Rank Priors for Continuous Non-Linear Dimensionality Reduction

Andreas Geiger
Department of Measurement and Control
Karlsruhe Institute of Technology

Raquel Urtasun and Trevor Darrell
UC Berkeley EECS & ICSI

Abstract

Discovering the underlying low-dimensional latent structure in high-dimensional perceptual observations (e.g., images, video) can, in many cases, greatly improve performance in recognition and tracking. However, non-linear dimensionality reduction methods are often susceptible to local minima and perform poorly when initialized far from the global optimum, even when the intrinsic dimensionality is known a priori. In this work we introduce a prior over the dimensionality of the latent space that penalizes high dimensional spaces, and simultaneously optimize both the latent space and its intrinsic dimensionality in a continuous fashion. Ad-hoc initialization schemes are unnecessary with our approach; we initialize the latent space to the observation space and automatically infer the latent dimensionality. We report results applying our prior to various probabilistic non-linear dimensionality reduction tasks, and show that our method can outperform graph-based dimensionality reduction techniques as well as previously suggested initialization strategies. We demonstrate the effectiveness of our approach when tracking and classifying human motion.

1. Introduction

Many computer vision problems involve high dimensional datasets that are computationally challenging to analyze. In such cases it is desirable to reduce the dimensionality of the data while preserving the original information in the data distribution, allowing for more efficient learning and inference. Linear dimensionality reduction techniques (e.g., PCA) have been very popular in the past, due to their simplicity and efficiency. However in practice, as shown below, they can result in poor approximations when dealing with complex datasets.

Graph-based methods, e.g., LLE [19] and Isomap [21] exploit local neighborhood distances to approximate the geodesic distance in the manifold. They have been shown to be very effective when dealing with large datasets that are homogeniously sampled. However, as demonstrated here, they suffer in the presence of noisy and sparse data. Unfortunately, a large set of real world computer vision datasets are sparse. Human motion datasets comprise small numbers of examples from different subjects performing different activities [2, 11]. While these databases are typically densely sampled in time, they are sparse in the motion style and activity type. Object recognition databases [1], also suffer from sparsity: only a few examples may be labeled for categories with large variation in appearence.

Non-linear probabilistic models, such as the GPLVM [13], can recover complex manifolds. They have received considerable attention in recent years, having been applied to human motion tracking [25, 23, 14, 10], detection [3, 18, 7], 3D shape estimation [20] and character animation [6, 24]. However, they have only been applied to small databases typically composed of very few examples of a single activity [25]. Moreover, the latent dimensionality was either chosen by the user [25, 23, 14] or optimized by cross-validation [20], which is computationally expensive.

While their representation power is desirable, such methods suffer from local minima, since they rely on optimization of complex non-linear functions that are generally non-convex. Even with the right dimensionality, they can result in poor representations if initialized far from the optimum [24]. Factors which contribute to this include the distortion introduced by the initialization and the non-convexity of the optimization. This is aggravated when optimizing extremely low-dimensional latent spaces, which is typically the case in applications such as tracking [25].

In this paper we present a new learning paradigm that reduces the problem of local minima by performing *continuous* dimensionality reduction. In contrast to previous GPLVM-based approaches, no distortion is introduced by an initialization step in our approach, since the latent coordinates are initialized to be the original observations. By introducing a prior over the dimen-

sionality of the latent space that encourages sparsity of the singular values, our method is able to simultaneously estimate the latent space and its dimensionality.

Regularization via sparsity has recently become a focus of attention in the machine learning and vision communities, and has been applied to feature selection for linear dimensionality reduction, e.g. Sparse PCA [28], or transfer learning [17]. Here we are interested in learning non-linear low-dimensional latent spaces; to our knowledge, ours is the first non-linear dimensionality reduction technique that penalizes the latent space rank and simultaneously optimizes the structure of a non-linear latent space as well as its intrinsic dimensionality.

In the remainder of the paper we first briefly review latent variable models. We then show how to incorporate a rank prior in the optimization of a non-linear probabilistic model, and demonstrate our approach when tracking and classifying complex articulated human body motions.

2. Background: Latent Variable Models

Latent Variable Models (LVMs), e.g., Probabilistic PCA [22], assume that the data has been generated by some latent (unobserved) random variables that lie on or close to a low-dimensional manifold. Let $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N]^T$ be the set of observations $\mathbf{y}_i \in \Re^D$, and let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]^T$ be the set of latent variables $\mathbf{x}_i \in \Re^Q$, with $Q \ll D$. Probabilistic LVMs relate the latent variables to a set of observed variables via a probabilistic mapping, $y^{(d)} = f(\mathbf{x}) + \eta$, with $y^{(d)}$ the d-th coordinate of $\mathbf{y}$, and $\eta \sim \mathcal{N}(0, \theta_3)$ iid Gaussian noise.

