise term, η_{t+n+1}. Averaging over the dataset,

$$\overline{y}_{t+n+1} = g'\overline{z}_t + \sum_{i=0}^{n-1} \Gamma_i' \overline{u}_{t+1+i} + \overline{\eta}_{t+n+1}$$

for $t = 1, \ldots, N-n-1$, where a bar over a variable denotes its average over the K trajectories in the dataset. We can collect all these equations together in a single matrix equation $\Xi\gamma + \epsilon = \Upsilon$, where

$$\Xi = \begin{pmatrix} \overline{z}_0' & \overline{u}_1' & \cdots & \overline{u}_n' \\ \overline{z}_1' & \overline{u}_2' & \cdots & \overline{u}_{n+1}' \\ \vdots & \vdots & \ddots & \vdots \\ \overline{z}_{N-n-1}' & \overline{u}_{N-n}' & \cdots & \overline{u}_{N-1}' \end{pmatrix},$$

$$\gamma = \begin{pmatrix} g \\ \Gamma_0 \\ \vdots \\ \Gamma_{n-1} \end{pmatrix}, \quad \epsilon = \begin{pmatrix} \overline{\eta}_{n+1} \\ \overline{\eta}_{n+2} \\ \vdots \\ \overline{\eta}_N \end{pmatrix}, \quad \Upsilon = \begin{pmatrix} \overline{y}_{n+1} \\ \overline{y}_{n+2} \\ \vdots \\ \overline{y}_N \end{pmatrix}.$$

We can estimate g and Γ_i by the equation $\widehat{\gamma} = (\Xi'\Xi)^{-1}\Xi'\Upsilon$. Since the rows of ϵ are not independent of each other, this is a biased estimator. However, we will see that it is *consistent* under certain conditions; that is, the probability that the error is larger than some positive constant shrinks to zero as the number of trajectories grows.

Initial State The next group of parameters to estimate are the initial state parameters μ_0 and Σ_0. Recall that $\mu_0^i = \mathrm{E}[Y_i | u_1 = \mathbf{0}, u_2 = \mathbf{0}, \ldots]$ (where μ_0^i is the ith element of μ_0). The sample mean is a natural estimator here, but we must account for the effect of the actions that were actually taken. Since we have estimates for g and Γ_i, we can compute $\widehat{L}$, an estimate of L. Using $\widehat{L}$, the estimate of the initial mean is computed by

$$\widehat{\mu}_0^i = \overline{y}_i - \sum_{j=1}^{i-1} \widehat{L}_j \overline{u}_{i-j}, \quad i = 1, \ldots, n.$$

To estimate Σ_0 using sample covariances, we must first compute estimates $\widehat{\mathrm{E}Y}_t^k$ of $\mathrm{E}[Y_t | u_1^k, \ldots, u_{t-1}^k]$ for $t = 1, \ldots, n$. This again requires the use of $\widehat{L}$ as well as the newly computed $\widehat{\mu}_0$; $\widehat{\mathrm{E}Y}_t^k$ is computed by starting with the initial mean and then adding in the effect of the action actions of trajectory k:

$$\widehat{\mathrm{E}Y}_t^k = \widehat{\mu}_0^t + \sum_{i=1}^{t-1} \widehat{L}_i u_{t-i}^k, \quad t = 1, \ldots, n.$$

We then use these in the standard sample covariance calculation to obtain the estimate of Σ_0. For $i, j = 1, \ldots n$:

$$\widehat{\Sigma}_0^{ij} = \frac{1}{K-1} \sum_{k=1}^{K} (y_i^k - \widehat{\mathrm{E}Y}_i^k)(y_j^k - \widehat{\mathrm{E}Y}_j^k).$$

Noise Parameters The only parameters remaining are C and σ^2, the distributional parameters of the noise terms η_{t+n+1}^k. These are computed using straightforward calculations of sample variance and covariance of the noise terms; this requires estimating those noise terms. To accomplish this, we solve (1) for η_{t+n+1} and replace variables with data and parameters with estimates:

$$\widehat{\eta}_{t+n+1}^k = y_{t+n+1}^k - \widehat{g}' z_t^k - \sum_{i=1}^{n-1} \widehat{\Gamma}_i' u_{t+i+1}^k.$$

