Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2008-056 September 26, 2008

Rank Priors for Continuous Non-Linear Dimensionality Reduction

Andreas Geiger, Raquel Urtasun, Trevor Darrell, and Rainer Stiefelhagen

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Rank Priors for Continuous Non-Linear Dimensionality Reduction

Andreas Geiger
Department of Measurement and Control
Karlsruhe Institute of Technology
geiger@mrt.uka.de

Raquel Urtasun, Trevor Darrell
MIT CSAIL
UC Berkeley EECS & ICSI
{rurtasun,trevor}@csail.mit.edu

Rainer Stiefelhagen
Interactive Systems Laboratory
Karlsruhe Institute of Technology
stiefel@ira.uka.de

Abstract

Non-linear dimensionality reduction methods are powerful techniques to deal with high-dimensional datasets. However, they often are 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, and simultaneously optimize both the latent space and its intrinsic dimensionality. Ad-hoc initialization schemes are unnecessary with our approach; we initialize the latent space to the observation space and automatically infer the latent dimensionality using an optimization scheme that drops dimensions in a continuous fashion. We report results applying our prior to various tasks involving probabilistic non-linear dimensionality reduction, and show that our method can outperform graph-based dimensionality reduction techniques as well as previously suggested ad-hoc initialization strategies.

1 Introduction

Many real-world problems involve high dimensional datasets that are computationally challenging to handle. 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 methods (e.g., PCA) are efficient but can miss important structure in the data; graph-based techniques, e.g., LLE [7] and Isomap [9], capture non-linear dependencies but require highly dense and homogeneously sampled manifolds for accurate modeling.

Non-linear dimensionality reduction techniques can be applied to more complex data, but generally suffer from local minima. Choosing the dimensionality of the latent space is non-trivial, and existing methods typically rely on cross-validation. Even when given the correct latent dimensionality, these techniques often do not succeed in practice when initialized far from the global minimum [11]. Factors which contribute to this include the distortion introduced by the initialization and the non-convexity of the optimization: when optimization is performed in a low dimensional space the model may not have the requisite degrees of freedom to avoid local minima.

In this paper we develop a prior on the dimensionality of the set of latent coordinates, encouraging low dimensional representations. Our *Rank Prior* enforces a penalty on the non-sparsity of the singular values of the matrix of latent variables, and automatically discovers the latent space and its dimensionality using a continuous optimization that drops dimensions on the fly. By initializing the

latent coordinates to the original space no distorsion is introduced; since we decrease the dimensionality slowly (starting from a high-dimensional space) extra flexibility is gained which allows the method to avoid local minima during optimization. To our knowledge, ours is the first non-linear dimensionality reduction technique that penalizes the latent space rank and simultaneously optimizes the structure of the latent space as well as its intrinsic dimensionality.

We demonstrate the effectiveness of our approach when learning probabilistic latent variable models with the Gaussian Process Latent Variable Model (GPLVM) [5], a generalization of Probabilistic PCA to the non-linear case that models the mapping from the latent space to the data space as a Gaussian process. The GPLVM has proven successful in many applications, but initialization, knowledge of the latent dimensionality, and/or additional prior knowledge were assumed (e.g., [2, 8, 11]). Incorporating our prior with the GPLVM objective results in an optimization problem that allows us to discover the latent space and the intrinsic dimensionality of artificial and real datasets. We further demostrate the effectiveness of our approach in tracking and classifying complex articulated human body motions from video.

2 Background: Gaussian Process Latent Variable Models

Latent Variable Models (LVMs), e.g., Probabilistic PCA [10] or MDS, assume that the data has been generated by some latent (unobserved) random variables that lie on or close to a low-dimensional manifold. Probabilistic LVMs relate the latent variables to a set of observed variables via a probabilistic mapping.

More formally, 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 $Q \ll D$. Let $y^{(d)} = f(\mathbf{x}) + \eta$ 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) [5] places a Gaussian process prior over the space of mapping functions f. Marginalizing over the functions f and assuming conditional independence of the output dimensions given the latent variables results in the GPLVM likelihood

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

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, $\Theta = \{\theta_1, \theta_2, \theta_3\}$, where θ_1 is the RBF lengthscale, θ_2 the kernel width, and θ_3 the observation noise. The latter contributes to a numerically stable inversion of the covariance $\mathbf{K}$.

