Towards Semi-supervised Manifold Learning: UKR with Structural Hints

Jan Steffen[(1)], Stefan Klanke[(2)], Sethu Vijayakumar[(2)] and Helge Ritter[(1)]

(1) Neuroinformatics Group, Bielefeld University, Germany
(2) Institute of Perception, Action and Behaviour, University of Edinburgh, UK

Abstract. We explore generic mechanisms to introduce *structural hints* into the method of Unsupervised Kernel Regression (UKR) in order to learn representations of data sequences in a semi-supervised way. These new extensions are targeted at representing a dextrous manipulation task. We thus evaluate the effectiveness of the proposed mechanisms on appropriate toy data that mimic the characteristics of the aimed manipulation task and thereby provide means for a systematic evaluation.

1 Introduction

Learning of control manifolds is emerging as one of the key challenges in unsupervised learning. Here, the Self-organising Map (SOM) has been influential in various pertinent approaches (cp. e.g.[1]). One more recent method, Unsupervised Kernel Regression (UKR, [6,4]), can be seen as a successor bridging between earlier "Parametrised SOM" (PSOM, [11]) and kernel methods (e.g.[8]).

In previous work [9], we have shown that UKR is well suited for representing human manipulation data. However, due to UKR being unable to incorporate prior knowledge about the data structure, generating *Manipulation Manifolds* (cp. [9]) from training sequences of hand posture data had been realised as supervised *construction* instead of automatic *learning*. In this paper, we present extensions to UKR for learning (periodic) sequences of chronologically ordered data and regularising intra-sequence characteristics which are aimed at learning *Manipulation Manifolds* in a *semi-supervised* manner. As basis for several error measures and thus a systematic evaluation of the new extensions, we perform an analysis on appropriate toy data which mimic the intrinsic characteristics of the targeted manipulation data. Whereas toy data always bare the risk of lacking transferability to the real data case, we here present promising first real data results in our targeted domain of dextrous manipulation.

We briefly recall UKR in Section 2 and present the new extensions in Section 3. Section 4 briefly summarises the original manipulation data and Section 5 addresses the corresponding toy data generation. Section 6 then uses this data for the evaluation of the new UKR extensions. Section 7 concludes the work.

2 Unsupervised Kernel Regression (UKR)

UKR is a recent approach to learning non-linear continuous manifolds, that is, finding a lower dimensional (latent) representation $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N) \in \mathbb{R}^{q \times N}$ of

a set of observed data $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N) \in \mathbb{R}^{d \times N}$ and a corresponding functional relationship $\mathbf{y} = \mathbf{f}(\mathbf{x})$. UKR has been introduced as the unsupervised counterpart of the Nadaraya-Watson kernel regression estimator by Meinecke et al. in [6]. Further development has lead to the inclusion of general loss functions, a landmark variant, and the generalisation to local polynomial regression [4]. In its basic form, UKR uses the Nadaraya-Watson estimator [7, 12]:

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{N} \mathbf{y}_i \frac{K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i)}{\sum_j K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_j)} \tag{1}$$

as smooth mapping $\mathbf{f} : \mathbf{x} \in \mathbb{R}^q \to \mathbf{y} \in \mathbb{R}^d$ from latent to observed data space ($K_{\mathbf{H}}$: Kernel with bandwidth $\mathbf{H}$). $\mathbf{X} = \{\mathbf{x}_i\}, i = 1..N$ now plays the role of input data to the regression function (1) and is treated as set of *latent parameters* corresponding to $\mathbf{Y}$. As the scaling and positioning of the $\mathbf{x}_i$'s are free, the formerly crucial bandwidths $\mathbf{H}$ become irrelevant and can be set to 1.