Now we can estimate C and σ^2 (taking advantage of the fact that $\mathrm{E}[\eta_{t+n+1}] = 0$):

$$\widehat{\sigma}^2 = \frac{1}{K(N-n)-1} \sum_{t=0}^{N-n-1} \sum_{k=1}^{K} (\widehat{\eta}_{t+n+1}^k)^2 \quad \text{and}$$

$$\widehat{C} = \frac{1}{K(N-n)-1} \sum_{t=0}^{N-n-1} \sum_{k=1}^{K} z_t^k \widehat{\eta}_{t+n+1}^k.$$

4.2. Convergence of the CE Algorithm

As implied by the name of the algorithm, CE produces *consistent* estimates of the parameters. However, there are certain requirements on the system and on the actions used to create the training data for this to hold; such a system and policy taken together are called *CE-learnable*, which we will define below.

More formally, the sequence of estimates produced by increasing the number of trajectories K (in a CE-learnable system) *converges in probability* to the true parameters, where convergence in probability is defined as follows:

Definition 1. *The sequence $\widehat{x}_1, \widehat{x}_2, \ldots$ converges to x in probability if $\lim_{n\to\infty} \Pr(|\widehat{x}_n - x| > \delta) = 0$ for positive δ. We write this as "$\widehat{x}_n \xrightarrow{p} x$ as $n \to \infty$."*

However, in order for the system to be CE-learnable, a minimal condition must be satisfied: $\Xi'\Xi$ must be invertible. In particular, it must be invertible in the limit as the number of trajectories grows. This is the only requirement for CE-learnability, but it bears some further discussion.

Assume that the actions u_t are generated by some stochastic policy such that $\mathrm{E}[u_t] = \pi_t$. Then, by the weak law of large numbers, $\Xi \xrightarrow{p} \Xi^*$ as $K \to \infty$, where

$$\Xi^* = \begin{pmatrix} \mathrm{E}[z_0' | u_i = \pi_i \forall i] & \pi_1' & \cdots & \pi_n' \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}[z_{N-n-1}' | u_i = \pi_i \forall i] & \pi_{N-n}' & \cdots & \pi_{N-1}' \end{pmatrix}.$$

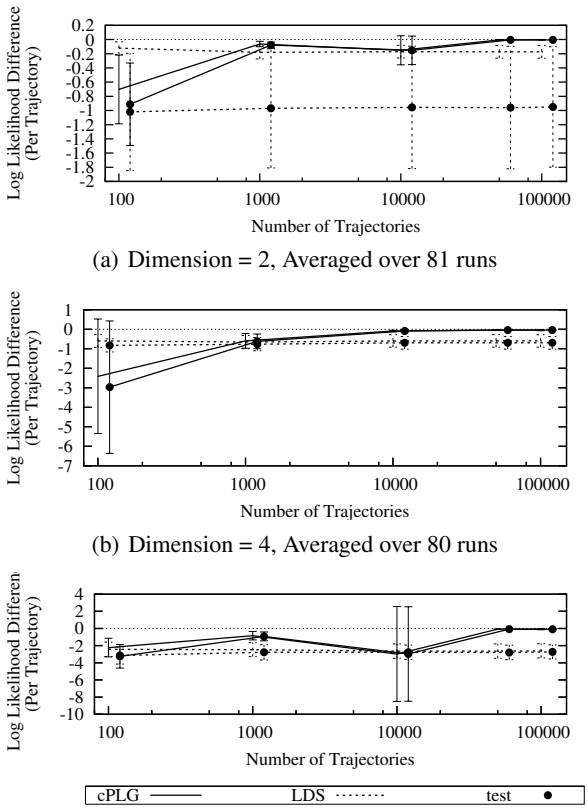
If Ξ^* has rank $(l+1)n$, $(\Xi^{*\prime}\Xi^*)^{-1}$ will exist (because the inner product is positive semi-definite of rank equal to Ξ^* and is a square matrix of dimension $(l+1)n$).