Learning in the GPLVM 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, PCA [6] cannot capture non-linear dependencies , LLE [7] gives a good initialization *only* if the data points are uniformly sampled in the manifold, and Isomap [9] has difficulty with non-convex datasets [4]. Generally, when initialized far from the true minimum, the GPLVM optimization can get stuck in local minima [5, 11].

To avoid this problem different priors over the latent space have been developed. In [13] a prior was introduced in the form of a Gaussian process over the dynamics in the latent space. This results in smoother manifolds but performs poorly when learning stylistic variations of a motion or multiple motions [11]. Urtasun et al. [11] proposed 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. However, such prior knowledge is not commonly available, reducing considerably the applicability of their technique. In contrast, the method developed below introduces a generic prior that requires no specific prior knowledge, directly penalizing the dimensionality of the latent space to learn effective low-dimensional representations.

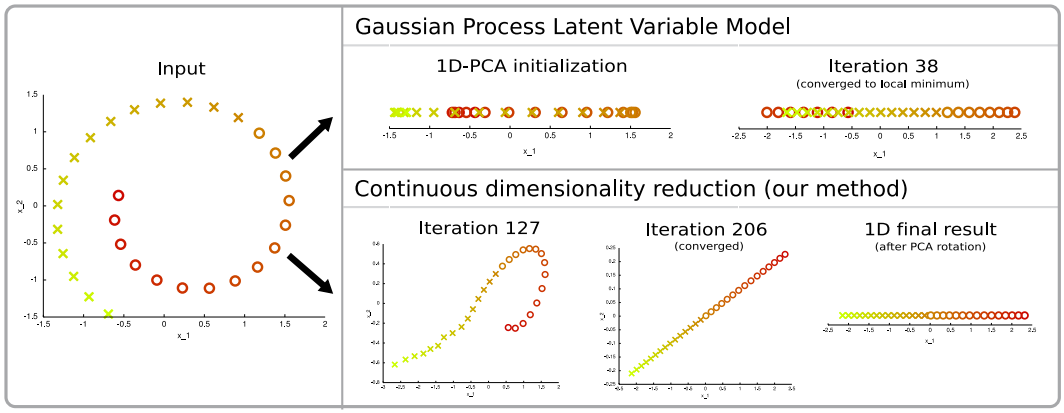


Figure 1: **Illustration of our Rank Prior with a GPLVM**: The goal is to recover the 1D manifold in a 2D space. The GPLVM gets stuck in local minima very early (upper row) since PCA initialization does not capture non-linear dependencies, whereas our method decreases dimensionality gradually and recovers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

3 Continuous Dimensionality Reduction via Rank Priors

We introduce a novel method for probabilistic non-linear manifold learning which avoids the initial distortion induced by ad-hoc initialization. We initialize the latent space to the high-dimensional observation space and define a *Rank Prior* which favors latent spaces with low dimensionality. Minimizing the dimensionality of the latent space is equivalent to minimizing $\mathcal{L}$, the rank of the Gram matrix of the rows of the unbiased matrix $\tilde{\mathbf{X}}$

$$\mathcal{L} = rank\left(\frac{1}{N-1}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right) \tag{1}$$

where $\tilde{\mathbf{X}}$ denotes the mean subtracted latent variables $\tilde{\mathbf{X}} = \mathbf{X} - \overline{\mathbf{X}}$ and $\overline{\mathbf{X}}_{ij} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{X}_{nj}$.

The cost function in Eq. (1) is discrete and thus difficult to minimize. Let the singular value decomposition (SVD) of $\tilde{\mathbf{X}}$ be $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are matrices containing the left and right singular vectors, and $\mathbf{\Sigma}$ comprises the singular values $\{\sigma_1(\tilde{\mathbf{X}}), ..., \sigma_D(\tilde{\mathbf{X}})\}$ on its diagonal. Then $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T$, and $\mathcal{L}$ can be minimized by minimizing the number of non-zero singular values of $\tilde{\mathbf{X}}$.

