Topologically-Constrained Latent Variable Models

Keywords: Gaussian processes, Latent Variable Models, Topology, Dimensionality Reduction, Human Motion

Raquel Urtasun   RURTASUN@CSAIL.MIT.EDU
UC Berkeley EECS & ICSI; CSAIL MIT

David J. Fleet   FLEET@CS.TORONTO.EDU
University of Toronto

Andreas Geiger   GEIGER@MRT.UKA.DE
Karlsruhe Institute of Technology

Jovan Popović   JOVAN@CSAIL.MIT.EDU
CSAIL MIT

Trevor J. Darrell   TREVOR@EECS.BERKELEY.EDU
UC Berkeley EECS & ICSI; CSAIL MIT

Neil D. Lawrence   NEIL.LAWRENCE@MANCHESTER.AC.UK
University of Manchester

Abstract

In dimensionality reduction approaches, the data are typically embedded in a Euclidean latent space. However for some data sets this is inappropriate. For example, in human motion data we expect latent spaces that are cylindrical or a toroidal, that are poorly captured with a Euclidean space. In this paper, we present a range of approaches for embedding data in a non-Euclidean latent space. Our focus is the Gaussian Process latent variable model. In the context of human motion modeling this allows us to (a) learn models with interpretable latent directions enabling, for example, style/content separation, and (b) generalise beyond the data set enabling us to learn transitions between motion styles even though such transitions are not present in the data.

1. Introduction

Dimensionality reduction is a popular approach to dealing with high dimensional data sets. It is often the case that linear dimensionality reduction, such as principal component analysis (PCA) does not ad-

equately capture the structure of the data. For this reason there has been considerable interest in the machine learning community in non-linear dimensionality reduction. Approaches such as locally linear embedding (LLE), Isomap and maximum variance unfolding (MVU) (Roweis & Saul, 2000; Tenenbaum et al., 2000; Weinberger et al., 2004) all define a topology through interconnections between points in the data space. However, if a given data set is relatively sparse or particularly noisy, these interconnections can stray beyond the 'true' local neighbourhood and the resulting embedding can be poor.

Probabilistic formulations of latent variable models do not usually include explicit constraints on the embedding and therefore the natural topology of the data manifold is not always respected [1]. Even with the correct topology and dimension of the latent space, the learning might get stuck in local minima if the initialization of the model is poor. Moreover, the maximum likelihood solution may not be a good model, due e.g., to the sparseness of the data. To get better models in such cases, more constraints on the model are needed.

This paper shows how explicit topological constraints can be imposed within the context of probabilistic la-

---

[1] An exception is the back-constrained GP-LVM (Lawrence & Quiñonero-Candela, 2006) where a constrained maximum likelihood algorithm is used to enforce these constraints.

tent variable models. We describe two approaches, both within the context of the Gaussian process latent variable model (GP-LVM) (Lawrence, 2005). The first uses prior distributions on the latent space that encourage a given topology. The second influences the latent space and optimisation through constrained maximum likelihood.

Our approach is motivated by the problem of modeling human pose and motion for character animation. Human motion is an interesting domain because, while there is an increasing amount of motion capture data available, the diversity of human motion means that we will necessarily have to incorporate a large amount of prior knowledge to learn probabilistic models that can accurately reconstruct a wide range of motions. Despite this, most existing methods for learning pose and motion models (Elgammal & Lee, 2004; Grochow et al., 2004; Urtasun et al., 2006) do not fully exploit useful prior information, and many are limited to modeling a single human activity (e.g., walking with a particular style).

This paper describes how prior information can be used effectively to learn models with specific topologies that reflect the nature of human motion. Importantly, with this information we can also model multiple activities, including transitions between them (e.g. from walking to running), even when such transitions are not present in the training data. As a consequence, we can now learn latent variable models with training motions comprising multiple subjects with stylistic diversity, as well as multiple activities, such as running and walking. We demonstrate the effectiveness of our approach in a character animation application, where the user specifies a set of constraints (e.g., foot locations), and the remaining kinematic degrees of freedom are infered.

2. Gaussian Process Latent Variable Models (GP-LVM)

We begin with a brief review of the GP-LVM (Lawrence, 2005). The GP-LVM represents a high-dimensional data set, $\mathbf{Y}$, through a low dimensional latent space, $\mathbf{X}$, and a Gaussian process mapping from the latent space to the data space. Let $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_N]^T$ be a matrix in which each row is a single training datum, $\mathbf{y}_i \in \Re^D$. Let $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]^T$ denote the matrix whose rows represent the corresponding positions in latent space, $\mathbf{x}_i \in \Re^d$. Given a covariance function for the Gaussian process, $k_Y(\mathbf{x}, \mathbf{x}')$, the likelihood of the data given the latent positions is,

$$p(\mathbf{Y} \mid \mathbf{X}, \bar{\beta}) = \frac{1}{Z_1} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{K}_Y^{-1}\mathbf{Y}\mathbf{Y}^T\right)\right), \quad (1)$$

