Gaussian Process Latent Variable Models for Dimensionality Reduction and Time Series Modeling

Nakul Gopalan
IAS, TU Darmstadt
nakul.gopalan@stud.tu-darmstadt.de

Abstract

Time series data of high dimensions are frequently encountered in fields like robotics, computer vision, economics and motion capture. In this survey paper we look first at Gaussian Process Latent Variable Model (GPLVM) which is a probabilistic nonlinear dimensionality reduction method. Further we discuss Gaussian Process Dynamical Model (GPDMs) which are based GPLVM. GPDM is a probabilistic approach to model high dimensional time series data in a low dimensional latent space with a dynamical model. We also discuss variational approximations of GPLVM and Variational Gaussian Process Dynamical System (VGPDS) which is a dynamical model based on these variational approximations.

1 Introduction

Probabilistic modeling of high dimensional time series data like human motion is an interesting machine learning problem. Applications for this form of probabilistic modeling have also been evaluated in the areas of speech synthesis [11] and modeling high dimensional video sequences [5]. In this survey paper we will discuss the topic of Gaussian process dynamical models (GPDMs) [1] with respect to human motion modeling, as most research in GPDMs is spurred by the need to model human motion for the purposes of tracking or activity recognition or just animating more human like characters.

Approaches before the GPDMs dealt with learning probability distributions over the space of poses and motions to learn motions which are similar to those in the training data. This task is challenging as the human motion is a high dimensional data and the motions are complex. Observations indicated that the poses of similar activities lie near a lower dimension nonlinear manifold. This observation led to the decoupling of motion and pose model in lower dimensions – motion is modeled by a dynamical process defined on the lower dimensional latent space and poses are generated by an observation process from the latent space to the high dimensional space [1].

Modeling of the time series data is achieved using dynamical systems. These can range from the simple hidden Markov models to the more expressive nonlinear dynamical systems like in [3] which use nonlinear basis functions. GPDMs are based on nonlinear methods as they use GPLVMs for dimensionality reduction. GPDMs use Gaussian Processes (GP) for regression in the latent space [1]. A Gaussian Process in [6] is defined as "a probability distribution over functions $y(x)$ such that the set of values of $y(x)$ evaluated at an arbitrary set of points $x_1 \dots x_{\mathrm{N}}$ jointly have a Gaussian distribution". The point to notice here is that the joint distribution over N variables $y_1 \dots y_{\mathrm{N}}$ is defined by second order statistics of mean and variance [6]. Hence to train these models all we need are the mean and variance of $y(x)$ where mean can be assumed to be zero and the variance is calculated by,

$$\mathbb{E}[y(x_{\mathrm{i}}), y(x_{\mathrm{j}})] = k(x_{\mathrm{i}}, x_{\mathrm{j}}) \tag{1}$$

The kernel k determines the properties of the generated function. For a smooth output function we can use squared-exponential covariance function as a kernel for a more noisy of Brownian like out-

put we can use an Ornstein-Uhlenbeck kernel [7]. The GP is used in GPDMs for two purposes – one as a prior for the dynamical model and the other as a part of GPLVM's for dimensionality reduction. Dimensionality reduction involves finding a suitable low dimensional representation for a high dimensional data which represents the most important features of the high dimensional data set. We have linear methods and non linear methods depending on the nature of the latent subspace manifold and mapping to the subspace. GPDMs use Gaussian Process latent variable model (GPLVMs) [2] which gives a joint distribution of the latent subspace data and the observed (high dimensional) data. GPLVM is a probabilistic approach which optimizes the points in the latent space (variables) most likely to represent the observed data and also optimizes over the hyperparameters it uses to represent the observed data space with respect to the latent space. This optimization does not give very good results as it is done simultaneously over the latent space and the hyperparameters. Hence some variational approximations are done to improve this optimization in a series of papers [4] [5].

The next section will describe the GPDMs and GPLVM in detail. In Section 3 we will discuss the improvements brought upon by the variational methods and finally in Section 4 we would have some concluding remarks about GPDMs.

2 Gaussian Process Dynamical Models

Modeling of high dimensional time series data is split into two problems- one of dimensionality reduction, for projecting the data from the latent space to observation space and other of a dynamical model in the latent space. We will first discuss dimensionality reduction methods leading up to GPLVM and GPDM.