We transform the discrete optimization criteria in Eq. (1) into a continuous one by introducing a *sparsity penalty on the singular values*. In particular we introduce a prior of the form

$$p(\mathbf{X}) = \frac{1}{Z}\exp\left(-\sum_{m=1}^{D}\varphi(\hat{\sigma}_m(\tilde{\mathbf{X}}))\right) \tag{2}$$

where $\hat{\sigma}_m(\tilde{\mathbf{X}}) = \frac{1}{\sqrt{N-1}}\sigma_m(\tilde{\mathbf{X}})$, φ is a sparsity penalty function, and Z a normalization constant[1].

One can consider different sparsity penalty functions. The identity function $\varphi(\hat{\sigma}) = \hat{\sigma}$ results in the L_1-norm since the singular values are always positive. Of particular interest to us are functions that drive small singular values faster towards 0 than larger ones. Examples of such functions are the logarithmic $\varphi(\hat{\sigma}) = \alpha\ln(1+\beta\hat{\sigma}^2)$ and the sigmoid $\varphi(\hat{\sigma}) = \alpha(1+\exp(-\beta(\hat{\sigma}-\gamma)))^{-1}$ functions, with α, β and γ constant parameters.

Minimizing the negative log posterior results in an optimization that reduces the dimensionality in a continuous fashion:

$$\min_{\mathbf{X}}\left[\frac{D}{2}\ln|\mathbf{K}(\mathbf{X},\Theta)| + \frac{D}{2}tr(\mathbf{K}(\mathbf{X},\Theta)^{-1}\mathbf{Y}\mathbf{Y}^T) + \sum_{m=1}^{D}\varphi(\hat{\sigma}_m(\tilde{\mathbf{X}}))\right] \quad s.t.\ \Delta_E = 0 \tag{3}$$

[1]Note the fact that this is an improper prior has no impact in the optimization since it acts as a constant when minimizing the negative log posterior.

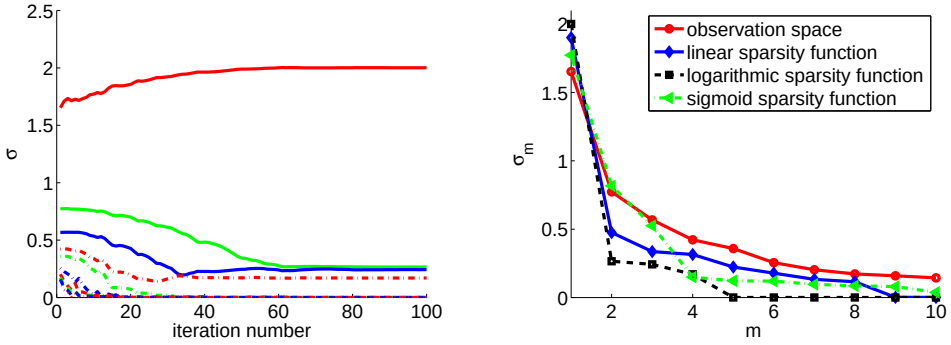


Figure 2: **Spectrum of a 30D motion database**: (a) Evolution of the first ten singular values as a function of the optimization iteration number for a logarithmic sparsity penalty function. (b) Spectrums learned after convergence by different sparsity penalty functions compared to the observation space spectrum (red).

where $\Delta_E = |E(\tilde{\mathbf{X}}) - E(\tilde{\mathbf{Y}})|$ is the difference between the energies of the spectrum of the mean subtracted latent coordinates $\tilde{\mathbf{X}}$ and the mean subtracted observations, $\tilde{\mathbf{Y}} = \mathbf{Y} - \bar{\mathbf{Y}}$, with $\bar{\mathbf{Y}}$ the mean of the observations. This constraint keeps the overall energy of the system constant. The energy of the spectrum can be defined as $E(\tilde{\mathbf{X}}) = \sum_{m=1}^{D} \hat{\sigma}_m^2(\tilde{\mathbf{X}})$.