where Z_1 is a normalization factor, $\mathbf{K}_Y$ is known as the kernel matrix, and $\bar{\beta}$ denotes the kernel hyperparameters. The elements of the kernel matrix are defined by the covariance function, $(\mathbf{K}_Y)_{i,j} = k_Y(\mathbf{x}_i, \mathbf{x}_j)$. A common choice is the radial basis function (RBF), $k_Y(\mathbf{x}, \mathbf{x}') = \beta_1 \exp(-\frac{\beta_2}{2}||\mathbf{x} - \mathbf{x}'||^2) + \frac{\delta_{\mathbf{x},\mathbf{x}'}}{\beta_3}$, where the kernel hyperparameters $\bar{\beta} = \{\beta_1, \beta_2, \beta_3\}$ determine the output variance, the RBF support width, and the variance of the additive noise. Learning in the GP-LVM consists of maximizing (1) with respect to the latent positions, $\mathbf{X}$, and the hyperparameters, $\bar{\beta}$.

When one has time-series data, $\mathbf{Y}$ represents a sequence of observations, and it is natural to augment the GP-LVM with an explicit dynamical model. For example, the Gaussian Process Dynamical Model (GPDM) models the sequence as a latent stochastic process with a Gaussian process prior (Wang et al., 2008) , i.e.,

$$p(\mathbf{X} \mid \bar{\alpha}) = \frac{p(\mathbf{x}_1)}{Z_2} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{K}_X^{-1}\mathbf{X}_{out}\mathbf{X}_{out}^T\right)\right) \quad (2)$$

where Z_2 is a normalization factor, $\mathbf{X}_{out} = [\mathbf{x}_2, ..., \mathbf{x}_N]^T$, $\mathbf{K}_X \in \Re^{(N-1)\times(N-1)}$ is the kernel matrix constructed from $\mathbf{X}_{in} = [\mathbf{x}_1, ..., \mathbf{x}_{N-1}]$, $\mathbf{x}_1$ is given an isotropic Gaussian prior and $\bar{\alpha}$ are the kernel hyperparameters for $\mathbf{K}_X$; below we use an RBF kernel for $\mathbf{K}_X$. Like the GP-LVM the GPDM provides a generative model for the data, but additionally it provides one for the dynamics. One can therefore predict future observation sequences given past observations, and simulate new sequences.

3. Top Down Imposition of Topology

The smooth mapping in the GP-LVM ensures that distant points in data space remain distant in latent space. However, as discussed in (Lawrence & Quiñonero-Candela, 2006), the mapping in the opposite direction is not required to be smooth. While the GPDM may mitigate this effect, it often produces models that are neither smooth nor generalize well (Urtasun et al., 2006; Wang et al., 2008).

To help ensure smoother, well-behaved models, (Lawrence & Quiñonero-Candela, 2006) suggested the use of *back-constraints*, where each point in the latent space is a smooth function of its corresponding point in data space, $x_{ij} = g_j(\mathbf{y}_i; \mathbf{a}_j)$, where $\{\mathbf{a}_j\}_{1 \le j \le d}$ is the set of parameters of the mappings. One possible mapping is a kernel-based regression model, where regression on a kernel induced feature space provides the mapping,

$$x_{ij} = \sum_{m=1}^{N} a_{jm} k(\mathbf{y}_i, \mathbf{y}_m) . \quad (3)$$

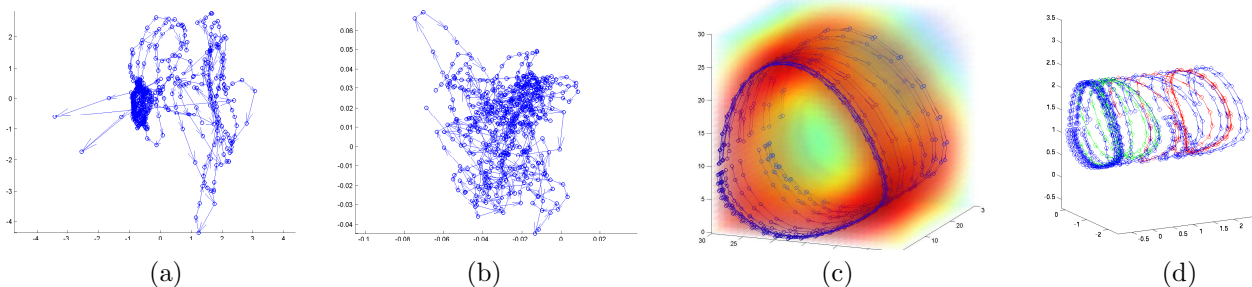(a)   (b)   (c)   (d)

Figure 1. When training data contain large stylistic variations and multiple motions, the generic GPDM (a) and the back-constrained GPDM (b) do not produce useful models. Simulations of both models here do not look realistic. (c,d) Hybrid model learned using local linearities for smoothness (i.e., style) and backconstraints for topologies (i.e., content). The training data is composed of 9 walks and 10 runs performed by different subjects and speeds. (c) Likelihood for the reconstruction of the latent points (d) 3D view of the latent trajectories for the training data in blue, and the automatically generated motions of Figs. 3 and 4 in green and red respectively.