The Gaussian Process Latent Variable Model (GPLVM) [13] places a Gaussian process prior over the space of mapping functions f. Marginalizing over f and assuming conditional independence of the output dimensions given the latent variables results in

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{Y}^{(d)}|\mathbf{0}, \mathbf{K})$$

where $\mathbf{Y}^{(d)}$ is the d-th column in $\mathbf{Y}$, and $\mathbf{K}$ is the covariance matrix, typically defined in terms of a kernel function. Here we use an RBF + noise kernel, $k(\mathbf{x}_i, \mathbf{x}_j) = \theta_1 \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\theta_2^2}\right) + \theta_3 \delta_{ij}$, since it allows for a variety of smooth non-linear mappings using only a limited number of hyperparameters, $\Theta = \{\theta_1, \theta_2, \theta_3\}$.

Learning is performed by maximizing the posterior $p(\mathbf{X}|\mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{X})p(\mathbf{X})$ with respect to the latent variables $\mathbf{X}$ and the kernel hyperparameters Θ. $p(\mathbf{X})$ encodes prior knowledge about the latent space $\mathbf{X}$.

PCA and graph-based techniques are commonly used to initialize the latent space in GPLVM-based dimensionality reduction; both offer closed-form solutions. However, as shown below, PCA [16] cannot capture non-linear dependencies, LLE [19] gives a good initialization *only* if the data points are uniformly sampled along the manifold, and Isomap [21] has difficulty with non-convex datasets [9]. Generally, when initialized far from the global minimum, the GPLVM optimization can get stuck in local minima [13, 24].

To avoid this problem different priors over the latent space have been developed. In [27] a prior was introduced in the form of a Gaussian process over the dynamics in the latent space. This results in smoother models but performs poorly when learning stylistic variations of a motion or multiple motions. In [24] a prior over the latent space, inspired by the LLE cost function, that encourages smoothness and allows the introduction of prior knowledge, e.g., topological information about the manifold, was proposed. However, such prior knowledge is not commonly available, reducing the applicability of this technique. In contrast, in this paper we introduce a generic prior that requires no specific prior knowledge, directly penalizing the dimensionality of the latent space to learn effective low-dimensional representations.

3. Continuous Dimensionality Reduction via Rank Priors

In this section we introduce a continuous dimensionality reduction technique that initializes the latent space to the observation space to avoid initial distorsions, and learns the latent space and its dimensionality by introducing a prior that penalizes latent spaces with high dimensionality. The dimensionality of the latent space can be described by the rank of the *Gram* matrix of the latent coordinates, which can be computed as the number of non-zero singular values of $\mathbf{X}$. However, it is difficult to enforce directly a prior on the rank since it is a discrete quantity.

Instead, we propose a relaxation that results in a penalty function which gradients are continuous and can be easily computed. In particular, we introduce a prior of the form

$$p(\mathbf{X}) = \frac{1}{Z} \exp\left(-\alpha \sum_{i=1}^{D} \phi(s_i)\right) \tag{1}$$

where s_i are the normalized singular values of the mean-subtracted matrix of latent coordinates, with D the dimensionality of the latent space, and Z a normalization constant.

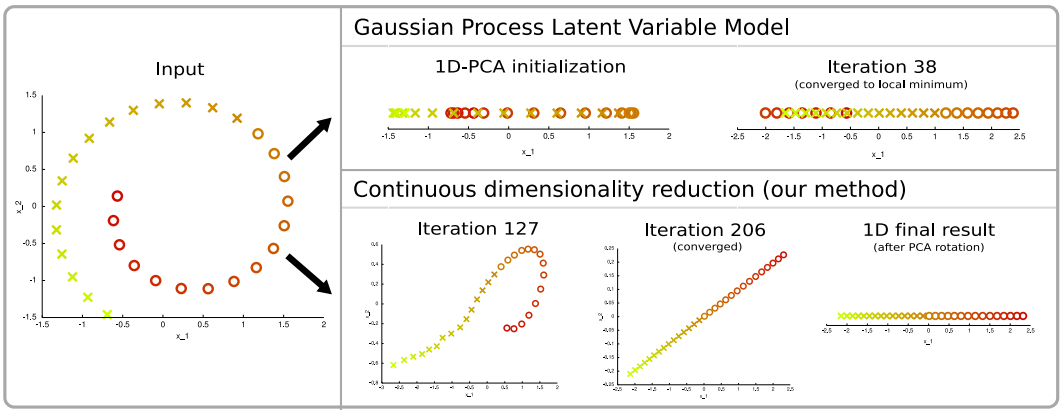


Figure 1. **Comparison of our Rank Prior with a GPLVM initilized to PCA**: The goal is to recover the 1D manifold that is embedded in a 2D space. The GPLVM gets stuck in local minima very early (upper row) since the PCA initialization does not capture non-linear dependencies, whereas our method decreases dimensionality gradually and recovers the correct structure (lower row).