The derivative of the rank prior with respect to the latent coordinates can be expressed as

$$\frac{\partial}{\partial X_{ij}} \sum_{m=1}^{D} \varphi(\hat{\sigma}_m) = \frac{1}{\sqrt{N-1}} \sum_{m=1}^{D} \frac{\partial \varphi(\hat{\sigma}_m)}{\partial \hat{\sigma}_m} U_{im} V_{jm} \ , \tag{4}$$

where $\frac{\partial \varphi(\hat{\sigma}_m)}{\partial \hat{\sigma}_m}$ depends on the sparsity function. The derivatives of the first two terms in Eq. (3) w.r.t. $\mathbf{X}$ and Θ are given in [5].

We use the SNOPT [1] non-linear constraint optimizer to minimize Eq. (3). After this optimization, we choose the latent dimension to be $Q = \text{argmax}_m \frac{\hat{\sigma}_m}{\hat{\sigma}_{m+1}+\epsilon}$, where $\epsilon \ll 1$, and $\hat{\sigma}_1 \geq \hat{\sigma}_2 \cdots \geq \hat{\sigma}_D$. The final steps consist of applying PCA in the optimized Q-dimensional space and optimizing $p(\mathbf{Y}|\mathbf{X}, \Theta)$ with respect to the hyperparameters Θ. 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 data to produce the most compact Q-dimensional representation.

Fig. 1 compares the GPLVM (initialized with PCA) with the result of optimizing Eq. (3) on a toy example where a 1D manifold is embedded in 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. 3 and 6.

Fig. 2 (a) depicts the evolution of the first ten singular values when optimizing Eq. (3) with a logarithmic sparsity penalty function for a motion database composed of 30D observations. Note how our method drops dimensions as the optimization evolves (i.e., the smallest singular values drop to zero within the first few iterations). A comparison of the spectrum of different sparsity penalty functions is shown in Fig. 2 (b). The L1-norm results in a poor estimation of the dimensionality, while the more aggressive sigmoid and logarithmic functions are able to recover the correct dimensionality in this example. In the remainder of the paper we use the logarithmic function since it converges faster than the L1-norm and has fewer parameters than the sigmoid.

4 Experimental results

In this section we demonstrate our approach in three different scenarios. We first compare our method to graph-based techniques and GPLVM with different initializations in artificial data. We illustrate our method's ability to estimate the latent space dimensionality in complex synthetic data.