This approach is known as the back-constrained GP-LVM. When learning the back-constrained GP-LVM, one needs to determine the hyperparameters of the kernel matrices (for the back-constraints and the covariance of the GP), as well as the mapping weights, $\{\mathbf{a}_j\}$. (Lawrence & Quiñonero-Candela, 2006) fixed the hyperparameters of the back-constraint's kernel matrix, optimizing over the remaining parameters.

Nevertheless, when learning human motion data with large stylistic variations or different motions, neither GPDM nor back-constrained GP-LVM produce smooth models that generalize well. Fig. 1 depicts three 3–D models learned from 9 walks and 10 runs. The GPDM (Fig. 1(a)) and the back-constrainted GPDM[2] (Fig. 1 (b)) do not generalize well to new runs and walks, nor do they produce realistic animations.

In this paper we show that with a well designed set of back-constraints good models can be learned (Fig. 1(c)). We also consider an alternative approach to the hard constraints on the latent space arising from $g_j(\mathbf{y}_i; \mathbf{a}_j)$. We introduce topological constraints through a prior distribution in the latent space, based on a neighborhood structure learned through a generalized local linear embedding (LLE) (Roweis & Saul, 2000). We then show how to incorporate domain-specific prior knowledge, which allows us to develop motion models with specific topologies that incorporate different activities within a single latent space and transitions between them.

3.1. Locally Linear GP-LVM

The locally linear embedding (LLE) (Roweis & Saul, 2000) preserves topological constraints by finding a representation based on reconstruction in a low dimensional space with an optimized set of local weightings. Here we show how the LLE objective can be combined

---

[2]We use an RBF kernel for the inverse mapping in (3).

with the GP-LVM, yielding a *locally linear GP-LVM* (LL-GPLVM).

The locally linear embedding assumes that each data point and its neighbors lie on, or close to, a locally linear patch on the data manifold. The local geometry of these patches can then be characterized by linear coefficients that reconstruct each data point from its neighbors. This is done in a three step procedure: (1) the K nearest neighbors, $\{\mathbf{y}_j\}_{j \in \eta_i}$, of each point, $\mathbf{y}_i$, are computed using Euclidean distance in the input space, $d_{ij} = ||\mathbf{y}_i - \mathbf{y}_j||^2$; (2) the weights $\mathbf{w} = \{w_{ij}\}$ that best reconstruct each data point from its neighbors are obtained by minimizing $\Phi(\mathbf{w}) = \sum_{i=1}^{N} ||\mathbf{y}_i - \sum_{j \in \eta_i} w_{ij}\mathbf{y}_j||^2$; and (3) the latent positions $\mathbf{x}_i$ best reconstructed by the weights w_{ij} are computed by minimizing $\Phi(\mathbf{X}) = \sum_{i=1}^{N} ||\mathbf{x}_i - \sum_{j \in \eta_i} w_{ij}\mathbf{x}_j||^2$.

In the LLE, the weight matrix $\mathbf{w}$ is sparse (only a small number of neighbors is used), and the two minimizations can be computed in closed form. In particular, computing the weights can be done by solving, $\forall j \in \eta_i$, the following system,

$$\sum_k C_{kj}^{sim} w_{ij}^{sim} = 1 \, , \tag{4}$$

where $C_{kj}^{sim} = (\mathbf{y}_i - \mathbf{y}_k)^T(\mathbf{y}_i - \mathbf{y}_j)$ if $j, k \in \eta_i$, and 0 otherwise. Once the weights are computed, they are rescaled so that $\sum_j w_{ij} = 1$.

The LLE energy function can be interpreted, for a given set of weights $\mathbf{w}$, as a prior that forces each latent point to be locally reconstructed by its neighbors,i.e., $p(\mathbf{X}|\mathbf{w}) = \frac{1}{Z} \exp\left\{-\frac{1}{\sigma^2}\Phi(\mathbf{X})\right\}$, where Z is a normalization constant, and σ^2 represents a global scaling of the prior. Note that strictly speaking this is not a proper prior as it is conditioned on the weights which depend on the training data. Following (Roweis & Saul, 2000), we first compute the neighbors based on the Euclidean distance. For each training point $\mathbf{y}_i$,

we then compute the weights solving Eq. (4).