2.1 Different Methods for Dimensionality Reduction

Dimensionality reduction is used to provide a mapping between an observed space $Y \in \mathbb{R}^{N\times D}$ to a latent space $X \in \mathbb{R}^{N\times d}$. The simplest example of dimensionality reduction would be Principal Component Analysis (PCA). PCA is defined as "an orthogonal projection of data onto a lower dimensional linear space, known as the *principle subspace*, such that the variance of the projected data is maximized" [6]. There have been other non linear methods like kernel PCA and multidimensional scaling which use the proximity in the high dimensional data to model its latent projections. These methods use a kernel or similarity measure as proximity which can be nonlinear to obtain the projection [2]. GPLVM also uses kernels as similarity measure between the latent variables. Probabilistic PCA (PPCA) is a Bayesian formulation of the PCA method. PPCA is a method that projects latent space data into observed space [2]. All the dimensionality reduction methods discussed before PPCA projected observation space data to latent space. There are difficulties in projecting latent space data to high dimensional space, primary being that the solution can be one to many.

Let us assume we have a D dimensional data $Y = [y_1 \dots y_{\mathrm{N}}]$ at N points. The d dimensional latent space data points corresponding to these high dimensional data points can be $X = [x_1 \dots x_{\mathrm{N}}]$. Hence the relationship between the latent space data and the high dimensional data with Gaussian noise added would be

$$y_{\mathrm{i}} = W x_{\mathrm{i}} + \eta_{\mathrm{i}} \tag{2}$$

where mapping $W \in \mathbb{R}^{D\times d}$ and noise $\eta_{\mathrm{i}} \in \mathbb{R}^{D\times 1}$ [2]. The noise is an independent sample from a spherical Gaussian distribution

$$p(\eta_{\mathrm{i}}) = \mathcal{N}(\eta_{\mathrm{i}}|0, \beta^{-1}I) \tag{3}$$

where $\beta^{-1}I$ is the covariance and the noise mean is zero [2]. From [2] the likelihood for a data point can be written as

$$p(y_{\mathrm{i}}|x_{\mathrm{i}}, W, \beta) = \mathcal{N}(y_{\mathrm{i}}|W x_{\mathrm{i}}, \beta^{-1}I). \tag{4}$$

We integrate over the latent variables to get the marginal likelihood

$$p(y_{\mathrm{i}}|W, \beta) = \int p(y_{\mathrm{i}}|x_{\mathrm{i}}, W, \beta)p(x_{\mathrm{i}})\mathrm{d}x_{\mathrm{i}} \tag{5}$$

where $p(x_{\mathrm{i}}) = \mathcal{N}(x_{\mathrm{i}}|0, I)$ is the prior over the latent variables [2]. This gives the the marginal likelihood of each data points as

$$p(y_{\mathrm{i}}|W, \beta) = \mathcal{N}(y_{\mathrm{i}}|0, WW^{\mathrm{T}} + \beta^{-1}I) \tag{6}$$

using which the likelihood of the full data set can be given as

$$p(Y|W,\beta) = \prod_{\mathrm{i}=1}^{N} p(y_{\mathrm{i}}|W,\beta). \tag{7}$$

The parameters W are found by maximization of Eq. 7 [2]. This solution is achieved when the mapping W spans the principal sub-space of the data [2]. This model therefore has a probabilistic interpretation of PCA[2]. Marginalizing the latent variables and optimizing the parameters to increase the marginalized likelihood using maximum likelihood is the standard approach for fitting latent variable models like the PPCA [2]. The GPLVM is a dual to the PPCA and marginalizes and optimizes the other way around as will be described below.

2.2 Gaussian Process Latent Variable Model

GPLVMs are used to map latent space variables to observational space and it operates in the same direction as PPCA. However in the GPLVM framework the mapping parameters W are viewed as random variables with a prior of

$$p(W) = \prod_{\mathrm{i}=1}^{N} \mathcal{N}(w_{\mathrm{i}}|0,I), \tag{8}$$

which is a spherical Gaussian distribution with zero mean [2]. Furthermore instead of marginalizing out the latent variables as in PPCA we marginalize out the weight vectors from Eq. 4 to get

$$p(Y|X,\beta) = \prod_{\mathrm{d}=1}^{D} p(y_{:,\mathrm{d}}|X,\beta), \tag{9}$$

where $y_{:,\mathrm{d}}$ is the dth column/ dimension of Y [2] and has a joint probability of

$$p(y_{:,\mathrm{d}}|X,\beta) = \mathcal{N}(y_{:,\mathrm{d}}|0, XX^{\mathrm{T}} + \beta^{-1}I). \tag{10}$$