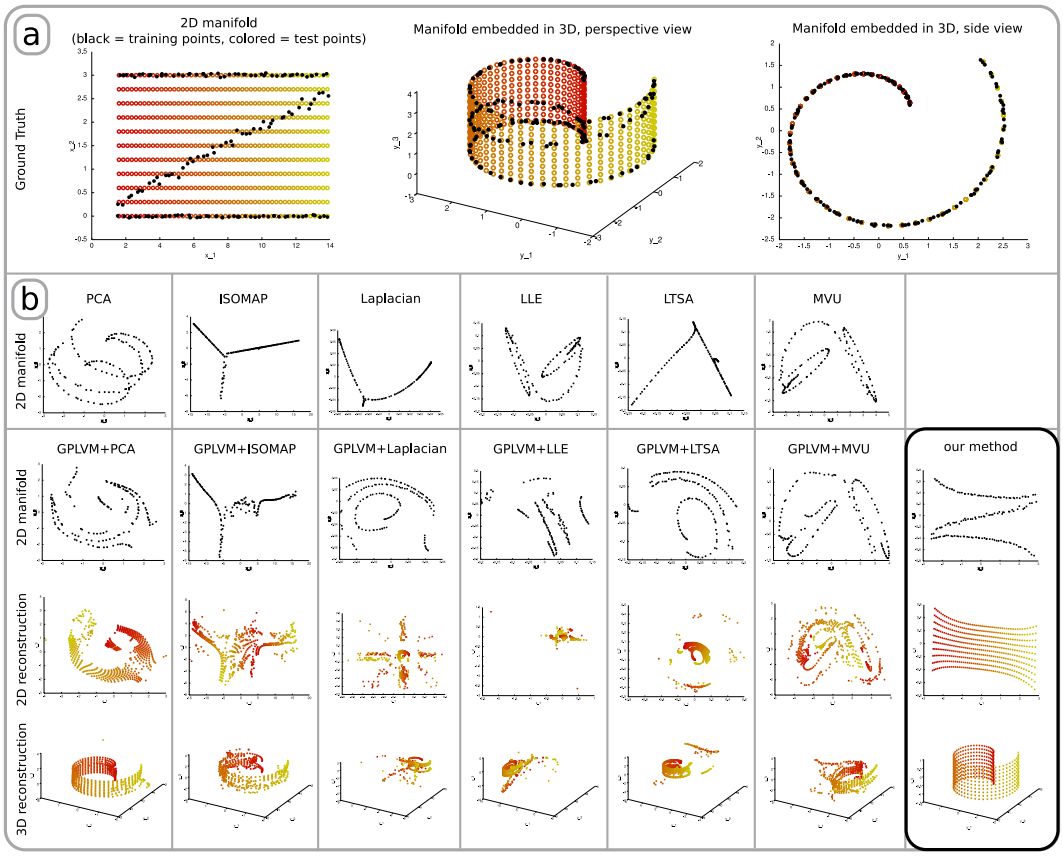


Figure 3: **Finding a 2D manifold in 3D space on a sparsely sampled swiss roll.** Only a sparse, noisy subset (depicted in black) of the full manifold is assumed to be known (a). (b) shows the initialization (with neighborhood size k=6), GPLVM result and 2D/3D reconstruction of the full manifold (from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

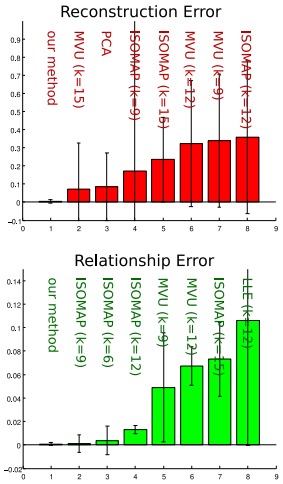
Finally we present an application of our technique to the challenging problem of tracking and classifying 3D articulated human body motion.

4.1 Experiment 1: Sparsely sampled swiss roll

The swiss roll is a widely used example of a 2D manifold which is embedded in a 3D space. Many state-of-the-art graph-based techniques, which rely on local neighborhoods, can be used to unravel it correctly if the data is homogenously sampled, the noise is small, and the neighborhood size is selected appropriately. However, real data often violates these assumptions resulting in poor performance.

We illustrate this problem by constructing a swiss roll which is sparsely sampled; only the black points in Fig 3 (a) are available for training. The first row in Fig. 3 (b) shows the result of applying PCA, Isomap, Laplacian Eigenmaps, LLE, LTSA and MVU (see [12] for a review on these techniques). The second row depicts our technique and the result of optimizing the GPLVM with different initializations. Finally the last two rows of Fig. 3 (b) show the test data (i.e., colored samples) reconstructed in the latent space and in the original space. Note that our method, unlike PCA, graph-based techniques and the GPLVM with any of the initializations, is able to recover the correct manifold.

We evaluate the performance of the different algorithms on this example computing 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



	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 4: **Quantitative performance on a synthetic sparse swiss roll example** Reconstruction and Relationship Error for the experiment in Fig. 3 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.

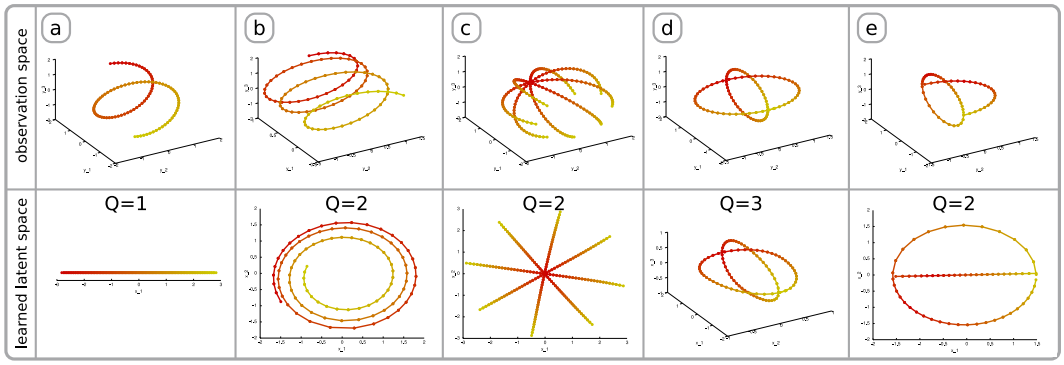


Figure 5: **Dimensionality estimation.** (Top) Five 2D manifolds embedded in 3D. (Bottom) Latent spaces and intrinsic dimensionalities Q learned using our continous dimensionality reduction method.