One necessary condition for learnability, then, is that the "π" part of Ξ^* (i.e. all but the n left-most columns) must have full rank. So actions selected from a mean-zero Gaussian distribution, for example, are not compatible with CE. One compatible policy (which we use in our experiments) uses a periodic sequence for the means of the actions. Here, only every nth action has a non-zero mean. The mean vector for those actions each have a single non-zero element that rotates in turn—so the nth action is drawn from a distribution with mean $(a\ 0\ 0\ \ldots)'$ (for positive a), the $(2n)$th has a mean of $(0\ a\ 0\ \ldots)'$, and so on. Therefore, after nl actions, each element of the action vector has been exercised once. In our experiments, we chose $a = 2$, and drew actions from a Gaussian distribution with means as just described and variance 1.

As stated above, when the system and training policy are CE-learnable, the parameter estimates are consistent.

Theorem 4. *If a dynamical system can be modeled by an n-dimensional cPLG and generates a training set whose trajectories are at least $(l+2)n$ timesteps long using a policy that is CE-learnable, then, as the number of trajectories K grows, the parameter estimates computed by the CE algorithm will converge in probability to the true parameters of that cPLG.*

We defer the proof until Appendix A.5.



(a) Dimension = 2, Averaged over 81 runs

(b) Dimension = 4, Averaged over 80 runs

(c) Dimension = 8, Averaged over 78 runs

Figure 1. Comparison of parameter estimation using EM with LDS and CE with cPLGs. Higher values indicate higher likelihood in the learned parameters; at zero, the likelihood of the learned parameters is identical to that of the generating parameters. Results on the test sets are offset on the horizontal axis.

4.3. Experimental Results

We compare Consistent Estimation for cPLGs to Expectation Maximization for LDSs (Ghahramani & Hinton, 1996) by generating datasets using random test systems, training models on the datasets using both algorithms, and comparing the likelihood of the data using the trained models. We found that CE outperformed EM, though there were a few errors with smaller datasets.

The test systems were LDSs with parameters generated randomly as by Rudary et al. (2005); B was generated in the same manner as H. Systems were generated with dimensions (n) of 2, 4, and 8, each with a 1-dimensional action. These systems were then used to create datasets with 100,000 trajectories, each of which was of length $5n$. For each of these datasets, we trained cPLGs and LDSs using the first 100, 1000, 10,000, 50,000, and 100,000 trajectories. The LDSs were trained using the EM software made available by Ghahramani (2002); EM stopped once the likelihood of the data changed less than 0.01% between iterations or once 1000 iterations had been completed. The plots in Figure 1 report the average error measure $\frac{l_t - l_a}{K}$ vs K, where l_t is the log-likelihood of the data with the trained parameters, l_a is the log-likelihood of the data given the actual parameters, and K is the number of trajectories used to train the model, with error bars showing one standard deviation. We reported the results both for the training data and for a test dataset of 10000 trajectories generated by the same model as the training set.

The update of Σ_t was modified from (3). Because of the nature of the CE algorithm and the cPLG model, the standard covariance update sometimes yields a matrix that is not positive semidefinite. When this occurred, $10I$ was added to Σ_t. This was occasionally necessary for smaller values of K, rarely for larger. The dips in accuracy at $K = 10000$ in 1(a) and 1(c) reflect a single dataset each for which this was necessary. These errors are offset by the superior performance of CE, particularly on the larger datasets. On all three model sizes considered, the likelihood of the data by the cPLGs approached the true likelihood as the training set increased in size, as would be expected given Theorem 4. On the other hand, the performance of EM did not appear to improve as the dataset grew.

5. Conclusion

We have introduced the *controlled* predictive linear-Gaussian model, a predictive state representation of discrete-time dynamical systems with real-valued observations and vector-valued actions. We have shown that n-dimensional cPLGs subsume n-dimensional linear dynamical systems while requiring fewer parameters.We have also introduced the predictive linear-quadratic Gaussian problem, a cost-optimization problem based on cPLGs. We showed that the optimal action in a cPLG under a quadratic cost function is the certainty equivalent optimal action, and a linear function of the state. We have shown an equivalence between the PLQG and LQG problems. Finally, we have proposed a consistent parameter estimation algorithm for cPLGs, CE, and shown that it compares favorably to EM for LDS models in experiments.