UKR training, that is finding optimal latent variables $\mathbf{X}$, involves gradient-based minimisation of the reconstruction error

$$R(\mathbf{X}) = \frac{1}{N} \sum_i \| \mathbf{y}_i - \mathbf{f}(\mathbf{x_i}; \mathbf{X}) \|^2 = \frac{1}{N} \| \mathbf{Y} - \mathbf{Y}\mathbf{B}(\mathbf{X}) \|_F^2 . \tag{2}$$

Here, $\mathbf{B}(\mathbf{X})$ with $(\mathbf{B}(\mathbf{X}))_{\mathbf{ij}} = \frac{K(\mathbf{x}_i - \mathbf{x}_j)}{\sum_k K(\mathbf{x}_k - \mathbf{x}_j)}$ is an $N \times N$ *basis function matrix.*

To avoid poor local minima, i.e. PCA [3] or Isomap [10] can be used for initialisation. These eigenvector-based methods are quite powerful in uncovering low-dimensional structures by themselves. Contrary to UKR, however, PCA is restricted to linear structures and Isomap provides no continuous mapping.

To avoid a trivial solution by moving the $\mathbf{x}_i$ infinitively apart from each other ($\mathbf{B}(\mathbf{X})$ becoming the identity matrix), several regularisation methods are possible [4]. Most notably, leave-one-out cross-validation (LOO-CV: reconstructing each $\mathbf{y}_i$ without using itself) is efficiently realised by zeroing the diagonal of $\mathbf{B}(\mathbf{X})$ before normalising its column sums to 1. The inverse mapping $\mathbf{x} = \mathbf{f}^{-1}(\mathbf{y}; \mathbf{X})$ can be approximated by $\mathbf{x}^\star = \mathbf{g}(\mathbf{y}; \mathbf{X}) = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{f}(\mathbf{x}; \mathbf{X})\|^2$.

3   UKR for data sequences

To enable the originally purely unsupervised UKR training to benefit from prior knowledge about the data structure, we introduce extensions which a) especially consider ordered data *sequences*, b) explicitly allow for *periodic* sequences, c) propagate the original intra-sequence order to their latent representations and d) propagate stability of non-temporal sequence parameters within the sequences.

a) We consider given affiliations to sequences which enables us to influence the latent parameter adaptation such that sequence-specific mechanisms can be involved in the training. To this end, we distinguish between one latent *temporal intra*-sequence dimension and the other *inter*-sequence *parameter* dimensions.

b) Periodic sequences consist of one periodic temporal and one/several (usually) non-periodic dimensions. To allow for such structure, we provide different univariate kernels K_l for different latent dimensions l. The basis functions

$(\mathbf{B}(\mathbf{X}))_{\mathbf{ij}}$ (cp. Sec.2) then consist of their normalised products (parametrised by $\mathbf{\Theta}_l$):

$$(\mathbf{B}(\mathbf{X}))_{\mathbf{ij}} = \frac{\prod_{l=1}^{q} K_l(x_{i,l} - x_{j,l}; \mathbf{\Theta}_l)}{\sum_{k}^{N} \prod_{l=1}^{q} K_l(x_{k,l} - x_{j,l}; \mathbf{\Theta}_l)}. \tag{3}$$

In the non-periodic case, the univariate versions of the kernels used in original UKR can be applied (e.g. Gaussian: $K_g(x_i - x_j; \Theta) = \exp\left[-\frac{1}{2}\Theta^2(x_i - x_j)^2\right]$). In analogy to original UKR, we assume no need for bandwidth control. However, to analyse potential cross-effects with the following new extensions, we also investigate different bandwidths for this case. For the periodic case, we propose the following cyclic kernel with bandwidth parameter Θ, periodic in $[0; \pi]$:

$$K_{\circlearrowleft}(x_i - x_j; \Theta) = \exp\left[-\frac{1}{2}\Theta^2 \sin^2(x_i - x_j)\right]. \tag{4}$$

Up to normalisation and scaling, the kernel is equivalent to the von Mises distribution [5] which has been already used by Bishop et al. [2] to represent periodic data characteristics. We chose the presented form for convenience reasons.