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 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 distance in the latent space and the distance in the observation space for two neighbors, and $\bar{\Gamma}_{i,j}$ is the mean ratio in the local neighborhood. Fig. 4 depicts these two error measures when performing the experiment in Fig. 3 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 $\alpha = 10$, $\beta = 10$, and $\Theta = \{0.5, 1.5, 0.01\}$. The hyperparameters were optimized for the GPLVM baselines. Note that our method outperforms the baselines independent of the initialization used for the GPLVM.

4.2 Experiment 2: Discovering the correct dimensionality

We illustrate our method's ability to discover the intrinsic dimensionality of the underlying manifold in 5 complex synthetic examples. In Fig. 5 (a) a spiral with a wide separation between rings is reduced to a 1D manifold. When the distance between the different rings decreases, the intrinsic manifold dimensionality changes from 1D to 2D (see Fig. 5 (b)), since relationships between points that have the same phase are considered. In Fig. 5 (c) the underlying 2D manifold from a cut-off sphere sampled along longitudinal lines is discovered. The manifold in Fig. 5 (d) is intrinsically 3D

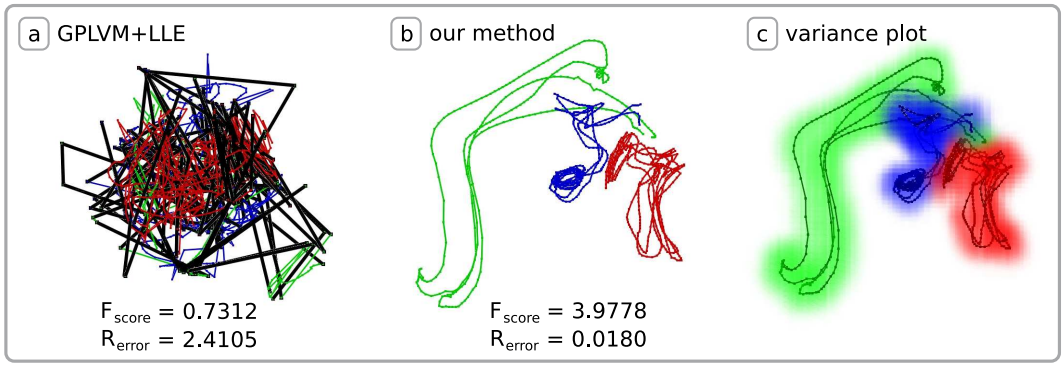


Figure 6: **Learning different types of motion into one single 3D latent space.** Each type is visualized with a different color (red = rolling, green = brooming, blue = milling). (a) depicts a 3D-GPLVM initialized to LLE, discontinuities are emphasized in black. (b,c) show the manifold learned by our method along with the variance.

and thus cannot be reduced (as discovered by our method), while the manifold in Fig. 5(e) can be reduced to 2D.

4.3 Experiment 3: Tracking and classification in the kitchen domain

An interesting real-world application of discovering low-dimensional structure in a high-dimensional space is tracking and classifying human motion from video sequences. Tracking consists of inferring the 3D locations of body joints from images. We use the kitchen dataset of [3], that consists of images from 2D cameras and ground truth joint angles of 3 motion types (i.e., rolling, milling, brooming) performed several times. We learned a single 3D latent space from 30D joint angle observations of 2 trials for each of the 3 different motions ($N = 1120$ training examples). Fig. 6 shows the result of learning such motions using (a) GPLVM initialized with LLE and (b) our method. For both methods we used sparsification to speed up learning. Note that our method, unlike the GPLVM initialized with LLE, is smooth (i.e., consecutive frames in time are close in latent space), and separates well the different classes. As shown in the figure, smoothness implies lower relationship error. To quantify the latter, we compute 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. Note that our method performs significantly better than GPLVM in terms of the relationship error and the fisher score.

