A Chinese restaurant process extension for mixtures of predictive linear gaussians

Anonymous Author(s)
Affiliation
Address
email

Abstract

The Predictive Linear Gaussian (PLG) represents Linear Dynamical Systems by updating the statistics of future observations. In this paper we study a Bayesian extension of Mixtures of Predictive Linear Gaussians (MPLG), which are an extension of PLGs to non-linear systems. We use the Chines Restaurant Process to jointly discover linear models and the regions where they are activated in in state space. The learned models are then non-linearly mixed to obtain a non-linear filter. The Bayesian approach to learning MPLG parameters allows us to tackle the model order selection problem. The proposed model can be seen as a Bayesian observable version of the unscented Kalman filter.

1 Introduction

Maintaining reliable state estimates is an important topic in the control and prediction of dynamical systems. Classical approaches such as the Kalman filter typically use hidden states in which the assumption is made that observations are emissions from these underlying hidden states. Because the real state is never observed, learning state transitions is generally a hard problem and is often done using expectation-maximization methods or subspace identification[9].

However, observable representations for many of these approaches exist: e.g. OOMs[4] and PSRs[6] for HMMs and POMDPs in the discrete case and the PLG[8] for the Kalman filter in the continuous case. It is on the PLG that we focus in this paper and more precisely on Mixtures of PLGs, a non-linear extension of the PLG.

We show that the MPLG lends itself to Bayesian learning methods, which allows us to overcome the model order selection problem and increases the robustness of the learning algorithm. We develop an extension of the MPLG based on Dirichlet process mixtures and Bayesian linear regression. More precisely, we cluster linear models together with points in state space. This approach allows us to learn the linear models that are necessary to explain the data and the regions where each linear model should be activated.

As the MPLG is very similar to the unscented Kalman filter[5], our extension of the MPLG can be seen as a Bayesian version of the unscented Kalman filter. In this work we limit ourselves to uncontrolled dynamical systems.

We begin this paper with an overview of the PLG and MPLG and point out their limitations. Next, we introduce our MPLG training process method on the Chinese Restaurant Process. To show performance of the algorithm we then compare our algorithm with autoregressive models, the standard PLG (which is equivalent to the Kalman filter) and the standard MPLG in the experimental section. Finally we discuss possible extensions in the future work section.

2 Methods

2.1 Predictive Linear Gaussians

The Predictive Linear Gaussian represents Linear Dynamical Systems (LDS) by maintaining a sufficient set of statistics over future observations. In the case of the PLG, the sufficient statistics are simply the mean and covariance matrix of a multivariate Gaussian (or its natural parameters) and the update function is linear. Any n^{th} order LDS can be represented by a PLG with a horizon of n steps.

Formally the state of a PLG at time t is defined as a multivariate Gaussian conditioned on the history up till time t:

$$p(F_t|h_t) = p(O_{t+1}O_{t+2}\dots O_{t+n}|o_0o_1\dots o_t) = \mathcal{N}(\mu_t, \Sigma_t) \tag{1}$$

Here O_{t+i} is the random variable describing the distribution of the observation(s) at time $t+i$. o_i is/are the observed value(s) at time i.

When the PLG is used as a filter, we need to maintain state by advancing the predictions in time and incorporating new observations. This is done in two phases, first the current state distribution $p(F_t|h_t)$ is extended to the joint distribution of F_t and one additional future step O_{t+n+1}. As the PLG is a linear model, O_{t+n+1} is a linear function of F_t:

$$O_{t+n+1} = f(F_t, \eta_{t+n+1}) = g^T F_t + b + \eta_{t+n+1} \tag{2}$$

The noise term η_{t+n+1} is additive Gaussian white noise, but this term is allowed to covary with the n future observations (i.e. the current state):

$$\eta_{t+n+1} \sim \mathcal{N}(0, \sigma_\eta^2) \tag{3}$$

$$\text{Cov}[F_t, \eta_{t+n+1}] = C_\eta \tag{4}$$

The extended distribution takes again the form of a $pn+p$-dimensional Gaussian[1], with p the number of dimensions of the observations:

$$p(F_tO_{t+n+1}|h_t) = \mathcal{N}\left[\begin{pmatrix} F_t \\ E_t \end{pmatrix}, \begin{pmatrix} \Sigma_t\, C_t \\ C_t^T V_t \end{pmatrix}\right] \tag{5}$$

$$E_t = g^T\mu_t + b \tag{6}$$

$$C_t = \text{Cov}[O_{t+n+1}, F_t] = \Sigma_t g + C_\eta \tag{7}$$

$$V_t = \text{Var}[O_{t+n+1}] = g^T\Sigma_t g + g^T C_\eta + C_\eta^T g + \sigma_\eta^2 \tag{8}$$

To complete the state update (i.e. to obtain Σ_{t+1} and μ_{t+1}, we need to condition on the next observation o_{t+1}.This can be done with standard formulas for Gaussians. The PLG can also be written in information form, yielding similar update functions.

Many interesting systems are non-linear and hence the PLG is insufficient to model these systems. Similar to the extended/unscented Kalman filter, non-linear extensions of the PLG exist to overcome the limitations of their linear counterparts.

In this paper, we focus on Mixtures of Predictive Linear Gaussians (MPLG). The state of the MPLG is still a multivariate Gaussian, but the update function becomes non-linear. The MPLG learns linear models for spatially adjacent points and combines the predictions of these models non-linearly.

To define distance, kernels are used and each linear model is assigned to one kernel, i.e. if we have J models then the MPLG consists of J sets of weights g_j, biases b_j and noise models C_η^j, σ_j^2 Predictions are made by combining these linear models. Equation 2 is replaced with:

$$O_{t+n+1} = \sum_{j=1}^{J} w(F_t)_j(g_j^T F_t + b_j + \eta_{t+n+1}^j) \tag{9}$$

[1]Here we use the naive extension to multiple dimensions. It is not always required to track n future steps of all variables.

The weights $w(F_t)_j$ are computed by re-normalizing the kernels (ξ_k are the kernel centers):

$$w(F_t)_j = \frac{K(\xi_j, F_t; \phi_j)}{\sum_{k=1}^{J} K(\xi_k, F_t; \phi_k)} \tag{10}$$

Note that the state update is now non-linear as the weights $w(F_t)$ are applied to the distribution F_t and that there is a noise term for each PLG. As the state F_t of the (M)PLG is a multivariate Gaussian distribution (instead of a point estimate), we have to propagate this distribution through the non-linear update function and estimate the posterior statistics. As in the unscented Kalman filter, sigma points provide an efficient alternative to sampling.

Wingate et al. developed Hybrid Particle-Analytical Inference to estimate the statistics of the extended distribution $F_t O_{t+n+1}$ for the MPLG. If the observations are p-dimensional, then we construct $2pn + 1$ sigma points:

$$f_t^{(2i)} = \mu_t + (\sqrt{pn\Sigma_t})_i \tag{11}$$

$$f_t^{(2i+1)} = \mu_t - (\sqrt{pn\Sigma_t})_i \tag{12}$$

$$f_t^0 = \mu_t \tag{13}$$

For each of these sigma points, the weights $w(f^{(i)})_j$ are computed. Next the parameters E_t, C_t, V_t are estimated by computing expectations over these sigma points:

$$H_t^i = \sum_{j=0}^{J} w(f^{(i)})_j (g_j^T f^{(i)} + b_j) \tag{14}$$

$$Q_t^i = \sum_{j=0}^{J} w(f^{(i)})_j {C_\eta^j}^T \Sigma_t^{-1} (f^{(i)} - \mu_t) \tag{15}$$

$$D_t^i = \sum_{j=0}^{J} w(f^{(i)})_j (\sigma_j^2 - {C_\eta^j}^T \Sigma_t^{-1} C_\eta^j) \tag{16}$$