Different penalty functions ϕ can be considered. Common choices for sparsity are the power family and the (generalized) elastic net [4]. In the power family

$$\phi(s_i, p) = |s_i|^p \tag{2}$$

sparsity is achieved for $p \leq 1$. The L_2 norm (i.e., $p = 2$) is a well studied penalty, but does not encourage sparsity. It is equivalent to a Gaussian prior over the singular values. The most commonly used penalty that encourage sparsity is the L_1 norm (i.e., $p = 1$), that results in a Laplace prior over the singular values in Eq. (1). This case is in general attractive since the penalty function is linear, and when the objective function is also linear the optimization can be effectively solved with a Linear Program, even with large number of variables [17].

However here we are interested in learning non-linear latent spaces; our objective function is non-linear even when ϕ is linear. In particular, we minimize the negative log posterior

$$\mathcal{L} = \frac{D}{2} \ln |\mathbf{K}| + \frac{D}{2} tr(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T) + \alpha \sum_{i=1}^{D} \phi(s_i) \ , \tag{3}$$

where α controls the influence of the penalty in the optimization. Of particular interest to us are functions ϕ that drive small singular values faster towards 0 than larger ones. Examples of such functions are the power family with $p < 1$, the logarithmic and sigmoid functions. In practice we use a logarithmic prior for all of our experiments since converges faster to a sparse solution

$$\phi(s) = \ln(1 + \beta s^2)$$

with β a constant.

The derivatives of our rank prior with respect to the latent coordinates $\mathbf{X}$ are

$$\frac{\partial}{\partial X_{ij}} \left(\sum_{m=1}^{D} \phi(s_m) \right) = \frac{1}{\sqrt{N-1}} \sum_{m=1}^{D} \frac{\partial \phi(s_m)}{\partial s_m} U_{im} V_{jm} \ , \tag{4}$$

where we have used the fact that the derivatives of the normalized singular values with respect to the latent coordinates can be easily computed as $\frac{\partial s_m}{\partial x_{ij}} = \frac{1}{\sqrt{N-1}} U_{im} V_{jm}$ [15], with $\mathbf{U}$ the matrix of left-singular vectors, and $\mathbf{V}$ the matrix of right-singular vectors of $\mathbf{X}$. The value of $\frac{\partial \phi(s_m)}{\partial s_m}$ depends on the sparsity function and the derivatives of the first two terms in Eq. (3) with respect to the latent coordinates $\mathbf{X}$ and the kernel hyperparameters Θ are given in [13].

Minimizing Eq. (3) results in a reduction of the energy of the spectrum (since the singular values are minimized). To prevent this from happening, one can instead solve a constrained optimization problem such that the energy of the singular values remains constant,

$$\min \mathcal{L} \quad s.t. \ \ \forall i \ s_i \geq 0, \quad \Delta E = 0, \tag{5}$$

where $\Delta E = E(\mathbf{Y}) - E(\mathbf{X})$ is the difference of energies of the observation space and the latent space, and the energy is computed as $E(\mathbf{X}) = \sum_i s_i^2$. We use SNOPT [5], a non-linear constraint optimizer to minimize Eq. (5).

Finally, we choose the dimensionality of the latent space to be

$$Q = \text{argmax}_i \frac{s_i}{s_{i+1} + \epsilon} \tag{6}$$

where $\epsilon \ll 1$, and $s_1 \geq s_2 \geq \cdots \geq s_D$.[1] Once Q is com-

[1]This strategy is commonly used in statistics to compute the amount of signal sources in noisy data.

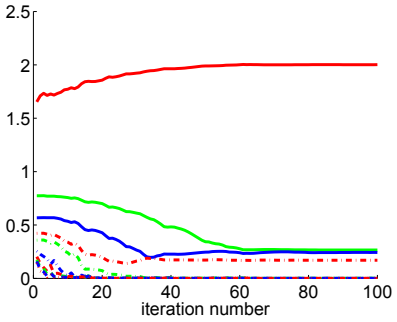


Figure 2. **Spectrum of a 30D motion database**: Evolution of the first ten singular values as a function of the optimization iteration number for a logarithmic sparsity penalty function.

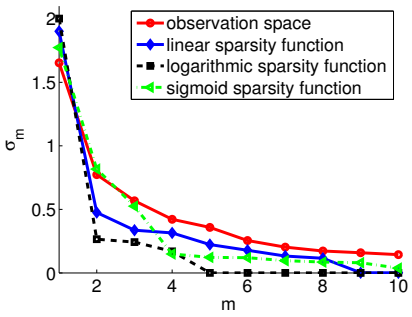


Figure 3. **Comparison of different penalty functions** on a 30D motion database. Note that the logarithmic penalty imposes sparsity more strongly.

puted, we apply PCA in the optimized low-dimensional space. Note that the mapping is still non-linear since PCA is performed in the latent space, not in the observation space, and simply rotates the latent coodinates to produce the most compact Q-dimensional representation. The last step consists of refining the kernel hyperparamters by optimizing $p(\mathbf{Y}|\mathbf{X}, \Theta)$ with respect to Θ keeping $\mathbf{X}$ fix.