Learning the LL-GPLVM is then equivalent to minimizing the negative log posterior of the model, [3] i.e.,

$$
\begin{aligned}
\mathcal{L}_{\mathcal{S}} &= \log\ p(\mathbf{Y}|\mathbf{X},\bar{\beta})\ p(\bar{\beta})\ p(\mathbf{X}|\mathbf{w}) \\
&= \frac{D}{2}\ln|\mathbf{K}_Y| + \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_Y^{-1}\mathbf{Y}\mathbf{Y}^T\right) + \sum_i \ln\beta_i \\
&\quad + \frac{1}{\sigma^2}\sum_{k=1}^{d}\sum_{i=1}^{N}\|x_i^k - \sum_{j=1}^{N} w_{ij}^k x_j^k\|^2 + C\ ,\ (5)
\end{aligned}
$$

where C is a constant, and x_i^k is the k-th component of $\mathbf{x}_i$. Note that we have extended the LLE to have a different prior for each dimension. This will be useful below as we incorporate different sources of prior knowledge. Fig. 2 (a) shows a model of 2 walks and 2 runs learned with the locally linear GPDM. Note how smooth the latent trajectories are.

We now have general tools to influence the structure of the models. In what follows we generalize the top-down imposition of topology strategies (i.e. back-constraints and locally linear GP-LVM) to incorporate domain specific prior knowledge.

4. Reflecting Knowledge in Latent Space Structure

A problem for modeling human motion data is the sparsity of the data relative to the diversity of naturally plausible motions. For example, while we might have a data set comprising different motions, such as runs, walks *etc.*, the data may not contain transitions between motions. In practice however, we know that these motions will be approximately cyclic and that transitions can only physically occur at specific points in the cycle. How can we encourage a model to respect such topological constraints which arise from prior knowledge?

We consider two alternatives to solve this problem. First, we show how one can adjust the distance metric used in the locally linear embedding to better reflect different types of prior knowledge. We then show how one can define similarity measures for use with the back-constrained GP-LVM. Both these approaches encourage the latent space to construct a representation that reflects our prior knowledge. They are complementary and can be combined to learn better models.

---

[3]When learning a locally linear GPDM, the dynamics and the locally linear prior are combined as a product of potentials. The objective function becomes $\mathcal{L}_{\mathcal{S}} + \frac{d}{2}\ln|\mathbf{K}_X| + \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_X^{-1}\mathbf{X}_{out}\mathbf{X}_{out}^T\right) + \sum_i \ln\alpha_i$, with $\mathcal{L}_{\mathcal{S}}$ defined as in (5).
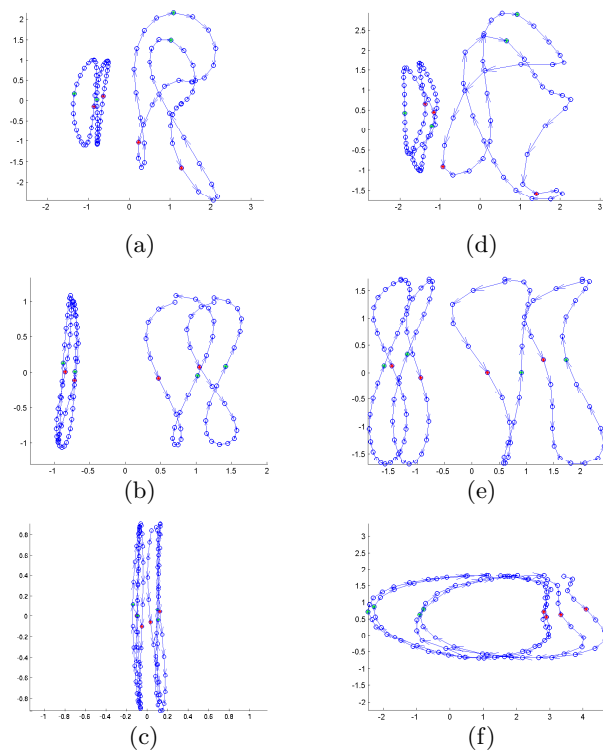

Figure 2. First two dimensions of 3–D models learned using (a) LL-GPDM (b) LL-GPDM with topology (c) LL-GPDM with topology and transitions. (d) Back-constrained GPDM with an RBF mapping. (e) GPDM with topology through backconstraints. (f) GPDM with backconstraints for the topology and transitions. For the models using topology, the cyclic structure is imposed in the last 2 dimensions. The two types of transition points (left and right leg contact points) are shown in red and green, and are used as prior knowledge in (c,f).

4.1. Prior Knowledge through Local Linearities

We now turn to consider how one might incorporate prior knowledge in the LL-GPLVM framework. This is accomplished by replacing the local Euclidean distance measures used in Section 3.1 with other similarity measures. That is, we can modify the covariance used to compute the weights in Eq. (4) to reflect our prior knowledge in the latent space. We consider two examples: the first involves transitions between activities; with the second we show how topological constraints can be placed on the form of the latent space.

Covariance for Transitions Modeling transitions between motions is important in character animation. Transitions can be infered automatically based on similarity between poses (Kovar et al., 2002) or at points of non-linearity of the dynamics (Bissacco, 2005), and they can be used for learning. For example, for motions as walking or running, two types of transitions can be identified: left and right foot ground contacts.