It can be seen from Eq. 10 that the marginal likelihood can be represented as a zero mean process with the covariance of $K = XX^{\mathrm{T}} + \beta^{-1}I$ [2]. Where each element of K, $k(x_{\mathrm{i}}, x_{\mathrm{j}})$ can be written as

$$k(x_{\mathrm{i}}, x_{\mathrm{j}}) = x_{\mathrm{i}}^{\mathrm{T}} x_{\mathrm{j}} + \beta^{-1}\delta_{\mathrm{ij}}, \tag{11}$$

this covariance function or kernel is similar to that of a Gaussian Process prior over a space of linear functions as described before. The marginal likelihood for dual probabilistic PCA is therefore a product of D independent Gaussian processes [2].

This means that the marginal likelihood of a latent variable representing a variable in a high dimensional space can be modeled using a GP. The next step would then be to train this GP so that it gives a mapping between the latent space and observational space. This is done by maximizing the marginal likelihood in Eq. 9. This can be done by substituting the individual marginal likelihoods in Eq. 9 from those in Eq. 10, taking a negative log and minimizing the log likelihood w.r.t. the latent variables and hyperparameters of the kernel. The negative log likelihood is given by the equation

$$L = \frac{N}{2}\ln(2\pi) + \frac{1}{2}\ln|K| + \frac{1}{2}\mathrm{tr}(K^{-1}S), \tag{12}$$

which needs to be minimized w.r.t. the latent variables X and hyperparameters of the kernel matrix K, where $S = D^{-1}YY^{\mathrm{T}}$. The hyperparamters are needed for learning the kernel which would give the required output function based on the training data. This kernel can be a "linear + RBF" kernel used in [1]

$$k(x_{\mathrm{i}}, x_{\mathrm{j}}) = \alpha_1 \exp(-\frac{\alpha_2}{2}\|x_{\mathrm{i}} - x_{\mathrm{j}}\|^2) + \alpha_3 x_{\mathrm{i}}^{\mathrm{T}} x_{\mathrm{j}} + \alpha_4^{-1}\delta_{\mathrm{x_i,x_j}}, \tag{13}$$

which has properties of smoothness and nonlinearity for the output function. The hyperparameters are the α parameters. Otherwise the kernel can be the automatic relevance determination (ARD) kernel used in [5]

$$k(x_{\mathrm{i}}, x_{\mathrm{j}}) = \sigma_{\mathrm{ard}}^2 \exp(-\frac{1}{2}\sum_{\mathrm{d}=1}^{\mathrm{D}} w_{\mathrm{d}}(x_{\mathrm{i,d}} - x_{\mathrm{j,d}})^2), \tag{14}$$

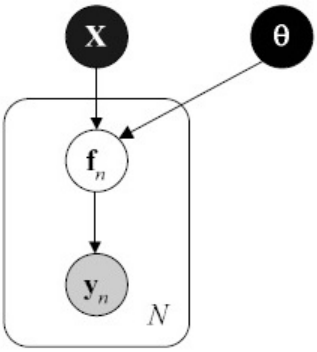

Figure 1: The Gaussian process as a latent variable model. Y is the observational space variable, X is the latent space variable and θ are the hyperparameters. Figure and caption from [2]

which allows training to switch off certain dimensions by reducing the parameter w_{d} to zero for unnecessary dimensions. In ARD kernel the hyperparameters are the w_{d} parameters.

Minimizing the Eq. 12 w.r.t. the hyperparamters of the kernel and latent variables is needed to maximize the likelihood. K is a nonlinear kernel w.r.t. X and hence the optimization has no closed form solution in case of GPLVMs. Gradient descent is used to find local minimas w.r.t. X and the hyperparameters of K, with initialization of X being done using PCA and the optimization using sparse methods to increase the speed of optimization. This is an intractable analytical optimization w.r.t. X and is solved in [4] where Bayesian approach is used to optimize the latent variables using a tight lower bound on the marginal likelihood. The graphical model of the GPLVM is shown in Fig.1 and the result of dimensionality reduction on the Oil Data Set explained in [6] with GPLVM's and PCA is given in Fig.2 . Notice the confidence bounds of the GPLVM latent space w.r.t. their mapping to an observational space point the more brighter a region the more confident the model is of the mapping at that point. Also quality of these visualizations were compared using nearest neighbour classification in the two dimensional latent space and GPLVM had better results than kernel PCA, Multidimensional scaling etc [2].