Fig. 1 compares the GPLVM (initialized via PCA) with the result of optimizing Eq. (5) on a toy example, where a 1D manifold is embedded in a 2D space. PCA provides a non-optimal initialization, and the GPLVM gets trapped in local minima, whereas our method recovers the correct structure. Note that our final PCA projection rotates the latent space and results in a 1D manifold. In this example, using spectral methods could lead to a successful initialization for the GPLVM. However, for more complex datasets this is not necessarily the case in general, as shown in Figs. 4 and 8.

Fig. 2 depicts the evolution of the first ten singular values when optimizing Eq. (5) with a logarithmic penalty function for a 30D motion database. Note how our method drops dimensions as the optimization evolves (i.e., the smallest singular values drop to zero within the first few iterations). The behavior of different penalty functions is shown in Fig. 3.

When the observations are high-dimensional, one can reduce the complexity by whitening the observations using PCA, removing negligible singular values. The complexity of our approach is of the same order as the complexity of the GPLVM, i.e., $O(N^3)$, since the complexity of computing the singular values of X is $\min(O(P^3), O(N^3))$, with P the dimensionality of the whitened space, and N the number of examples. When using sparsification the complexity can be reduced.

We initialize the kernel width using neighborhood distances, and we set the noise variance to 0.0001. A value of $\alpha \in (1, 10)$ is used in practice.

4. Examples and Experimental results

In this section we demonstrate the effectiveness of our approach to discover the latent structure and its dimensionality in a variety of artificial datasets. We then illustrate the application of our method to the problem of tracking and classifying 3D articulated motion.

4.1. Dimensionality and latent structure estimation

Graph-based methods rely on local neighborhoods to unravel the data and discover the underlying latent structure, but often suffer in the presence of sparse and noisy data. We illustrate this problem on a sparse swiss role, which is a 2D manifold embedded in a 3D space. We simulate sparsity by providing as training data only the black points in Fig. 4 (a). The first row in Fig. 4 (b) shows the result of applying PCA, Isomap, Laplacian Eigenmaps, LLE, LTSA and MVU (see [26] for a review on these techniques). The second row depicts our technique and the result of optimizing the GPLVM with these algorithms as initialization. The last two rows of Fig. 4 (b) show the test data (i.e., colored samples) reconstructed in the 2D latent space and in the original 3D space. Note that our method, unlike PCA, graph-based techniques and the GPLVM with any of the initializations, is able to recover the correct structure.

We evaluate quantitatively the manifolds estimated by the different algorithms computing both a global and a local measure of accuracy. The *reconstruction error* is a global measure of the ability to generalize, and was obtained by first finding the latent coordinates $\mathbf{x}^*$ of the test data $\mathbf{y}^*$ by maximizing $p(\mathbf{x}^*|\mathbf{y}^*, \mathbf{X}, \mathbf{Y})$, and then computing the average mean prediction error $\frac{1}{N_t}\sum_i \|\mu(\mathbf{x}_i^*) - \mathbf{y}_i^*\|_2$, with N_t the number of test data. The *relationship error*, R_{error}, measures how well local neighborhoods are preserved and is defined as $R_{error} = \sum_{i=1}^{N_t}\sum_{j\in\eta_i}\left(\Gamma_{i,j} - \bar{\Gamma}_{i,j}\right)^2$, where η_i is the