In the periodic case, kernel bandwidth regulation is needed since the effective space in corresponding dimensions is constrained due to its periodic nature and fixed bandwidths cannot be compensated by scaling the latent parameters.

c) "cyclic data order": To propagate the original chronological order of N_S data sequences $\mathcal{S}_\sigma = (\mathbf{y}_1^\sigma, .., \mathbf{y}_{N_\sigma}^\sigma)$, $\sigma = 1..N_S$ to the corresponding latent parameters $(\mathbf{x}_1^\sigma, .., \mathbf{x}_{N_\sigma}^\sigma)$, the values x_{i,d_t}^σ, $i = 1..N_\sigma$ in the *temporal* latent dimension d_t need to reflect the order of the original data sequence. In the periodic case, such condition is difficult to induce without any assumptions about the underlying sequences. However, by providing sequences of complete cycles, we can consider the first data point in the sequence as successor of the last one: $\mathbf{x}_0^\sigma = \mathbf{x}_{N_\sigma}^\sigma$. If so, a penalty term in the loss function can preserve the cyclic data order:

$$E_{cseq}(\mathbf{X}) = \sum_{\sigma=1}^{N_S} \sum_{i=1}^{N_\sigma} \sin^2(x_{i,d_t}^\sigma - x_{(i-1),d_t}^\sigma). \tag{5}$$

d) One strong assumption which we want to be reflected in the latent space is, that the values of the non-temporal dimensions are approximately constant within single sequences. This consideration stems from the generation of our manipulation data (see next Section for a short description). The basic idea is that the underlying movement parameters usually do not change during single sequences – e.g., for cap turning, the radius of the cap does not change during the turning. We realise this regularisation of intra-sequence parameter variations as penalty term to the loss function which penalises high variances in the non-temporal dimensions $k = 1..q$, $k \neq d_t$:

$$E_{pvar}(\mathbf{X}) = \sum_{\sigma=1}^{N_S} \sum_{k \neq d_t} \frac{1}{N_\sigma} \sum_{i=1}^{N_\sigma} \left(x_{i,k}^\sigma - \langle x_{\cdot,k}^\sigma \rangle\right)^2 \tag{6}$$

The overall loss function then can be denoted as $E(\mathbf{X}) = R(\mathbf{X}) + \lambda_{cseq}E_{cseq}(\mathbf{X}) + \lambda_{pvar}E_{pvar}(\mathbf{X})$. The new parameters are $(\Theta_1, \ldots, \Theta_q, \lambda_{cseq}, \lambda_{pvar})$.

Fig. 1. Example of a hand posture sequence corresponding to a training manipulation of a bottle cap ($r = 2.0cm$). Note the periodic nature of the movement.

4 Manipulation Data

As described in Sec.1, the presented extensions are aimed at learning the representation of a manipulation task (i.e. turning a bottle cap). The set of training data, which has been used already for the initial manifold *construction* in [9], consists of sequences of hand postures (each a 24D joint angle vectors) recorded during the turning movement for different cap radii ($r = 1.5cm$, $2.0cm$, $2.5cm$, $3.0cm$ and $3.5cm$). The movement itself is periodic in the sense that the beginning and end postures are (in principle) the same. For each radius, we produced five to nine sequences of about 30 to 45 hand postures each – in total 1204 for all sequences and all radii. Figure 1 exemplary visualises one of such sequences.

5 Toy Data for Evaluation

To evaluate the new UKR extensions, we generate toy data with similar intrinsic characteristics as the manipulation data in [9] briefly described in the last section. The utilisation of toy data provides us with knowledge about underlying true structures and enables us to compute a variety of error measures not accessible otherwise (cp. Sec. 6 for details). As basis for an adequate toy data generation, we thoroughly investigate the real data. Here, we especially try to uncover the intrinsic data structures reflecting our prior knowledge of the generated manipulation data. From the generation process, we assume the existence of a periodic structure reflecting the periodic nature of the cap turning movement and an additional non-periodic expansion reflecting the different cap radii used for the sequence generation. By using Isomap [10] – a powerful method for uncovering hidden intrinsic structures in large data sets – we are able to reinforce these assumptions: a three-dimensional Isomap embedding of our manipulation data (see Fig.2a) reveals a cylinder-like structure describing a periodicity living in the x/y dimensions and a non-periodic extension in z direction.