To model such transitions, we define an index on the frames of the motion sequence, $\{t_i\}_{i=1}^N$. We then define subsets of this set, $\{\hat{t}_i\}_{i=1}^M$, which represents frames where transitions are possible. To capture transitions in the latent model we define the elements for the covariance matrix as follows,

$$C_{kj}^{trans} = 1 - \left[\delta_{kj} \exp(-\zeta(t_k - t_j)^2) \right] \qquad (6)$$

with ζ a constant, and $\delta_{ij} = 1$ if t_i and t_j are in the same set $\{\hat{t}_k\}_{k=1}^M$, and otherwise $\delta_{ij} = 0$. This covariance encourages the latent points at which transitions are physically possible to be close together.

Covariance for Topologies We now consider covariances that encourage the latent space to have a particular topology. Specifically we are interested in suitable topologies for walking and running data. Because the data are approximately periodic, it seems appropriate to have a non-Cartesian topology. To this end one can extract the phase of the motion[4], ϕ, and use it with a covariance to encourage the latent points to exhibit a periodic topological structure within a Cartesian space. As an example we consider a cylindrical topology within a 3–D latent space by constraining two of the latent dimensions with the phase. In particular, to represent the cyclic motion we construct a distance function on the unit circle, where a latent point corresponding to phase ϕ is represented with coordinates $(\cos(\phi), \sin(\phi))$. To force a cylindrical topology on the latent space, we specify different covariances for each latent dimension

$$C_{k,j}^{cos} = (\cos(\phi_i) - \cos(\phi_k))(\cos(\phi_i) - \cos(\phi_j)) \quad (7)$$
$$C_{k,j}^{sin} = (\sin(\phi_i) - \sin(\phi_k))(\sin(\phi_i) - \sin(\phi_j)), \quad (8)$$

with $k, j \in \eta_i$. The covariance for the remaining dimension is constructed as usual, based on Euclidean distance in the data space. Fig. 2 (b) shows a GPDM constrained in this way, and in Fig. 2 (c) the covariance is augmented with transitions.

Note that the use of different distance measures for each dimension of the latent space implies that the neighborhood and the weights in the locally linear prior will also be different for each dimension. Here, three different locally linear embeddings form the prior distribution.

4.2. Prior Knowledge with Back Constraints

As explained above, we can also design back-constraints to influence the topology and learn useful

---

[4]The phase can be easily extracted from the data by Fourier analysis or by detecting key postures and interpolating the phases between them. Another idea, not further explored here, would be to optimize the GP-LVM with respect to the phase.

transitions. This can be done by replacing the kernel of Eq. (3). Many kernels have interpretations as similarity measures. In particular, any similarity measure that leads to a positive semi-definite matrix can be interpreted as a kernel. Here, just as we define covariance matrices above, we extend the original formulation of back constraints by constructing similarity measures (i.e., kernels) to reflect prior knowledge.

Similarity for Transitions To capture transitions between two motions, we wish to design a kernel that expresses strong similarity between points in the respective motions where transitions may occur. We can encourage transition points of different sequences to be proximal with the following kernel matrix for the back-constraint mapping:

$$k^{trans}(t_i, t_j) = \sum_m \sum_l \delta_{ml} k(t_i, \hat{t}_m) k(t_j, \hat{t}_l) \qquad (9)$$

where $k(t_i, \hat{t}_l)$ is an RBF centered at $\hat{t}_l$, and $\delta_{ml} = 1$ if $\hat{t}_m$ and $\hat{t}_l$ are in the same set. The influence of the back-constraints is controlled by the support width of the RBF kernel.

Topologically Constrained Latent Spaces We now consider kernels that force the latent space to have a particular topology. To force a cylindrical topology on the latent space, we can introduce similarity measures based on the phase, specifying different similarity measures for each latent dimension. As before we construct a distance function in the unit circle, that takes into account the phase. A periodic mapping can be constructed from a kernel matrix as follows,

$$x_{n,1} = \sum_{m=1}^N a_m^{cos} k(\cos(\phi_n), \cos(\phi_m)) + a_0^{cos} \delta_{n,m},$$
$$x_{n,2} = \sum_{m=1}^N a_m^{sin} k(\sin(\phi_n), \sin(\phi_m)) + a_0^{sin} \delta_{n,m},$$

where k is an RBF kernel function, and $x_{n,i}$ is the ith coordinate of the nth latent point. These two mappings project onto two dimensions of the latent space, forcing them to have a periodic structure (which comes about through the sinusoidal dependence of the kernel on phase). Fig. 2 (e) shows a model learned using GPDM with the last two dimensions constrained in this way (the third dimension is out of plane). The first dimension is constrained by an RBF mapping on the input space. Each dimension's kernel matrix can then be augmented by adding the transition similarity of Eq.(9), resulting in the model shown in Fig. 2 (f).