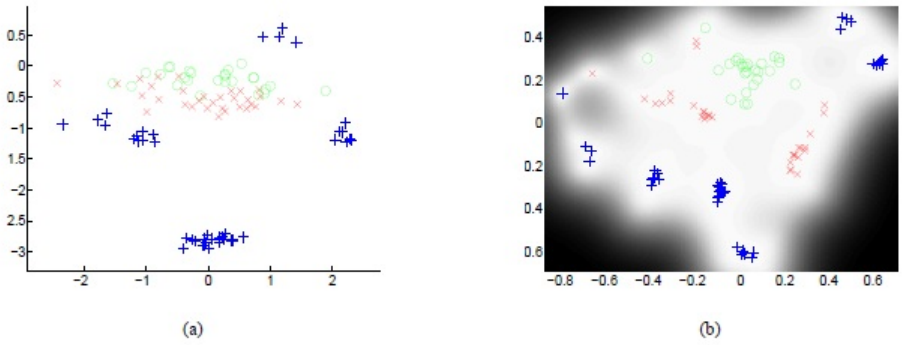

Figure 2: Visualization of the Oil data with (a) PCA (a linear GP-LVM) and (b) A GP-LVM which uses an RBF kernel- the more brighter a region the more confident the model is of the mapping at that point. Figure and caption from [2]

GPLVMs give us a probabilistic approach to model mapping between the points in observable space and in latent space. These models generalize well to new latent space points whose corresponding observational space points can be computed by GP regression since the model is GP based. The method for GP regression is simple and involves computing the new mean and covariance functions

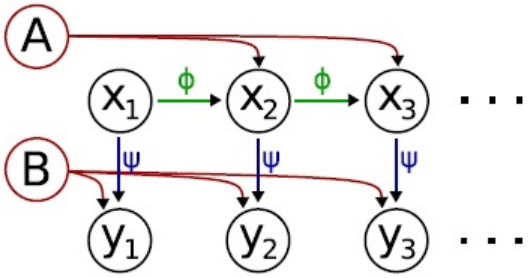

Figure 3: Time series graphical model - GPDM model Dynamical model in the latent space and projection to observational space. Figure from [12]

for new latent space point observed. More details can be found in [7]. Dynamical models are introduced in the next subsection.

2.3 Gaussian Process Dynamical Models

GPLVMs allow us model joint probabilities of observational and latent space data points. GPDMs allow us to model time series data using a dynamical model in the latent space based on a GP and also uses another GP to map the latent space data to observational space. This can be seen in Fig.3. The basic model can be written as

$$x_{\mathrm{t}} = f(x_{\mathrm{t}-1}, A) + n_{\mathrm{x,t}}, \tag{15}$$

$$y_{\mathrm{t}} = g(x_{\mathrm{t}}, B) + n_{\mathrm{y,t}}, \tag{16}$$

where $f(x;A) = \sum_{\mathrm{i}} a_{\mathrm{i}}\phi_{\mathrm{i}}(x)$ and $f(x;B) = \sum_{\mathrm{j}} b_{\mathrm{j}}\psi_{\mathrm{j}}(x)$ are linear combinations of nonlinear basis functions and $n_{\mathrm{y,t}}$ and $n_{\mathrm{x,t}}$ are zero-mean, isotropic, white Gaussian noise processes [1]. Since f and g are linear in terms of A and B, A and B can be assumed to have an isotropic Gaussian prior allowing them to be marginalized out [1]. This gives us the following marginal probability for the observation model

$$p(Y|X,\bar{\beta},W) = \frac{W^{\mathrm{N}}}{\sqrt{(2\pi)^{\mathrm{ND}}|K_{\mathrm{Y}}|^{\mathrm{D}}}} \exp(-\frac{1}{2}\mathrm{tr}(K_{\mathrm{Y}}^{-1}YW^2Y^{\mathrm{T}})), \tag{17}$$

where $\bar{\beta}$ are K_{Y} kernel matrix parameters between input latent variables. The kernel used is the RBF kernel, and W is the scale diagonal matrix to make sure each observational dimension is of the same length scale and exerts the same influence in optimization of the kernel parameters β [1]. The marginal probability for the dynamical model after marginalizing A is