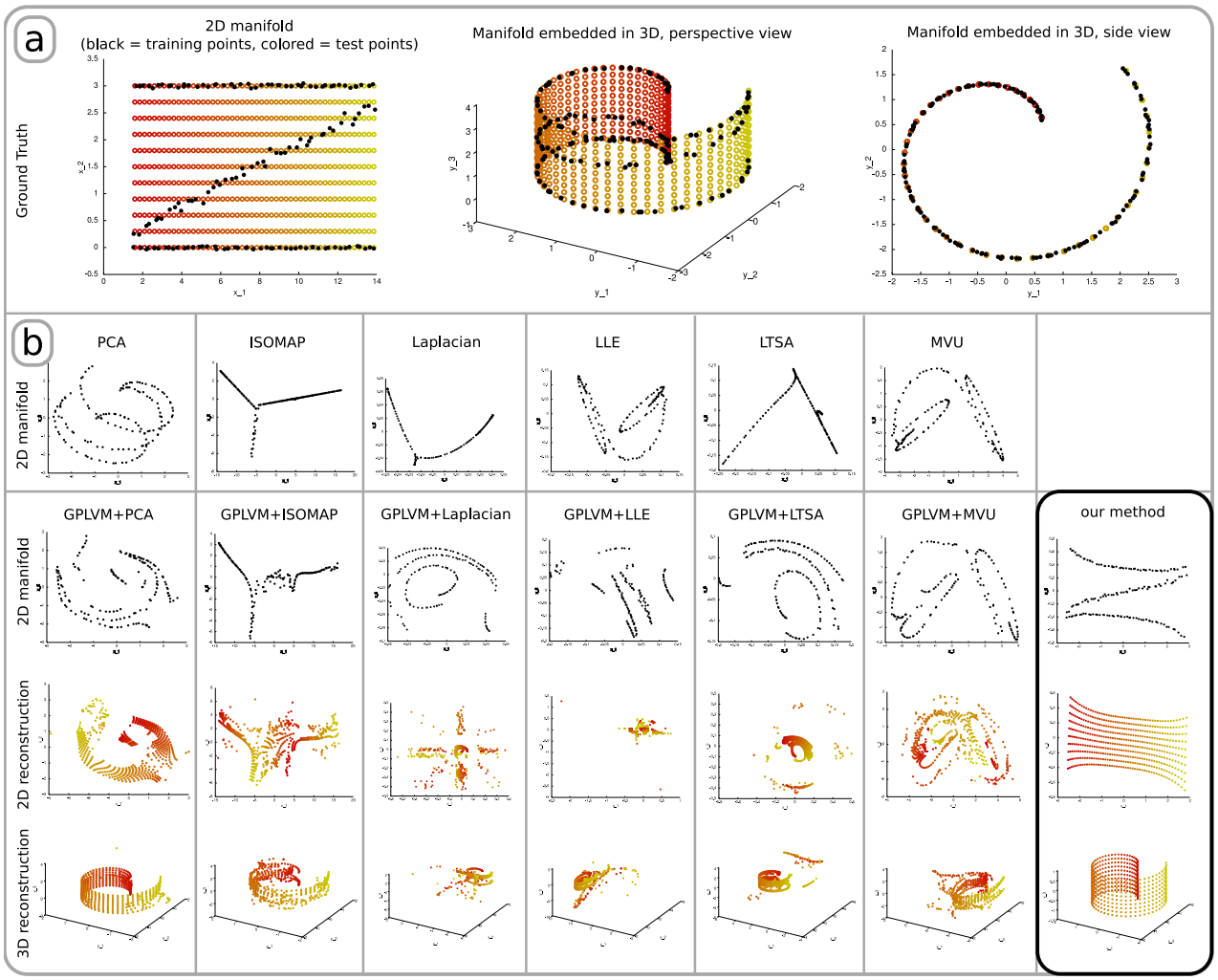


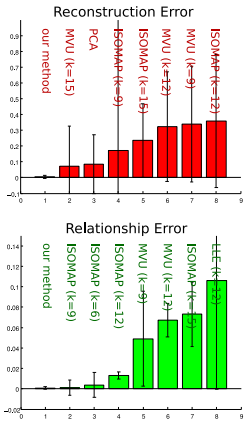
Figure 4. **Finding a 2D manifold in 3D space on a sparsely sampled swiss roll.** Only a sparse, noisy subset (depicted in black) of the full manifold is assumed to be known (a). (b) shows the initialization (with neighborhood size k=6), GPLVM result and 2D/3D reconstruction of the full manifold (from top to bottom).

set of neighbors of the i-th test data, $\Gamma_{i,j} = \frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\|\mathbf{y}_i - \mathbf{y}_j\|_2}$ is the ratio between the distances in the latent space and in the observation space for two neighbors, and $\bar{\Gamma}_{i,j}$ is the mean ratio in the local neighborhood. Fig. 5 depicts the two error measures when performing the experiment in Fig. 4 averaged over 20 random partitions of the data. We use a local neighborhood of size 4 to compute the relationship error in all experiments, and a logarithmic sparsity function with $\alpha = \beta = 10$, and $\Theta = \{0.5, 1.5, 0.01\}$ for our approach. We report results over a wide range of parameters for the different techniques. The hyperparameters were optimized for the GPLVM baselines. Note that our method outperforms the baselines independent of the initialization used for the GPLVM.

We further illustrate our method on 5 complex synthetic examples. In Fig. 6 (a) a spiral with a wide separation between rings is reduced to a 1D manifold. When the distance between the different rings decreases, the manifold dimensionality changes from 1D to 2D (see Fig. 6 (b)), since relationships between points that have the same phase are considered. In Fig. 6 (c) a 2D manifold from a cut-off sphere sampled along longitudinal lines is recovered. The manifold in Fig. 6 (d) is determined to be 3D, while its truncated version in Fig. 6 (e) is estimated to be 2D.