While these are promising early developments, work remains to be done. First, outperforming EM on random problems indicates that CE and cPLGs show promise, but this is not the gold standard; EM is particularly useful for refining parameters in an "almost known" LDS, for example, and other parameter estimation algorithms exist. Second, the invalid models that are sometimes learned with small datasets are worrisome, as well as the fact that CE cannot be used to estimate parameters from a single (long) trajectory, as EM can be. Thus, a new (or modified) parameter estimation algorithm is an important area of future work—we are exploring a maximum likelihood algorithm. Another area of future work is the development of a PLG model that allows vector-valued observations; PLGs could

then model anything that an LDS with Gaussian noise can.

Acknowledgements

The authors would like to thank David Wingate for enlightening conversations. This work is supported by the National Science Foundation under Grant Number IIS-0413004 and by a grant from Intel Corp. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or of Intel.

References

Catlin, D. E. (1989). *Estimation, control, and the discrete Kalman filter*. Springer-Verlag New York.

DeGroot, M. H., & Schervish, M. J. (2002). *Probability and statistics*. Addison Wesley.

Ghahramani, Z. (2002). *Machine learning toolbox v1.0 [Computer software]*. Retrieved February 16, 2005, www.gatsby.ucl.ac.uk/~zoubin/software/ lds.tar.gz.

Ghahramani, Z., & Hinton, G. E. (1996). *Parameter estimation for linear dynamical systems* (Technical Report CRG-TR-96-2). Dept. of Computer Science, U. of Toronto.

Jaeger, H. (1997). *Observable operator models II: Interpretable models and model inductions* (Arbeitspapiere der GMD 1083). GMD, St. Augustine.

Ljung, L. (1987). *System identification: Theory for the user*. Prentice-Hall, Inc.

Rudary, M., Singh, S., & Wingate, D. (2005). Predictive linear-Gaussian models of stochastic dynamical systems. *UAI 21*.

Singh, S., James, M. R., & Rudary, M. (2004). Predictive state representations: A new theory for modeling dynamical systems. *UAI 20*.

Welch, G., & Bishop, G. (2004). *An introduction to the Kalman filter* (Technical Report TR 95-041). U. of N. Carolina at Chapel Hill, Dept. of Computer Science.

Wolfe, B., James, M. R., & Singh, S. (2005). Learning predictive state representations in dynamical systems without reset. *ICML 22*.

A. Appendices

A.1. State Update

Recall that $\mu_t = \mathrm{E}[Z_t|h_t, u_{t+1} = \mathbf{0}, u_{t+2} = \mathbf{0}, \ldots]$ and $\Sigma_t = \mathrm{Var}[Z_t|h_t, u_{t+1} = \mathbf{0}, u_{t+2} = \mathbf{0}, \ldots]$. To update the state, we must account for two new pieces of information: the selection of an action and a new observation. In order to compute the full update, we will first compute the distribution of Z_{t+1} given the new action $u_{t+1} = u$.

Because the actions have a linear effect on the observations, we can write the difference in expectation given $u_{t+1} = u$ as opposed to $u_{t+1} = \mathbf{0}$ as a linear function of u:

$$L_i u = \mathrm{E}[Y_{t+i+1}|h_t, u_{t+1} = u, u_{t+2} = \mathbf{0}, \ldots] - \mathrm{E}[Y_{t+i+1}|u_{t+1} = \mathbf{0}, u_{t+2} = \mathbf{0}, \ldots],$$

where L_i is a $1 \times l$ row vector. We can write a recursive definition for L_i:

$$L_i = \Gamma'_{n-i} + \sum_{j=1}^{i-1} g_{n-i+j+1} L_j. \tag{4}$$

Since $Z_{t+1} = GZ_t + \eta_{t+n+1} e_n$, we can write its distribution as a function of the state variables μ_t and Σ_t:

$$Z_{t+1}|h_t, u_{t+1} = \mathbf{0}, \ldots \sim \mathcal{N}(G\mu_t,\ G\Sigma_t G' + F).$$