$$p(X|\bar{\alpha}) = \frac{p(x_1)}{\sqrt{(2\pi)^{(\mathrm{N}-1)\mathrm{d}}|K_{\mathrm{X}}|^{\mathrm{d}}}} \exp(-\frac{1}{2}\mathrm{tr}(K_{\mathrm{X}}^{-1}X_{2:\mathrm{N}}X_{2:\mathrm{N}}^{\mathrm{T}})), \tag{18}$$

where $\bar{\alpha}$ are the kernel matrix hyperparameters between the previous i.e. $N-1$ input latent variables. The kernel matrix itself is "RBF + linear", hence has non linear input terms which can only be locally optimized [1]. The improvement to this optimization is dealt in [5]. Since the conditional distributions are in a GP formulation prediction to next instant of the observational model can be done by using the GPDM as a prior as like in a GP [1]. Learning the GPDM model is learning the optimum parameters $X, \bar{\beta}, \bar{\alpha}, W$ for observed training data Y. The training data used in [1] was the Carnegie Mellon University motion capture (CMU mocap) database. We would explain the Maximum a posteriori (MAP) estimation method and mention the other methods used in [1] for optimization of GPDMs.

MAP estimation for GPDM is performed to minimize joint negative log posterior of the hyperparameters given as $-\ln p(X, \bar{\alpha}, \bar{\beta}, W|Y)$, this is given as

$$L = L_X + L_Y + \sum_j \ln \beta_{\mathrm{j}} + \frac{1}{2\kappa^2}\mathrm{tr}(W^2) + \sum_j \ln \alpha_{\mathrm{j}}, \tag{19}$$

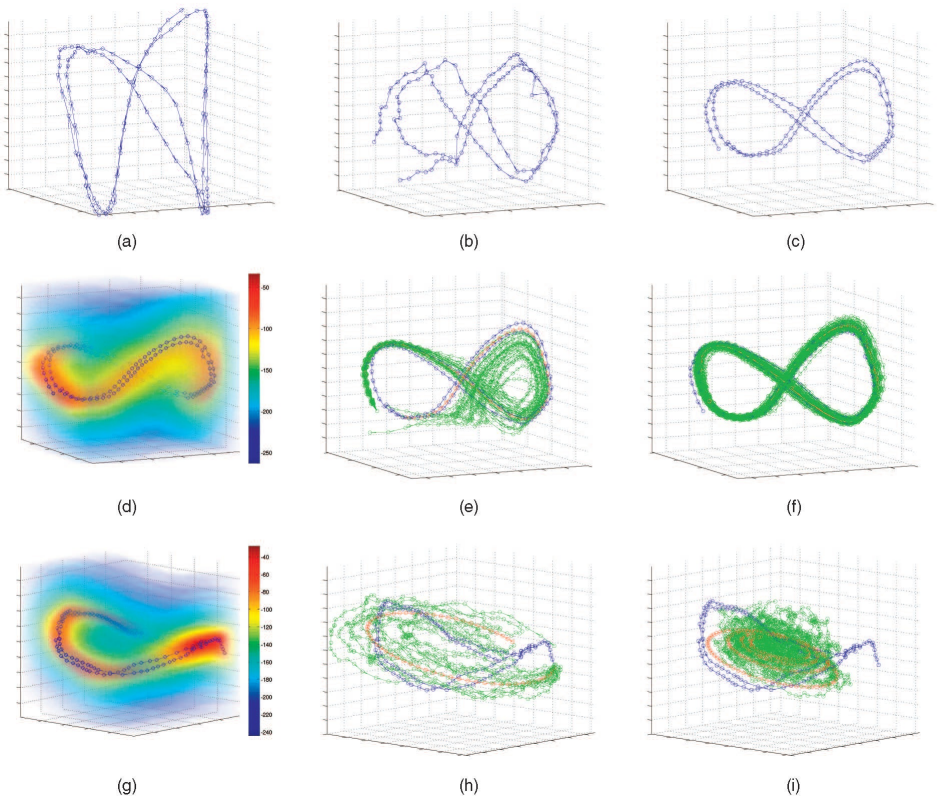

Figure 4: Models learned from a walking sequence comprising two gait cycles. (a) The PCA initializations and the latent coordinates learned with (b)GPLVM and (c) GPDM are shown in blue. Vectors depict the temporal sequence. (d) $-\ln$ variance for reconstruction shows positions in latent space that are reconstructed with high confidence. (e) Random trajectories drawn from the dynamic predictive distribution by using hybrid Monte Carlo (HMC) are green, whereas the red trajectory is the mean prediction sample. (f) Longer random trajectories drawn from the dynamic predictive distribution. (g), (h), and (i) $-\ln$ variance for reconstruction, random trajectories, and longer random trajectories created in the same fashion as (d), (e), and (f), using a model learned with the linear dynamics kernel. Note that the samples do not follow the training data closely, and longer trajectories are attracted to the origin. Figure and captions from [1].