4.2. Tracking and classifying human motion

We conducted experiments tracking and classifying complex motions in synthetic and real data. We created semi-synthetic databases using motion capture data, where the task is, given 2D joint locations in



	Reconstruction Error		Relationship Error	
	mean	stddev	mean	stddev
our method	**0.0041**	**0.0107**	**0.0008**	**0.0013**
PCA init	0.0845	0.1860	0.4232	0.1915
ISOMAP init (k=6)	0.3813	0.1794	0.0038	0.0120
ISOMAP init (k=9)	0.1720	0.8453	0.0011	0.0076
ISOMAP init (k=12)	0.3583	0.4221	0.0130	0.0035
ISOMAP init (k=15)	0.2350	0.2326	0.0731	0.0314
Laplacian init (k=6)	3.7027	2.6553	9.2735	2.5540
Laplacian init (k=9)	2.3136	1.3150	0.6426	8.1716
Laplacian init (k=12)	1.6038	0.2878	7.9784	1.5029
Laplacian init (k=15)	0.5561	2.0321	2.2400	0.0508
LLE init (k=6)	3.5514	2.1871	7.5591	1.4807
LLE init (k=9)	2.6175	2.5141	8.5485	1.4881
LLE init (k=12)	1.6235	1.1386	0.1058	0.1064
LLE init (k=15)	2.7300	4.5488	1.7820	8.9215
LTSA init (k=6)	2.6377	0.7273	1.2636	1.6498
LTSA init (k=9)	3.5149	3.8835	2.0575	2.4038
LTSA init (k=12)	2.1734	3.2099	2.3700	3.0718
LTSA init (k=15)	4.0950	3.6681	2.1819	4.5150
MVU init (k=6)	0.3783	0.5381	0.1238	0.0055
MVU init (k=9)	0.3383	0.3665	0.0491	0.0465
MVU init (k=12)	0.3228	0.3477	0.0672	0.0164
MVU init (k=15)	0.0706	0.2560	0.5856	2.8057

Figure 5. **Quantitative performance on a synthetic sparse swiss roll example** Reconstruction and Relationship Error for the experiment in Fig. 4 averaged over 20 random partitions of the data. (Left) 8 best dimensionality reduction techniques. (Right) More detailed results, including PCA and graph-based methods with different neighborhood sizes.

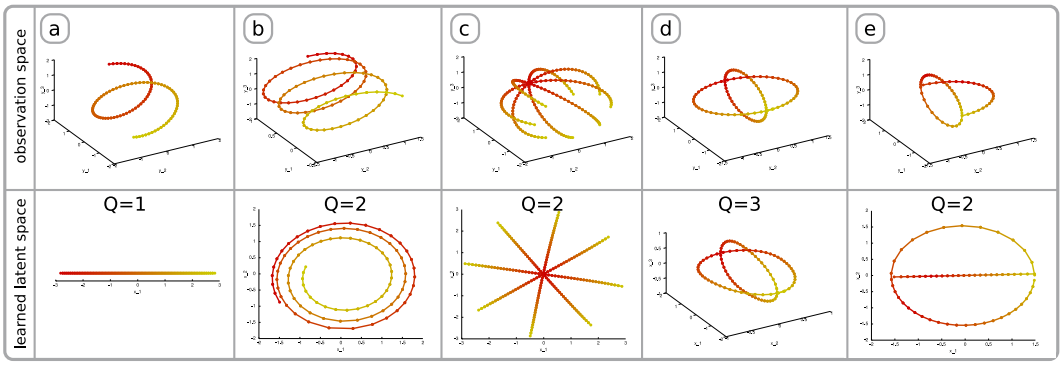
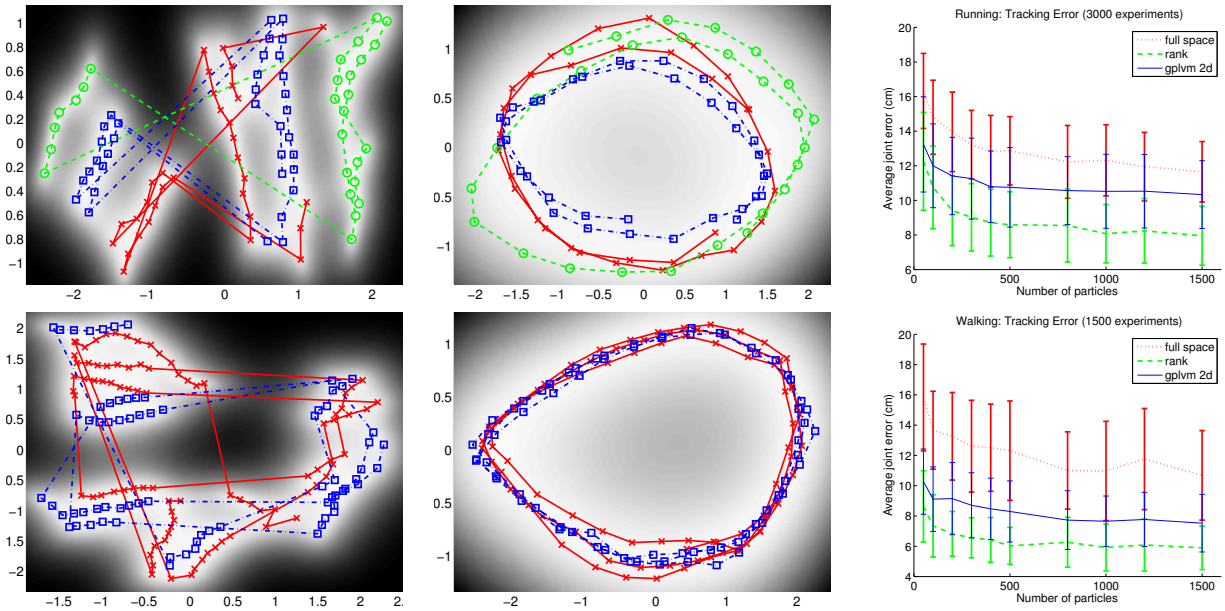