To unfold the 2D representation of the periodicity, we can apply $atan2$ on the x/y-part of the embedding data yielding the basis for the corresponding 1D "angle" $\in [0; \pi]$. In combination with the original z component, we receive a 2D representation of the formerly 3D Isomap embedding and of the 24D original hand posture data, respectively. This data can be used as latent initialisation of the UKR model[1] as visualised in Fig.2b. Here, it turns out that the different sequences (connected) are not clearly separated and even sequences corresponding to different cap radii (encoded by different colours) partly overlap.

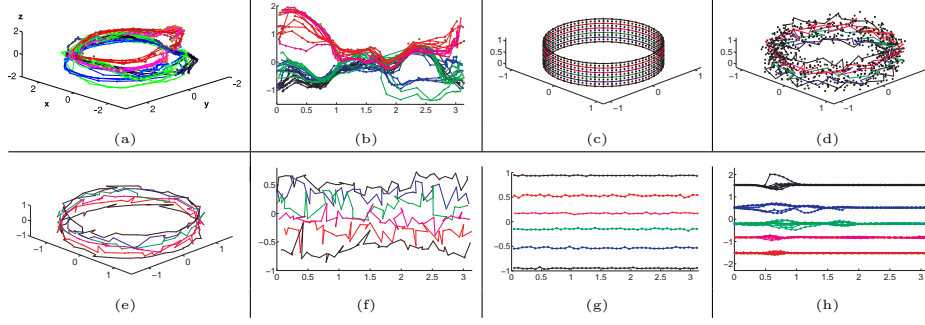[1] The 2D latent space with one periodic kernel has the topology of a cylinder surface.

Fig. 2. **(a)** 3D Isomap embedding of 24D hand posture data recorded during the turning movement of a bottle cap. Different colours encode different cap radii. **(b)** atan2-mapping of (a). **(c)** noise-free training data (red, connected); test data (black, single points). **(d)** noisy training/test data. **(e)** Toy data Isomap embedding (cp. (a)). **(f)** atan2-mapping of (e). **(g-h)** Results for toy (g) and real (h) data.

To reflect similar characteristics in our toy data and to provide an informative basis for the later evaluation, we aim at a simple low-dimensional toy data structure that produces Isomap embeddings of a similar form as the real data. To this end, we generate ordered (connected) data samples from the surface of a cylinder geometry (height=1, radius=1, Fig.2c) living in 3D together with noisy versions (Gaussian noise, $\sigma = 0.1$, e.g. Fig.2d). Such data then yield Isomap embeddings which a) provide a periodicity b) a non-periodic *parameter* expansion and c) are organised in *chronologically* ordered sequences ("trials") and thus are quantitatively similar to the Isomap embedding of the real data (Fig.2a/e) and its 2D mapping (Fig.2b/f). Within this cylinder structure, cross sectional rings of different height levels model sequences for different cap radii in the real data. As basis for the evaluation, we generated six training data rings and six overlapping together with five intermediate test data rings (cf. Fig.2c).

In anticipation of the following, Fig.2(g-h) depict the resulting latent parameters from training with toy and real data, respectively, having considered the results from the next section. The similarity of both latent structures supports the appropriateness of the toy data for the use with our real manipulation data.