where,

$$L_Y = \frac{D}{2}\ln|K_Y| + \frac{1}{2}\mathrm{tr}(K_Y^{-1}YW^2Y^{\mathrm{T}}) - N\ln|W|, \tag{20}$$

$$L_X = \frac{d}{2}\ln|K_X| + \frac{1}{2}\mathrm{tr}(K_X^{-1}X_{2:\mathrm{N}}W^2X_{2:\mathrm{N}}^{\mathrm{T}}) + \frac{1}{2}x_1x_1^{\mathrm{T}}, \tag{21}$$

The latent coordinates are initialized using PCA and minimization of L is done by alternatively minimizing W in closed form and $X, \bar{\alpha}, \bar{\beta}$ using Scaled Conjugate Gradient (SCG) [1]. The result for this model learning is shown in Fig.4 as the GPDM on a 3D latent space is learnt using the training data of two gait cycles of a person walking. It can be seen that the trajectories of GPDM are much smoother and consecutive poses are much closer to each other than that of GPLVM which consists of a dynamic model in the observation space and is hence more jerky when uncertain. Also since it is a probabilistic model using GP new trajectories can be generated with latent space input and missing trajectories of a missing frames can be generated.

There are other variations of the optimization methods available in the GPDMs. One deals with scaling the L_X term in Eq. 19 by D/d such that the latent dynamical module is optimized by the same factor as the dimensionality reduction module, this leads to smoother trajectories for GPDMs. Also are methods which optimize the hyperparameters separately in one MAP stage and the latent

variables in another MAP stage and this two stage MAP method gives the best result for motions with high variance. The two stage MAP estimate is computationally quite expensive compared to other learning methods. The detailed algorithm for each of these methods is available in [1].

The GPDMs in [1] have dynamical models without a control input making them unsupervised in their methods. In [8] we have Bayesian filtering to estimate the state of a dynamical system based on prediction and observation models. The Bayesian filter apart from input state X also consists of a control variable U to get the observed measurements Y. The observation model is represented as a GP with mean and variance based on previous states and output measurements. The prediction model is represented as a GP based on mean and variance calculated using previous input states and control signals. These models mirror those of GPDMs as the prediction model is a dynamical model except it is also dependent on a control input making it supervised in nature [1]. The Observation model is comparable to the dimensionality reduction module of GPDMs as it maps the input state to the output measurement state. The results of [8] show that the GP based non-parametric Bayesian filters perform much better compared to their parametric counterparts when plugged into different types of filters like particle filters and extended Kalman filters.

In all the methods mentioned above the optimization of the marginal likelihood of Eq. 12 or Eq. 19 is intractable analytically as the kernel consists of nonlinear input terms which can not be marginalized analytically. Hence the latent variables get optimized to a local minima or when given more latent dimensions the likelihood of the data went up because of overfitting. Now we look for variational methods which will make the likelihood marginalization tractable analytically hence improving the quality of results of GPDMs.

3 Variational Gaussian Process Dynamical Systems

The marginal likelihoods of Eq. 12 or Eq. 19 are analytically intractable as mentioned before. Titsias and Lawrence in [4] developed a Bayesian approach to marginalization the latent variable X and optimize the resulting lower bound for the hyperparameters to obtain much better likelihood results. These hyperparameters can allow us to automatically detect the number of dimensions needed to represent the data in the latent space and also help in model comparisons. We take a short glance at the method itself describing the method used to compute the intractable lower bound. Then mention Variational Gaussian Process Dynamical Systems(VGPDS) [5] which use these optimization methods in a dynamical setting like GPDMs.