$$E_t = \frac{1}{2pn+1} \sum_{i=0}^{2pn+1} H_t^i + Q_t^i \tag{17}$$

$$C_t = \frac{1}{2pn+1} \sum_{i=0}^{2pn+1} (H_t^i + Q_t^i) {f^{(i)}}^T - E_t \mu_t^T \tag{18}$$

$$V_t = \frac{1}{2pn+1} \sum_{i=0}^{2pn+1} H_t^i {H_t^i}^T + Q_t^i {H_t^i}^T + H_t^i {Q_t^i}^T + Q_t^i {Q_t^i}^T + D_t^i - E_t E_t^T \tag{19}$$

Here $w(f^{(i)})_j$ are the mixing weights/probabilities of PLG j being the appropriate model for the point $f^{(i)}$. The algorithm is completed as in the standard PLG case, by creating the extended distribution $F_t O_{t+n+1}$ and subsequent conditioning or marginalizing.

One particular questions that remains is how to define the kernels. Normally one uses the Kernel Recursive Least Squares algorithm [2] to learn the kernel centers. In this case the algorithm tries to satisfy the approximate linear dependence condition (ALD) and hence creates a sparse solution. Points are added to a dictionary of kernel centers to make sure this condition is satisfied, given the level of sparsity required. These kernel centers are then used by the MPLG algorithm, by centering each PLG around one of the kernel centers. In the normal MPLG, the linear models are learned by weighted linear regression. Let $f_t = (o_t o_{t+1} \dots o_{t+n} 1)^T$, $\tilde{F} = (f_0 f_1 \dots)$ and $\tilde{W}_j = \mathrm{diag}(w(f_o) w(f_1) \dots)$, where the o_i are taken from a training set, then:

$$\begin{pmatrix} g_j \\ b_j \end{pmatrix} = (\tilde{F} \tilde{W}_j \tilde{F}^T)^{-1} \tilde{F} \tilde{W}_j (o_n o_{n+1} \dots)^T \tag{20}$$

Another extension of the standard PLG are Exponential Family Predictive Representations of State (EFPSR). In this case the assumption that the state is a multivariate Gaussian is dropped and replaced with a general exponential family distribution. As before, the sufficient statistics of this

distribution constitute state and the goal of the EFPSR is to maintain state by extending and conditioning/marginalizing this distribution. As shown by Wingate et al. the PLG and MPLG can be cast as a special case of the EFPSR. In this paper we do not consider the EFPSR and leave the extension to exponential family distributions as future work.

2.2 The Chinese Restaurant Process Applied To the PLG

2.2.1 Chinese Restaurant Process

Dirichlet Process Mixtures (DPM) [1] are widely used in nonparametric clustering. A clean way to describe the DPM is via the Chinese Restaurant Process (CRP). A CRP has two parameters: α the concentration parameter and G_0 the base distribution. One can generate samples from a CRP mixture model by executing the following steps. Let there be J clusters, n_j points in cluster j, the cluster index for point x_i is c_i, the parameter θ_j is assigned to cluster j. First draw a cluster assignment c_i based on $\mathbf{c}_{-i} = c_1, \ldots, c_{i-1}$:

$$p(c_i = j|\mathbf{c}_{-i}) \propto \begin{cases} n_j & j \leq J \\ \alpha & j = J+1 \end{cases},$$

When $j = J+1$ then draw a new θ_{J+1} from G_0. The final step is to draw x_i from $H(\theta_j)$. This generative model can also be described as a DPM:

$$\begin{aligned} G &\sim DP(\alpha, G_0), \\ \theta_i|G &\sim G, \\ x_i|\theta_i &\sim H(\theta_i). \end{aligned}$$