Figure 6. **Dimensionality estimation.** (Top) Five 2D manifolds embedded in 3D. (Bottom) Latent spaces and dimensionalities Q learned using our continous dimensionality reduction method.

monocular images, infer the 3D pose. The databases were composed of running examples from 3 different subjects, and walking examples from a single subject, all 62D. We split the data into training and testing by randomly sampling 3 running and 2 walking motions for training and 10 running and 5 walking motions for testing. Note that most of the splits contain training motions from only 2 or fewer subjects, requiring generalization to unseen styles. We use the Condensation algorithm [12] with a second-order Markov model and the reprojection error as image likelihood [25, 23]. We compare our approach to two baselines: tracking in the original space, and tracking in a latent space learned using GPLVM with PCA initialization. Fig. 7 (right) depicts tracking accuracy in cm averaged over 10 splits as a function of the number of particles. Note that our approach (green) outperforms significantly the baselines. Tracking in the original space (red) results in the worst performance. For all splits, our approach discovered a 2D latent space. GPLVM with PCA initialization performes worse than our approach since it learns non-smooth latent spaces, as illusted in Fig. 7 (left).

In the second experiment we track and classify human motion from real images in a kitchen domain [8]. The dataset consists of multiple instances of rolling, milling and brooming motions performed in front of a multi-camera array and synchronized mocap. Our method found a 3D latent space from 30D joint angle observations of 2 trials for each of the 3 different motions ($N = 1120$ training examples). Note that in previous work, without the inclusion of hand-tuned prior knowledge, GPLVM-based approaches were not able to learn latent spaces with multiple motions [24]. We used a simple sparsification technique in order to speed up learning: Instead of optimizing one single GP, we subdivided the training data into overlapping *local* and *global* sets and calculate the product of those small-sized GPs. This reduces the computational cost from $\mathcal{O}(N^3)$ to $\mathcal{O}(2N^2)$. Fig. 8 shows the result of learning



(GPLVM initialized with PCA) (our method) (Error comparison)

Figure 7. **Tracking running (top) and walking (bottom) motions from 2D mocap data.** (Left) Example of a 2D space learned with GPLVM initialized to PCA. (Middle) Latent space learned with our rank priors. (Right) Tracking performance in cm for our approach vs. tracking in the original space and GPLVM initialized with PCA, when avaraged over 10 splits. Note that our method outperforms the baselines for all number of particles.

	GPLVM initialized LLE	Our approach
F_{score}	0.7312	3.9778
R_{error}	2.4105	0.0180

Table 1. Fisher score and reconstruction error for GPLVM initialized with LLE and our approach.

such motions using our method. Note that the latent space is smooth (i.e., consecutive frames in time are close in latent space), and separates well the different classes. To quantify the latter, we computed the Fisher score defined as $F_{score} = tr\left(S_w^{-1} S_b\right)$, where S_w is the within class matrix and S_b is the between class matrix. Smoothness implies lower relationship error, as depicted by Table 1. Note that our method performs significantly better than traditional GPLVM in terms of the relationship error and the fisher score.

Fig. 9 depicts tracking and classification performance for the milling and rolling motions. The 3D latent space learned by our approach is depicted by Fig. 8. For tracking we used a particle filter in the low dimensional space with second-order Markov dynamics. Our image likelihood is based on low-level silhouette features. We labeled the data using 7 classes (rest, grasp pin, rolling, grasp broom, brooming, grasp mill, milling) and used Nearest Neighbors for classification. Our method significantly outperforms the GPLVM with LLE initialization in both tracking accuracy and classification performance.

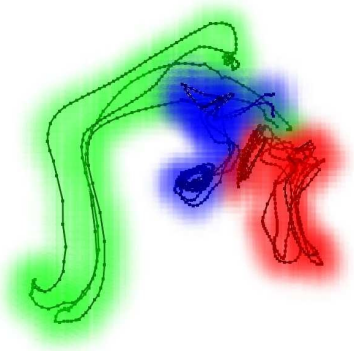


Figure 8. **Learning different types of motion into one single 3D latent space** using our rank prior. The variance of the mapping is depicted in transparency (red = rolling, green = brooming, blue = milling).