4.3. Model Combination

One advantage of our framework is that covariance matrices can be combined in a principled manner to form

new covariance matrices. Covariances can be multiplied (on an element by element basis) or added together. Similarly, similarities can be combined. Multiplication has, loosely speaking, an 'AND gate effect', i.e. both similarity measures must agree that an object is similar for their product to express similarity. Adding them produces more of an 'OR gate effect', i.e. if either representation expresses similarity the resulting measure will also express similarity.

The two sections above have shown how to incorporate prior knowledge in the GP-LVM by means of 1) local linearities and 2) back-constraints. In general, the latter should be used when the manifold has a well-defined topology, since it has more influence on the learning. When the topology is not so well defined (e.g., due to noise) one should use local linearities. Both techniques are complementary and can be combined straightforwardly by including priors over some dimensions, and constraining the others through back-constraint mappings. Fig. 1 shows a model learned with LL-GPDM for smoothness and back-constraints for topology.

4.4. Multiple Activities and Transitions

Once we know how to ensure that transition points are close together and that the latent structure has the desired topology, we still need to address two issues. How do we learn models that have very different dynamics? How can we simulate dynamical models that lie somewhere between the different training motions? Our goal in this section is to show how latent models for different motions can be learned independently, but in a shared latent space that facilitates transitions between activities with different dynamics.

Let $\mathbf{Y} = [\mathbf{Y}_1^T, ..., \mathbf{Y}_M^T]^T$ denote training data for M different activities. Each $\mathbf{Y}_m$ comprises several different motions. Let $\mathbf{X} = [\mathbf{X}_1^T, ..., \mathbf{X}_M^T]^T$ denote the corresponding latent positions. When dealing with multiple activities, a single dynamical model cannot cope with the complexity of the different dynamics. Instead, we consider a model where the dynamics of each activity are modeled independently[5]. This has the advantage that a different kernel can be used for each activity.

To enable interpolation between motions with different dynamics, we combined these independent dynamical models in the form of a mixture model. This allows us to produce motions that gracefully transition between different styles and motion types (Figs. 3 and 4).

---

[5]Another interpretation is that we have a block diagonal kernel matrix for the GP that governs the dynamics.

5. Results

We demonstrate the effectiveness of our approach with two applications. First we show how models of multiple activities can be learned, and realistic animations can be produced by drawing samples from the model. We then show an interactive character animation application, where the user specifies a set of sparse constraints and the remaining kinematic degrees of freedom are infered.

5.1. Learning multiple activities

We first considered a small training set comprised of 4 gait cycles (2 walks and 2 runs) performed by one subject at different speeds. Fig. 2 shows the latent spaces learned under different prior constraints. All the models are learned using two independent dynamical models, one for walking and one for running. Note how the phases are aligned when imposing a cylindrical topology, and how the LL-GPDM is smooth. Notice the difference between the LL-GPDM (Fig. 2 (c)) and the backconstrained GPDM (Fig. 2 (f)) when transition constraints are included. Nevertheless, both models ensure that the transition points (shown in red and green) are proximal.

Fig. 1 (c) shows a hybrid model learned using LL-GPDM for smoothness, and back-constraints for topology. The larger training set comprises approximately one gait cycle from each of 9 walking and 10 running motions performed by different subjects at different speeds (3 km/h for walking, 6–12 km/h for running). Colors in Fig. 1 (a) represent the variance of the GP as a function of latent position. Only points close to the surface of the cylinder produce poses with high certainty.

We now illustrate the model's ability to simulate different motions and transitions. Given an initial latent position $\mathbf{x}_0$, we generate new motions by sampling the mixture model, and using mean prediction for the reconstruction. Choosing different initial conditions results in very different simulations (Fig. 1 (d)). The training data are shown in blue. For the first simulation (depicted in green), the model is initialized to a running pose with a latent position not far from walking data. The system transitions to walking quite naturally. The resulting animation is depicted in Fig. 3. For the second example (in red), we initialize the simulation to a latent position far from walking data. The system evolves to different running styles and speeds (Fig. 4). Note how the dynamics, and the strike length, change considerably during simulation.
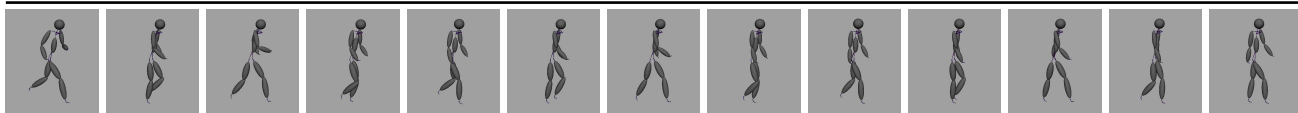
Figure 3. **Transition from running to walking**: The system transitions from running to walking in a smooth and realistic way. The transition is encouraged by incorporating prior knowledge in the model. The latent trajectories are shown in green in Fig. 1 (d).