In order to update the conditional distribution to take the new action $u_{t+1} = u$ into account, the mean must be changed to $G\mu_t + Lu$, where L is the $n \times l$ matrix whose ith row is given by L_i. The action does not affect the variance. Now we can write the joint distribution of Y_{t+1} and Z_{t+1}:

$$\begin{pmatrix} Y_{t+1} \\ Z_{t+1} \end{pmatrix} \Bigg| h_t, u_{t+1} = u, u_{t+2} = \mathbf{0}, \ldots \sim \mathcal{N}\left(\begin{pmatrix} e_1'\mu_t \\ G\mu_t + Lu \end{pmatrix}, \begin{pmatrix} e_1'\Sigma_t e_1 & f_t' \\ f_t & G\Sigma_t G' + F \end{pmatrix} \right).$$

Since Y_{t+1} and Z_{t+1} are jointly Gaussian, we can apply a standard result (e.g., Catlin, 1989) to compute the distribution conditioned on observing $Y_{t+1} = y_{t+1}$:

$$Z_{t+1}|h_t, Y_{t+1} = y_{t+1}, u_{t+1} = u, u_{t+2} = \mathbf{0}, \ldots \sim \mathcal{N}(\mu_{t+1}, \Sigma_{t+1}),$$

where μ_{t+1} and Σ_{t+1} are defined as in (2) and (3). Thus, we have derived these state updates.

A.2. The LDS

An LDS models a system with two sequences of random variables; Y_t is the real-valued[5] observation at time t, and X_t is the (unobserved) n-vector of state at time t. The system is regulated by actions; the l-vector action u_t has an additive linear effect on state variable X_{t+1}.

The initial state X_1 is normally distributed:

$$X_1 \sim \mathcal{N}(\widehat{x}_1^-, P_1^-). \tag{5}$$

[5]The LDS formalism allows vector-valued Y_t as well, but we restrict our attention to scalar Y_t.

Each subsequent state vector is computed as a linear function of the previous state with additive Gaussian noise, plus a linear function of the action:

$$X_{t+1}|X_t = x_t, u_t \sim \mathcal{N}(Ax_t + Bu_t, Q). \tag{6}$$

Finally, the observation is a noisy linear function of X_t:

$$Y_t|X_t = x_t \sim \mathcal{N}(Hx_t, R). \tag{7}$$

Because X_t is unobservable, an LDS is tracked by the Kalman filter, which maintains a state estimate $\widehat{x}_t^-$ and a covariance matrix P_t^-. The semantics of these variables is that $X_t|h_{t-1} \sim \mathcal{N}(\widehat{x}_t^-, P_t^-)$; consult Welch and Bishop (2004) for details.

A.3. Proof of Theorem 1

We prove that every n-dimensional controlled LDS has an equivalent n-dimensional cPLG by constructing a cPLG given an n-dimensional LDS with parameters A, B, H, Q, R, $\widehat{x}_1^-$, and P_1^-. Because of the additive linear effect of the actions and the fact that μ_t is conditioned on future actions being $\mathbf{0}$, the parameters shared with the original PLG remain unchanged:

$$\mu_t = M\widehat{x}_{t+1}^-, \quad \Sigma_t = MP_{t+1}^-M' + \Psi + RI,$$
$$C = \Psi_{n+1} - \Psi g - Rg,$$
$$\sigma^2 = HS_nH' + R - g'\Psi_{n+1} - g'C,$$

and g is any solution to $g'M = HA^n$, where

$$M = \begin{pmatrix} H \\ HA \\ \vdots \\ HA^{n-1} \end{pmatrix}, \tag{8}$$

$$\Psi_{ij} = \begin{cases} HA^{i-j}S_{j-1}H' & 1 \le j < i \\ HA^{j-i}S_{i-1}H' & i \le j \le n \end{cases},$$
$$S_i = \sum_{k=1}^{i} A^{k-1}Q(A^{k-1})',$$

and $\Psi_{n+1}^i = HA^{n+1-i}S_{i-1}H'$, where Ψ_{n+1}^i is the ith element of the n-vector Ψ_{n+1} (Rudary et al., 2005).

The only parameter yet to find is Γ (or, equivalently, L). We can compute L as follows (suppressing some of the conditions for readability and space):

$$\begin{aligned} L_1u &= \mathrm{E}[Y_{t+1}|u_{t+1} = u] - \mathrm{E}[Y_{t+1}|u_{t+1} = \mathbf{0}] \\ &= HBu \end{aligned}$$

Thus, $L_1 = HB$. Similarly, we obtain $L_2 = HAB$. Combining these with similar results for $L_3, L_4, \ldots$, we obtain $L = MB$. To compute Γ_i, solve (4).