5. Conclusions and Future Work

In this paper we have presented a new method for non-linear dimensionality reduction that penalizes high dimensional spaces and results in an optimization problem that continuously reduces dimensionlity while solving for the latent coordinates. Our approach can discover the structure of the latent space and its intrinsic dimensionality, without an ad-hoc initialization step. Our approach has proven superior to PCA, graph-based and non-linear dimensionality reduction

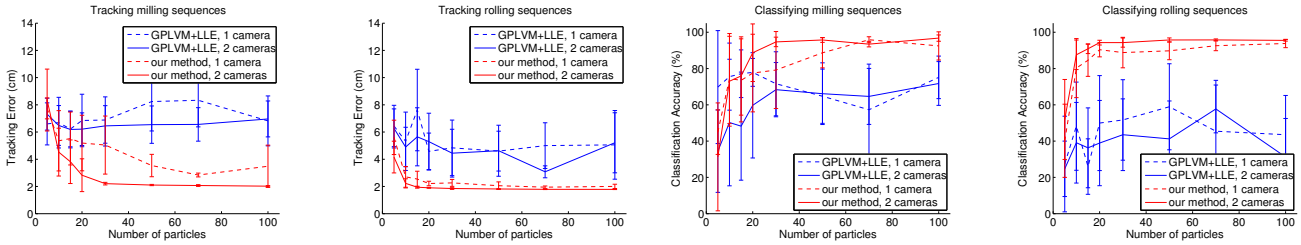


Figure 9. **Tracking and classification performance** for milling and rolling motions using our method (red) and GPLVM initialized to LLE (blue) as a function of the number of particles used in the particle filter.

techniques in a variety of tasks involving synthetic and real-world databases, including tracking and classifying human motion. While our approach avoids most of the local minima present in the GPLVM, the objective function remains non-linear; we plan as future work to investigate simulated annealing and stochastic gradient descend to further mitigate this problem.

Acknowledgements

The first author would like to thank Rainer Stiefelhagen for his support and advice. The last author would like to acknowledge support from NSF #IIS-0704479.

References

[1] Caltech. www.vision.caltech.edu/Image_Datasets/.

[2] CMU Mocap Database. http://mocap.cs.cmu.edu/.

[3] C. H. Ek, P. H. Torr, and N. D. Lawrence. Gaussian process latent variable models for human pose estimation. In *MLMI*, pages 132–143.

[4] J. H. Friedman. Fast sparse regression and classification. Technical report, Standford, 2008.

[5] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. Technical Report NA–97–2, 1997.

[6] K. Grochow, S. Martin, A. Hertzmann, and Z. Popovic. Style-based inverse kinematics. In *SIGGRAPH*, 2004.

[7] A. Gupta, T. Chen, F. Chen, D. Kimber, and L. S. Davis. Context and observation driven latent variable model for human pose estimation. In *CVPR*, 2008.

[8] T. F. A. W. H. Koehler, M. Pruzinec. Automatic human model parametrization from 3d marker data for motion recognition. In *Proceedings of WSCG*, 2008.

[9] S. Harmeling. Exploring model selection techniques for nonlinear dimensionality reduction. Technical report, Edinburgh University, 2007.

[10] S. Hou, A. Galata, F. Caillette, N. Thacker, and P. Bromiley. Real-time body tracking using a gaussian process latent variable model. *ICCV*, 2007.

[11] Humaneva. http://vision.cs.brown.edu/humaneva/.

[12] M. Isard and A. Blake. Condensation conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998.

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

[14] K. Moon and V. Pavlovic. Impact of dynamics on subspace embedding and tracking of sequences. In *CVPR*, pages 198–205, 2006.

[15] T. Papadopoulo and M. I. A. Lourakis. Estimating the jacobian of the singular value decomposition: Theory and applications. In *Research report, INRIA*, 2000.

[16] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2, 1901.

[17] A. Quattoni, M. Collins, and T. Darrell. Transfer learning for image classification with sparse prototype representations. In *CVPR*.

[18] A. F. R. Navaratnam and R. Cipolla. The joint manifold model for semi-supervised multi-valued regression. Rio de Janeiro, Brazil, 2007.

[19] S. T. Roweis and L. K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, 2000.

[20] M. Salzmann, R. Urtasun, and P. Fua. Local deformation models for monocular 3d shape recovery. In *CVPR*, 2008.

[21] J. B. Tenenbaum, V. d. Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 2000.

[22] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal Of The Royal Statistical Society Series B*, 61(3):611–622, 1999.

[23] R. Urtasun, D. J. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. 2006.

[24] R. Urtasun, D. J. Fleet, A. Geiger, J. Popovic, T. Darrell, and L. N.D. Topologically-constrained latent variable models. In *ICML*, 2008.

[25] R. Urtasun, D. J. Fleet, A. Hertzman, and P. Fua. Priors for people tracking from small training sets. In *ICCV*, pages 403–410, Beijing, China, 2005.

[26] L. J. P. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality reduction: A comparative review. 2007.

[27] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *PAMI*, 30(2):283–298, 2008.

[28] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15, 2004.