This model can be used to cluster data, this is clustering is made explicit in the CRP setting by the indexes c_i. The distribution on cluster assignments given the data is not analytically tractable, so we have to resort to other techniques like Markov Chain Monte Carlo (MCMC) or variational approximations. In this paper we have employed Gibbs Algorithm 3 from [7]. It is important to mention that this sampler can only be used when the base distribution G_0 is conjugate to H. The Gibbs sampler works directly on cluster assignments, the latent variables θ_i are integrated out. Each sampling step uses the following equations to select a sample c_i:

$$p(c_i = j|\mathbf{c}_{-i}, G_0, \alpha, D) \propto \begin{cases} n_j \int_\theta p(x_i|\theta) p(\theta|G_0, \alpha, D_j) & j \leq J \\ \alpha \int_\theta p(x_i|\theta) p(\theta|G_0, \alpha) & j = J+1 \end{cases},$$

where c_i is the cluster assignation for x_i, c_{-i} are the cluster assignations for all points except x_i, G_0 is the base measure, α is the concentration parameter, D is the entire data set, D_j is the data assigned to cluster j, n_j is the number of points already assigned to cluster j and J is the number of existing (non-empty) clusters. Now will discuss how this sampler can be used to learn a MPLG.

2.2.2 How to learn the MPLG using the CRP

Our first goal is to do model order selection for the MPLG: how many different models do we need to model our data? The second goal is: what are the underlying linear models that can predict the next state. Both of these goals can be reached by doing inference using the following DPM:

$$\begin{aligned} H &\sim DP(\alpha, G_0), \\ \theta|H &\sim H, \\ f_t|\mu_f, \Sigma_f &\sim \mathcal{N}(\mu_f, \Sigma_f), \\ o_t|f_t, g, \sigma_o &\sim \mathcal{N}(g^T f_t, \sigma_o^2), \end{aligned}$$

where g is the linear trend in a PLG, f_t is a sequence of n observations: $f_t = [o_{t+1}, \ldots, \emptyset_{t+n}]^T$, and o_{t+n+1} is the observation following f_t. The variable θ from the general DPM above contains the following components: $\mu_f, \Sigma_f, \sigma_o, g$. To construct this DP, we have used the assumption from Equation: 2. The underlying DPM that we use, is the same one as the Gaussian example in the Dirichlet Process-Generalized Linear Model (DP-GLM)[3]. We used the following factorized base distribution G_0: σ_o is constant, $g \sim \mathcal{N}(0, \sigma_g)$ and either $\mu_f, \Sigma_f \sim GaussianWishart(\mu_{f_0}, \Delta, \nu, \kappa)$ or

$\mu_f \sim N(\mu_{f_0}, \Sigma_{f_0})$ and Σ_f constant. Because this base distribution is conjugate to the distribution on θ, Gibbs algorithm 3 is applicable.

It is easy to see how we can employ this model determine the number of PLG's. It is simply the number of different $J+1$ where J is the number of non-empty clusters at a specific Gibbs iteration. The extra cluster is the cluster which contains no data, but could contain a previously unseen data point, it is a type of default cluster. The underlying linear models can also be extracted from the cluster assignment. Let f_t be a specific observed data point. We can compute the expected value of o_{t+n+1}:

$$\begin{aligned} E[o_{t+n+1}|\mathbf{c}, D, f_t] &= \frac{\alpha}{b} p(f_t|c_{new} = J+1) E[o_{t+n+1}|c_{new} = J+1, D, f_t] \\ &\quad + \sum_{j=1}^{J} \frac{n_j}{b} p(f_t|c_{new} = j, \mathbf{c}, D) E[o_{t+n+1}|c_{new} = j, \mathbf{c}, D, f_t], \\ b &= \alpha p(f_t|c_{new} = J+1) + \sum_{k=1}^{J} n_j p(f_t|c_{new} = j, \mathbf{c}, D), \end{aligned}$$

where c_{new} is the cluster that (f_t, o_{t+n+1}) is assigned to. The likelihood $p(f_t|c_{new} = j, \mathbf{c}, D)$ can be computed as follows:

$$p(f_t|c_{new} = j, \mathbf{c}, D) = \int_{\mu_t, \sigma_t} p(f_t|\mu_t, \Sigma_t) p(\mu_t, \Sigma_t|D_j).$$