Figure 4. **Different running styles and speeds**: The system is able to simulate a motion with considerably changes in speed and style. The latent trajectories are shown in red in Fig. 1 (d).
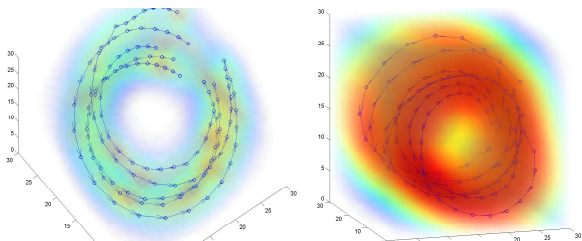

Figure 5. Single activity 3D latent models learned from (left) 5 jumps of 2 different subjects using local linearities, (b) 7 walking cycles of one subject using back-constraints.

5.2. Character animation from constraints

A key problem in the film and game industry is the lack of tools to allow designers to easily generate animations. Traditional techniques such as keyframing are time consuming; an expert can expend days in generating a few seconds of animation. A very useful tool would provide the user with a simple way of generating motions from constraints that she/he defined. Typical constraints are keyframes (i.e., specification of the position of the full body in a particular time instant), or joint trajectories. Here we use the topologically constrained motion models as priors over the space of possible motions.

Our motion estimation formulation is based on a state-space model with a GPDM prior over pose and motion. Given the state, $\phi_t = (\mathbf{y}_t, \mathbf{x}_t)$, the goal is to estimate the state sequence $\phi_{1:T} = (\phi_1, \cdots, \phi_T)$ that satisfies the user constraints $\mathbf{u}_{1:J}$. Inference is performed in a *Batch* mode, so that the state is infered all at once. The posterior can be expressed as

$$p(\phi_{1:T}|\mathbf{u}_{1:J}, \mathbf{M}) \propto p(\mathbf{u}_{1:J}|\phi_{1:T})p(\phi_{1:T}|\mathbf{M}) \quad (10)$$

where we assumed that $p(\mathbf{u}_{1:J})$ is uniformily distributed; all the user constraints are equally probable. The prediction distribution $p(\phi_{1:T}|\mathbf{M})$ can be further factored as follows

$$p(\phi_{1:T}|\mathbf{M}) = p(\mathbf{x}_{1:T}|\mathbf{M}) \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{M}) \quad (11)$$

Rather than approximating the entire posterior, we use hill-climbing to find MAP estimates. Assuming that the user constraints are noise-free, the minimization can be expressed as

$$\min_{\phi_{1:T}} \quad \mathcal{L}_{pose} + \mathcal{L}_{dyn} + \mathcal{L}_{smooth}$$

$$\text{subject to} \quad ||\mathbf{u} - \mathbf{f}(\mathbf{y}_{\psi(\mathbf{u})})|| = \mathbf{0} \quad (12a)$$

where $\mathbf{f}$ is a forward kinematics function (i.e., a function that maps joint angles to positions in the 3D world), $\psi(\mathbf{u})$ is a function that outputs the frame where each constraint $\mathbf{u}_j$ is defined, $\mathcal{L}_{pose} = -\sum_i \ln p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{M})$ and $\mathcal{L}_{dyn} = -\ln p(\mathbf{x}_{1:T}|M)$ are the pose and dynamics likelihood from the GPDM prior (Urtasun et al., 2006), and $\mathcal{L}_{smooth} = \frac{1}{\sigma_s^2} \sum_{t=1}^{T-1} \sum_{j=1}^{P} \frac{1}{\sigma_j^2}(y_{t+1}^j - y_t^j)^2$ is a term that encourage smooth motions, where y_t^j is the j-th component of $\mathbf{y}_t$, and σ_j^2 is a constant that encounters from the fact that each degree of freedom has a different variance.

Initialization is important since a large number of variables need to be optimised and our objective function is non-convex. In particular, we sample the model starting at each training point and use as initialization the sample that is closest to the user constraints.

To demonstrate the effectiveness of our approach we learned models of two different motions, walking and jumping (Fig. 5). We impose smoothness and cyclic topologies using back-constraints for the walking and local linearities for the jumping. We demonstrate the ability of the model to generalize to unseen styles.