Fig. 7 depicts tracking and classification performance for the milling and rolling motions[2] when using the models depicted by Fig. 6. We used a particle filter tracking that operates in the low dimensional space and models the dynamics with a first order Markov model. 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 NN for classification. Our method significantly outperforms the GPLVM with LLE initialization in both tracking accuracy and NN classification performance when using both, one or two cameras. We observed that the milling motion is more ambiguous than the rolling motion and thus can be tracked reliably only when using two cameras, whereas one camera proved sufficient for the rolling motion.

5 Conclusion

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 drops dimensions while solving for the latent coordinates. Our method can discover the dimensionality of the latent space and its intrinsic dimensionality, without the requirement of ad-hoc initialization. Our approach has proven superior to PCA, graph-based and non-linear dimensionality reduction techniques in a variety of synthetic and real-world databases in the task of dimensionality reduction, tracking and classifying articulated human motion. Our method is general and we believe it can be applied to any dimensionality reduction technique that can be expressed as the minimization of a cost function that is a function of the latent variables.

[2]No results are shown for brooming since no test data is available.

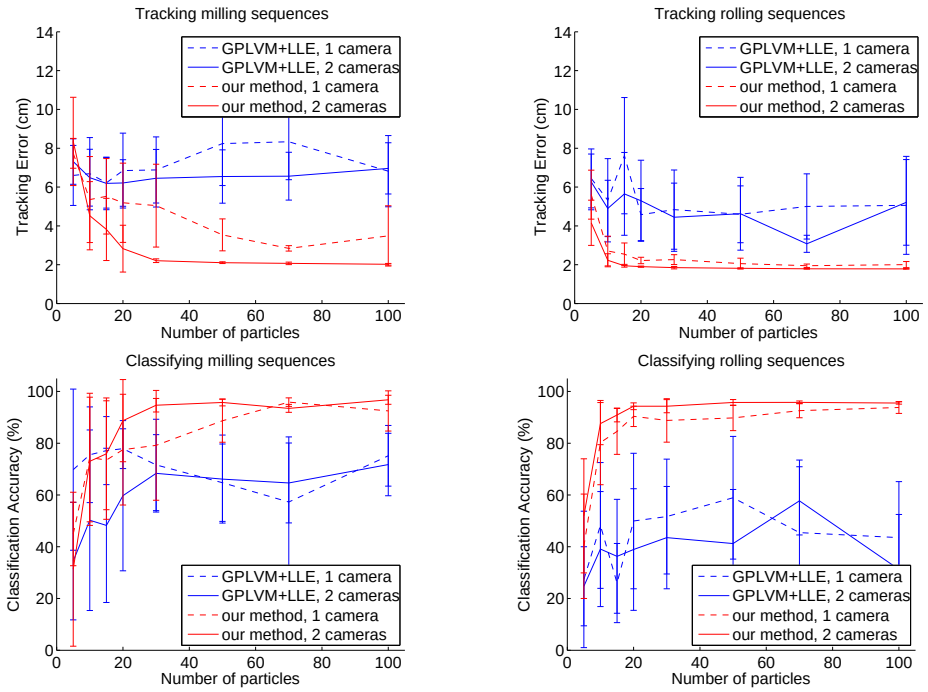


Figure 7: **Tracking and NN classification performance** for milling and rolling motions using our method (red) and GPLVM initialized to LLE (blue) as a function of the number of particles used in the particle filter.

References

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

[2] K. Grochow, S. Martin, A. Hertzmann, and Z. Popovic. Style-based inverse kinematics. In *Proceedings of SIGGRAPH 2004*. ACM Press / ACM SIGGRAPH, 2004.

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

[4] S. Harmeling. Exploring model selection techniques for nonlinear dimensionality reduction. Technical report, School of Informatics, Edinburgh University, Scotland, March 2007.

[5] N. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 2005.

[6] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

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

[8] M. Salzmann, R. Urtasun, and P. Fua. Local deformation models for monocular 3d shape recovery. In *Proceedings of the Conference in Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2008.

[9] J. B. Tenenbaum, V. d. Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.

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

[11] R. Urtasun, D. J. Fleet, A. Geiger, J. Popovic, T. Darrell, and L. N.D. Topologically-constrained latent variable models. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, 2008.

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

[13] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, 2008.