6 Evaluation and Results

We evaluate the new extensions to UKR with the training/test data described in the last section. We incorporate our prior knowledge about the data – periodic sequences and non-periodic height levels corresponding to the periodic movement and the non-periodic radii variation in the manipulation data – in form of the specification of two associated latent dimensions: one periodic ($K_1(\cdot; \Theta_1) = K_{\circlearrowleft}$) *temporal* and one non-periodic ($K_2(\cdot; \Theta_2) = K_g$) *parameter* dimension. The loss function then consists of the reconstruction error and the penalty terms introduced in Section 3: $E(\mathbf{X}) = R(\mathbf{X}) + \lambda_{cseq} \cdot E_{cseq}(\mathbf{X}) + \lambda_{pvar} \cdot E_{pvar}(\mathbf{X})$.
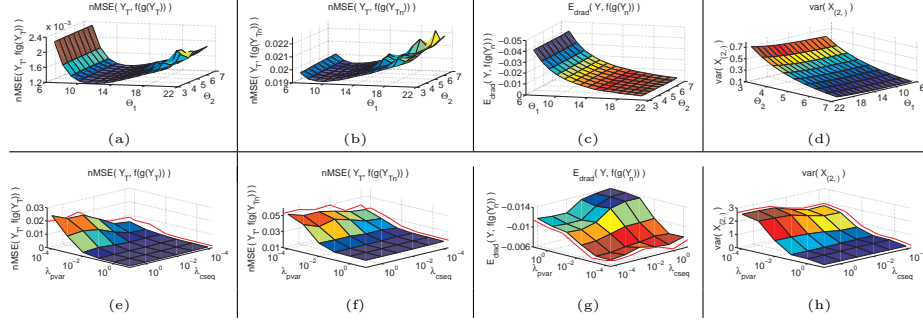
Fig. 3. Evaluation results **(a-d)** for $\lambda_{cseq} = \lambda_{pvar} = 1$ and **(e-h)**$(\Theta_1, \Theta_2) = (12, 4)$, red lines:$\lambda_{cseq}/\lambda_{pvar} = 0$. Please refer to text for further explanation.

As proposed in the last section, we compute 3D Isomap embeddings (for this data very robust in the choice of the neighbourhood parameter; here $K = 10$) of the noisy training data $\mathbf{Y}$ (cf. Fig.2e), and again use atan2 to retrieve a 2D latent initialisation for the UKR model (Fig.2f).

The evaluation focusses on the effect of different combinations of the inverse bandwidths Θ_1, Θ_2, and the penalty weightings $\lambda_{cseq}, \lambda_{pvar}$. From our toy data structure, we derive initial guesses for good bandwidth parameters ($\Theta_1 = 14, \Theta_2 = 5$ based on average inter-point distances) and evaluate correspondingly Θ_1 for values $\{7, 8, .., 14, .., 21\}$ and Θ_2 for $\{3, 3.5, .., 5, .., 7\}$. As for λ_{cseq} and λ_{pvar}, no assumptions could be made, we choose $\lambda_{cseq}, \lambda_{pvar} \in \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1\}$. For each tuple $(\Theta_1, \Theta_2, \lambda_{cseq}, \lambda_{pvar})$, 10 training runs with 10 noisy versions of the training data are conducted. Each run consists of 500 optimisation steps including LOO-CV (exemplary result: Fig.2(g)). Initial tests yielded the most promising results for $\lambda_{cseq} = \lambda_{pvar} = 1$ which thus provides a good starting point for the evaluation of Θ_1 and Θ_2.

Fig.3(a-d) depict the corresponding reconstruction errors for varying bandwidth parameters Θ_1, Θ_2. Fig.3a shows the normalised mean square error (nMSE) between noise-free test data $\mathbf{Y}_T$ (the underlying true cylinder geometry) and its UKR reconstructions $\mathbf{f}(\mathbf{g}(\mathbf{Y}_T))$, visualising UKR's ability to generalise to unseen data from the underlying structure. Fig.3b shows the nMSE between $\mathbf{Y}_T$ and the reconstruction of its noisy versions $\mathbf{f}(\mathbf{g}(\mathbf{Y_{Tn}}))$, visualising UKR's robustness in representing the underlying structure and its ability to correct noisy input data. The bias of $\mathbf{f}(\mathbf{g}(\cdot))$ towards the inner of the underlying structure (a known problem in original UKR) is depicted in Fig.3c for noisy training data $\mathbf{Y}_n$.