We first show how the model can produce realistic animations from a very small set of user defined constraints. The user specifies the contact points of the foot with the ground (first row of Fig. 6) for walking and the foot trajectories for the jumping (third row of Fig. 6), and the rest of the degrees of freedom are infered producing very realistic animations.
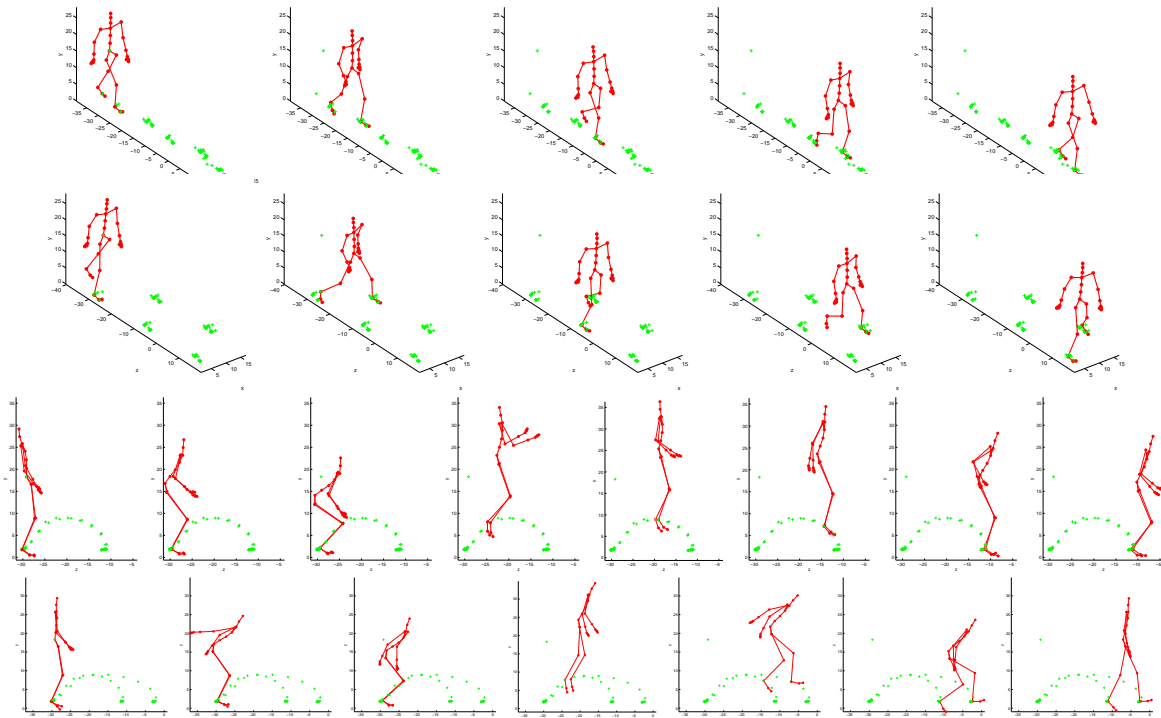
The model can also generalize to styles very different

Figure 6. Animations generated from a set of foot constraints (green). **First row**: Normal walk. **Second row**: Generalization a different style by changing the user constraints to be separate in the coronal plane. **Third row**: Short jump. **Last row**: Longer stylistic jump. *See video at http://people.csail.mit.edu/rurtasun*

from the ones in the training set, by imposing constraints that can be satisfied only by motions very different from the training data. In particular, the user placed the foot constraints far in the coronal plane for walking. Consequently the character opens the legs to satisfy the constraints (second row of Fig. 6). In the last row of Fig. 6 the user places the foot trajectories to create a jump with a style very different from the traning data (the character opens his legs and bends his body and arms in an exaggerated way).

6. Conclusions

In this paper we have proposed a general framework of probabilistic models that learn smooth latent variable models of different activities within a shared latent space. We have introduced a principled way to include prior knowledge, that allow us to learn specific topologies and transitions between the different motions. Although we have learned models composed of walking, running and jumping, our framework is general, being applicable in any data sets where there is a large degree of prior knowledge for the problem domain, but the data availability is relatively sparse compared to its complexity.

References

Bissacco, A. (2005). Modeling and Learning Contact Dynamics in Human Motion In *CVPR.* (pp. 421–428).

Elgammal, A., & Lee, C. (2004). Inferring 3D Body Pose from Silhouettes using Activity Manifold Learning. In *CVPR* (pp. 681–688).

Grochow, K., Martin, S., Hertzmann, A., & Popovic, Z. (2004). Style-based inverse kinematics In *SIGGRAPH.*

Kovar, L., Gleicher, M., & Pighin, F. (2002). Motion Graphs In *SIGGRAPH,* (pp. 473–482).

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with gaussian process latent variable models. *JMLR, 6,* 1783–1816.

Lawrence, N. D., & Quiñonero-Candela, J. (2006). Local distance preservation in the GP-LVM through back constraints In *ICML* (pp. 96–103).

Rasmussen, C. E., & Williams, C. K. (2006). *Gaussian process for machine learning.* MIT Press.

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science, 290.*

Tenenbaum, J., de Silva, V., & Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science, 290,* 2319–2323.

Urtasun, R., Fleet, D. J., & Fua, P. (2006). 3d people tracking with gaussian process dynamical models. In *CVPR,* (pp. 932–938).

Wang, J., Fleet, D. J., & Hertzman, A. (2008). Gaussian process dynamical models . In *PAMI 30*(2), 283–298 .

Weinberger, K. Q., Sha, F., & Saul, L. K. (2004). Learning a kernel matrix for nonlinear dimensionality reduction. In *ICML,* (pp. 106–113).