The expectation $E[o_{t+n+1}|c_{new} = J, \mathbf{c}, D, f_t]$ is always analytically tractable given our conjugate base distribution. Let g_j be the most likely value of the posterior on g given the data in cluster j i.e. the mean of the posterior. When we insert this into the expectation than

$$\begin{aligned} E[o_{t+n+1}|c_{new} = J+1, D, f_t] &= 0, \\ E[o_{t+n+1}|c_{new} = j, \mathbf{c}, D, f_t] &= g_j^T f_t. \end{aligned}$$

This result indicates that we can use the g_j directly as linear trends in the MPLG.

Up till now, we have used the same assumptions as the standard MPLG model to learn the linear models and just as in the MPLG. This means that we have ignored the covariance of the noise with the predictions We can plug the different g_j directly into the MPLG model. Furthermore, the mixing weights used in equation 10 can be computed as a ratio of posterior likelihoods instead of kernels:

$$w(f_t)_j = \frac{n_j p(f_t|c_{new} = j, \mathbf{c}, D)}{p(f_t|c_{new} = J+1) + \sum_{k=1}^{J} n_k p(f_t|c_{new} = k, \mathbf{c}, D)}.$$

Given these newly defined k̈ernelsänd linear trends, we can learn the noise covariance just as in the normal MPLG model and this completes the MPLG training. The prediction/filtering does not differ from the standard MPLG.

There is one final remark that has to be made. Because we are using Gibbs sampling, we have to average the predictions out over different samples. In the ideal case this could be done in every extend and condition step. The problem is that this is not possible because the number of PLG's can vary and that this makes it also impossible to learn the noise model per PLG. Instead, we have used different Gibbs sampling iterations and trained an entire MPLG model for each iteration. The predictions are then averaged out over the different models.

3 Experiments

In this section we consider different kinds of experiments to show the modeling power of our Bayesian approach to the PLG. We first apply our model to toy data sets to show how our model works. Next we consider non-linear oscillators and show that our technique can learn these patterns and their structure.

3.1 Setup

Each experiment consists of three phases: the model learning phase, the noise learning phase and the testing phase. During the learning phase, linear models are discovered by the Gibbs sampler.

Next, we estimate the noise parameters by running the algorithm on the train set with all C_η^j set to zero and recording the means of the predictions μ_t as in [10].

Finally during the testing phase, the algorithm is run in filter or free-run mode. In filter mode, a new observation o_{t+1} becomes available and we condition on this observation. This mode is useful for one-step predictions. In free-run mode, we first run the algorithm as in the filter mode for a certain number of steps. Next we marginalize out O_{t+n+1} instead of conditioning on this variable, so we maintain the distribution F_t instead of $F_t|h_t$. This way the usefulness of the algorithm for long-term predictions can be assessed.

We compare our results to the standard PLG model and to Kernel Auto Regression (KAR).

3.2 Toy data

The first dataset that we considered is the biped dataset [10]. It is a 2 dimensional linear system, performing a rotation. There is noise on both the rotation dynamics as the observation. The value of the second dimension cannot go below -0.5 (without observation noise). The dataset and the clustering are shown in Figure: **??**. The length of the history was 1. This figure shows the structure that the CRP has found in the data. One part of the dataset consists of a rotation, the other part is a translation. The results are shown in Table: 1. We can see that the best MSE in filter mode is achieved by KAR, but the CRP extended MPLG is a very close second and the PLG trails behind. The mean result for the MPLG indicates that the MPLG has learned the structure of the underlying system much better than the standard PLG. These results are confirmed when we take a look at the freerun experiments. The CRP-MPLG outperforms the other techniques with an MSE only slightly higher than in filter mode. The KAR method achieves very bad MSE values in filter mode, indicating that these methods have less predictive power.