From Eq.17 we need to compute the following marginal likelihood

$$p(Y) = \int p(Y|X)p(X)dX, \tag{22}$$

which is intractable as X appears nonlinearly inside the inverse of the covariance $K_X + \beta^{-1}I$. A variational distribution of $q(X)$ is chosen such that it is flexible enough to approximate the posterior distribution of $p(X|Y)$ [4].

$$q(X) = \prod_{\mathrm{i}=1}^{\mathrm{N}} \mathcal{N}(x_{\mathrm{n}}|\mu_{\mathrm{n}}, S_{\mathrm{n}}), \tag{23}$$

Next the Jensen's lower bound on the $\log(p(Y))$ is then calculated as,

$$F(q) = \int q(X) \log \frac{P(Y|X)P(X)}{q(X)} dX, \tag{24}$$

$$F(q) = \tilde{F}(q) - KL(q||p), \tag{25}$$

The second term in 25 is the KL divergence and can be calculated as it is between two Gaussian distributions [4]. The first term can then be written as

$$\tilde{F}(q) = \sum_{\mathrm{d}=1}^{\mathrm{D}} \int q(X) \log(p(y_{\mathrm{d}}|X))dx, \tag{26}$$

this integral is still intractable because of the $\log p(y_{\mathrm{d}}|X)$ which has X in a nonlinear function inside the covariance matrix [4]. The idea in [4] is to introduce a GP latent function value $f_{\mathrm{d}} \in \mathbb{R}^N$ such

(a) (b) (c)

Figure 5: The last frame of the training video (a) is smoothly followed by the first frame (b) of the generated video. A subsequent generated frame can be seen in (c). Figure and citation from [5]

that the complete likelihood associated with the marginal $p(y_{\mathrm{d}}|X)$ is

$$p(y_{\mathrm{d}}, f_{\mathrm{d}}|X) = p(y_{\mathrm{d}}|f_{\mathrm{d}})p(f_{\mathrm{d}}|X), \tag{27}$$

Next we sample M auxiliary inducing variables [4] in $u_{\mathrm{d}} \in \mathbb{R}^M$. These are evaluated at inducing input locations $Z \in \mathbb{R}^{M\times d}$. Using the inducing variables we augment the joint probability calculated in Eq. 27,

$$p(y_{\mathrm{d}}, f_{\mathrm{d}}, u_{\mathrm{d}}|X, Z) = p(y_{\mathrm{d}}|f_{\mathrm{d}})p(f_{\mathrm{d}}|u_{\mathrm{d}}, X, Z)p(u_{\mathrm{d}}|Z), \tag{28}$$

The likelihood $p(y_{\mathrm{d}}|X)$ can be equivalently computed from the above augmented model by marginalizing out $p(fy_{\mathrm{d}}, u_{\mathrm{d}})$. Most importantly this is true for any value of the inducing inputs Z [4]. This means that, unlike X, the inducing inputs Z are not random variables. Neither are they model hyperparameters, they are variational parameters. Hence joint distribution of fy_{d} and u_{d}

$$p(f_{\mathrm{d}}, u_{\mathrm{d}}) = p(f_{\mathrm{d}}|u_{\mathrm{d}}, X)\phi(u_{\mathrm{d}}), \tag{29}$$

where $\phi(u_{\mathrm{d}})$ is variational distribution over the inducing variables u_{d} [4]. This can be substituted in Eq. 28 and the K_X matrix would not be inverted in the covariance of the distribution making the marginalization tractable. The complete derivation is solved in [4]. This has been summarized in [4] as "' the variational method allows us to compute a Jensens lower bound on the GPLVM marginal likelihood and the key to obtaining this bound is to introduce auxiliary variables into the model similar to those used in sparse GP regression"'. The hyperparamters after this are calculated using a MAP estimate or gradient descent as usual, but now the hyperparameters are optimized more accurately allowing kernels like ARD to be used without overfitting [5].

Predictions in VGPDS are done like in any Gaussian process calculating the mean and covariance based on GP regression of the marginal probabilities of the future output values based on current or trained output values. With approximations in VGPDS the mean and covariance functions can be calculated analytically. Experiments on modeling human motion capture data by VGPDS show that it has better root mean square error performance than its counterparts. Also using the ARD kernels the model figures out the dimensionality of the data itself. More interesting results were in form of high definition video sequence modeling. The HD video sequence experiments of [5] include the HD video of 150 frames of a woman talking named the Missa dataset. The video is of $103,680$ dimensions. Also trained are periodic sequences of a dog running consisting 60 frames and $230,400$ dimensions and a 9×10^5 dimensional artificial video of ocean waves. The models were tested by extrapolating the videos by 7 frames over the total size of the video and comparing the new frames to results with nearest neighbour reconstruction. VGPDS performed better than nearest neighbour in all cases [5]. Another result which was interesting was to use the model trained by the dog data set to run for 40 more frames. The result was 40 frames of high quality sharp images as shown in Fig. 5 with comparison of length scales of ARD covariance function before and after training on the dog data set in Fig. 6. As can be seen with the ARD kernel the weights of unnecessary dimensions have been turned to zero.