Using these parameters, we obtain identical distributions over observations from the original LDS model and the new cPLG model, thus proving the result.

A.4. Proof of Theorem 3

We wish to show that any full-rank, n-dimensional LQG has an equivalent n-dimensional PLQG. By Theorem 1, there is a cPLG equivalent to the LDS; thus we need only show that an equivalent cost function exists, which we show by construction.

Since M is invertible, the following identities obtain: $\widehat{x}_t^- = M^{-1}\mu_{t-1}$, $B = M^{-1}L$, and $M^{-1}G = AM^{-1}$. The cost-function parameters of the PLQG are derived from the LQG's cost function parameters as follows: $W_{\mu,f} = M'^{-1}W_{x,f}M^{-1}$, $W_\mu = M'^{-1}W_xM^{-1}$, and $W_{\mu,u} = W_{x,u}M^{-1}$ (W_u is the same in the PLQG as in the LQG). The cost-function structure of LQGs is similar to PLQGs; we write the optimal expected cost-to-go at time t given h_{t-1} as $J_t^x(h_{t-1}) = \widehat{x}_t^{-\prime}V_t^x\widehat{x}_t^- + v_t^x$ and $V_{T+1}^\mu = W_{x,f}$. Suppose that $V_{t+1}^\mu = M'^{-1}V_{t+1}^xM^{-1}$. Then

$$\begin{aligned} V_t^\mu = {} & M'^{-1}W_xM^{-1} + GM'^{-1}V_{t+1}^xM^{-1}G - \\ & (M'^{-1}W_{x,u}' + G'M'^{-1}V_{t+1}^xM^{-1}L) \times \\ & (W_u + L'M'^{-1}V_{t+1}^xM^{-1}L)^{-1} \times \\ & (W_{x,u}M^{-1} + L'M'^{-1}V_{t+1}^xM^{-1}G) \\ = {} & M'^{-1}[W_x + A'V_{t+1}^xA - \\ & (W_{x,u}' + A'V_{t+1}^xB)(W_u + B'V_{t+1}^xB)^{-1} \times \\ & (W_{x,u} + B'V_{t+1}^xA)]M^{-1}, \end{aligned}$$

which means $\mu_{t-1}'V_t^\mu\mu_{t-1} = \widehat{x}_t^{-\prime}V_t^x\widehat{x}_t^-$, so the expected optimal cost-to-go differs by a history-independent constant. It follows that both models select the same action.

A.5. Proof of Theorem 4

The proof of CE's consistency depends on the following result from statistics: If $\widehat{x}_n \xrightarrow{p} x$ as $n \to \infty$, and $f : \mathcal{R}^k \to \mathcal{R}^m$ is continuous at x, then $f(\widehat{x}_n) \xrightarrow{p} f(x)$ as $n \to \infty$ (e.g., p. 234 of DeGroot & Schervish, 2002).

It follows from the weak law of large numbers that $\overline{\eta}_t \xrightarrow{p} 0$ for all t. From this it follows that $\Xi\gamma \xrightarrow{p} \Upsilon$. Rudary et al. (2005) showed that this can be rewritten as $\widehat{\gamma} = (\Xi'\Xi)^{-1}\Xi'\Upsilon \xrightarrow{p} \gamma$ when $(\Xi'\Xi)^{-1}$ is defined in the limit.

Since $\widehat{\gamma}$ is consistent, $\widehat{L}$ is as well (from the result stated above). From this and the fact that $\overline{y}_t \xrightarrow{p} \mathrm{E}[Y_t|u_1 = \overline{u}_1, \ldots]$ we get $\widehat{\mu}_0 \xrightarrow{p} \mu_0$. Likewise, $\widehat{\mathrm{E}Y_t^k} \xrightarrow{p} \mathrm{E}[Y_t|u_1 = u_1^k, \ldots]$, so $\widehat{\Sigma}_0$ is just a sample covariance, which is a well-known consistent estimator. Similarly, $\widehat{\sigma}^2$ and $\widehat{C}$ are a sample variance and sample covariance, respectively, and thus are consistent estimators.