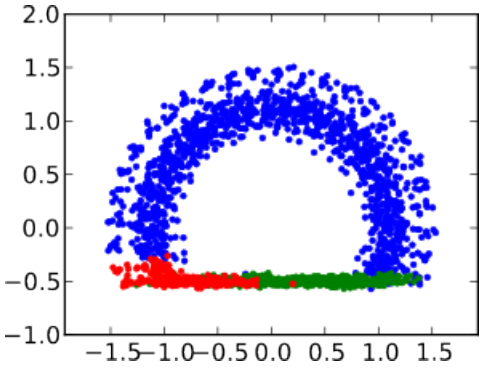

Figure 1: Representative clustering on the biped dataset. The clustering shows where in state space a specific cluster becomes active.

4 Future work

We see a number of possible extensions for this line of work. One of the goals of our model is to learn the underlying structure of dynamical systems. One interesting research path would be to incorporate temporal information in this structure. Another one would be to use a hierarchical setup (the Hierarchical Dirichlet Process) to jointly learn different systems with similar dynamics.

One drawback of the current model is that a specific linear model is tied to a single cluster in state space. We are working on an extensions of this model that enables us to share the same linear models across different clusters in state space. This gives us a hierarchical structure with the linear trends on top, where each linear trend can be activated by different kernels in state space.

Table 1: Results on biped

METHOD	MEAN MSE	BEST MSE
KAR-FILTER	0.078	0.0021
PLG-FILTER	0.0056	0.0035
CRP-MPLG-FILTER	0.0033	0.0022
KAR-FREE	0.6079	0.0326
PLG-FREE	0.0059	0.0035
CRP-MPLG-FREE	0.0036	0.0021

Our current model is based on Bayesian linear regression, which maps almost directly to the (M)PLG. The EFPSR maintains state by updating the parameters of a general exponential family distribution.

5 Conclusion

We began with this paper with a review of Predictive Linear Gaussians and one of their non-linear extensions: Mixtures of Predictive Linear Gaussians. We explained that MPLGs are typically trained with a frequentist approach based on linear regression. Using the Chinese restaurant process and Bayesian linear regression we showed that the MPLG lends itself to Bayesian learning methods. Our main result is not quantitative in the sense that our extension of the MPLG is guaranteed to have better performance than the standard MPLG, but that we developed a robust Bayesian learning algorithm for the MPLG that allows us to tackle the model order selection problem and that provides a solid foundation towards structure learning and transfer learning. In a filter setting this algorithm can be seen as an observable Bayesian version of the unscented Kalman filter.

References

[1] C E Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics*, 2(6):1152–1174, 1974.

[2] Y Engel, S Mannor, and R Meir. The Kernel Recursive Least-Squares Algorithm. *IEEE Transactions on Signal Processing*, 52(8):2275–2285, 2004.

[3] Lauren A Hannah, David M Blei, and Warren B Powell. Dirichlet Process Mixtures of Generalized Linear Models. *Most*, 92(438):1–37, 2009.

[4] H Jaeger. Observable operator models for discrete stochastic time series. *Neural Computation*, 12(6):1371–1398, 2000.

[5] Simon J Julier and Jeffrey K Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Int Symp AerospaceDefense Sensing Simul and Controls*, volume 3, pages 182–193. Spie, 1997.

[6] Michael L Littman, Richard S Sutton, and Satinder P Singh. Predictive Representations of State. In Thomas G Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Neural Information Processing Systems NIPS*, number 14, pages 1555–1561. MIT Press, 2002.

[7] Radford M Neal. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal Of Computational And Graphical Statistics*, 9(2):249, 2000.

[8] Matthew Rudary and David Wingate. Predictive Linear-Gaussian Models of Stochastic Dynamical Systems. *Engineering*, pages 501–508, 2005.

[9] P Van Overschee and B De Moor. *Subspace identification for linear systems: theory, implementation, applications*, volume 2008. Kluwer Academic Publishers, 1996.

[10] David Wingate and Satinder Singh. Mixtures of Predictive Linear Gaussian Models for Nonlinear Stochastic Dynamical Systems. In *National Conference on Artificial Intelligence - AAAI*, 2006.