4 Discussion

Probabilistic approach of modeling high dimensional time series data is shown to make the modeling less complex and achieve more realistic results. In this GPDM's have been more successful at getting realistic results without jumps as in GPDMs the latent space poses are close to each other if they occur together.

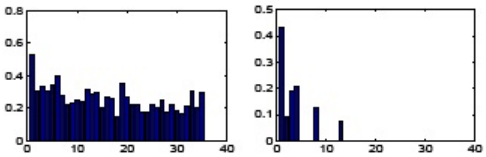

Figure 6: the initial lengthscales (left) of the ARD covariance function and the values obtained after training (right) on the dog data set - as can be seen with the ARD kernel the weights of unnecessary dimensions have been turned to zero after model learning.Figure and caption from [5].

A few comparative studies [9] [10] have been done on GPDM, GPLVM, PCA, and other flavours of GPLVM like back-constrained GPLVM (BC-GPLVM). The study in [9] is on animations and generating realistic looking poses from these similarly trained models. Their results take into account metrics like mean error at a given number of latent space dimensions, learning time, interpolation quality and they have reported that none of the GPLVM based methods have a significant advantage over each other. The results are considerably better than PCA for the purpose of motion synthesis and interpolation.

The study in [10] is done on robotic motion data with a 7- DOF robotics arm. The interpolation error between the poses is measured with root mean square error in the joint or task space between the original and the generated pose. The study suggests that GPLVM based methods do improve the quality of interpolation. But methods in their standard settings do not necessarily lead to successful reconstructions unless the data is densely sampled. It states for example that all GPLVM based methods suggest initializing the latent space variables using PCA but better results have been obtained with ad-hoc parallel lines. On a positive note [10] mentions that dimensionality reduction produces more robust reconstruction in presence of noise also that adding more prior information about the data sets improves the results.

VGPDS is a fully Bayesian approach for modeling dynamical systems through probabilistic nonlinear dimensionality reduction. It is a general method to model complex correlations in high dimensional data[5]. A prominent feature being ARD kernel usage which means we do not need to select the number of latent dimensions. With regards to VGPDS any comparative studies could not be found. VGPDS can probably solve some problems of initializing that were mentioned in [10]. Also VGPDS avoids the cubic complexity of GPs by using variational methods[5] hence it can take in larger dimensional data than previous methods. Future work in this direction is to add more application specific knowledge like sophisticated covariance functions for specific areas like robotics, computer vision or finance.

References

[1] Wang, J.; Fleet, D.;Hertzmann, A. *Gaussian Process Dynamical Models for Human Motion*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008.

[2] Lawrence N. *Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models*. Journal of Machine Learning Research, 2005

[3] Ijspeert, A.;Nakanishi, J.; Pastor, P; Hoffmann, H.; Schaal, S. *Learning nonlinear dynamical systems models*. Neural Computation, 2012

[4] Titsias, M.; Lawrence N. *Bayesian Gaussian Process Latent Variable Model*. Artificial Intelligence and Statistics, 2010

[5] Damianou, A.; Titsias, M.; Lawrence N. *Variational Gaussian Process Dynamical Systems*. Advances in Neural Information Processing Systems, 2011

[6] Bishop, C. *Pattern Recognition and Machine Learning*. Springer, 1st ed. 2006. corr. 2nd printing ed., 2007

[7] Rasmussen, C.; Williams,C. *Gaussian Processes for Machine Learning*. MIT Press, 2006

[8] Ko, J.; Fox, D. *GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models*. Intelligent Robots and Systems, 2008

[9] Quirion,S.; Duchesne,C.; Laurendeau,D.; Marchand,M. *Comparing GPLVM Approaches for Dimensionality Reduction in Character Animation*. Journal of WSCG, 2008

[10] Bitzer, S.; Klanke, S.; Vijayakumar, S. *Does Dimensionality Reduction Improve the Quality of Motion Interpolation?*. ESANN, 2009

[11] Henter, G.; Frean, M.; Kleijn, B. *Gaussian Process Dynamical Models for Nonparametric Speech Representation and Synthesis*. ICASSP, 2012

[12] Geiger, A. *Gaussian Processes for Machine Learning - Seminar at KTH*. Seminar at Universitat Karlsruhe, 2007