Fig.3(a-b) show a clear error dependency on Θ_1 and minimal errors for $\Theta_1 = 12$ (Fig.3a) or $\Theta_1 = 10$ (Fig.3b). However, as the bias significantly increases with decreasing Θ_1 (Fig.3c), we use $\Theta_1 = 12$ for further evaluation. As assumed before, there is no significant dependency on Θ_2 due to the free positioning of the latent parameters in the non-periodic dimensions. This is shown in Fig.3: whereas the errors stay approximately constant (Fig.3(a-c)), the variance in the latent *parameter* dimension varies strongly for changing Θ_2 (Fig.3(d)).
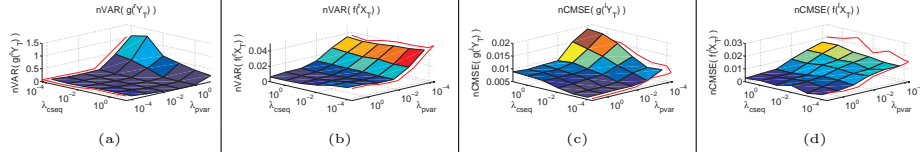
Fig. 4. Application err's $(\Theta_1, \Theta_2) = (12, 4)$, Red line: $\lambda_{cseq}/\lambda_{pvar} = 0$ (details: see text).

Fig.3(e-h) depict errors for fixed bandwidth parameters $(\Theta_1, \Theta_2) = (12, 4)$ and different combinations of $\lambda_{cseq}, \lambda_{pvar}$. Fig.3(e-f) reveal that high values of λ_{cseq} – which stronger force correctly ordered latent parameters – negatively influence the reconstruction error. However, high values of λ_{pvar} damp the reconstruction error in general and are able to overrule the negative effect of the sequence order penalty. Indeed, as depicted in Fig.3g, both high weightings of E_{cseq} and E_{pvar} yield high radius errors. Logically consistent, high values of λ_{pvar} strongly damp the variance in the latent data dimension (cp. Fig.3h).

For applications exploiting the aimed sequence-reflecting latent structure, not only the pointwise nMSE, but also structure-related errors are of interest. Fig.4a shows a normalised variance in the latent parameter dimension ("nVAR") of observed fix-parameter sequences (lines in observed space) mapped into latent space $(\mathbf{g}(^\mathbf{r}\mathbf{Y_T}))$ providing a measure for the distortion of the line projection and thus for the distortion in the parameter dimension. The plot uncovers that high weightings of E_{pvar} (reducing general reconstruction errors; cp.Fig.3(e-h)) only result in stable sequence projections for strongly weighted E_{cseq}. Fig.4b shows the inverse projection direction, corresponding to reproducing/synthesising sequences in original data space with fixed sequence parameters. Again, only combined high E_{pvar}- and E_{cseq}-weightings produce stable sequences. Fig.4(c-d) investigate the corresponding inverse situations. Fig.4c visualises temporal synchronisation distortions of the latent space projections of sequence parameter modulations in observed space for fixed points in time. To take account for the periodic nature of the latent temporal dimension, we calculate nMSEs on the angular deviations from the mean ("nCMSE") of the analysed line and take the underlying kernel period into account. Like this, the nCMSE has similar characteristics as the nVAR for non-periodic dimensions. Here, high λ_{cseq}-weightings result in higher distortions of the projections. However, for the targeted high weightings of E_{pvar}, the negative effect of higher values for λ_{cseq} still is in a reasonable region. Fig.4d visualises the inverse mapping, measuring the distortions of projections of lines in latent space with zero-variance in the temporal dimension back into observed space. Again, for high λ_{pvar}, the effect of the sequence penalty E_{cseq} is strongly dominated by the effect of E_{pvar}.

To sum up: whereas the choice of Θ_2 is less important, the inverse bandwidth Θ_1 should be set to a value (slightly) smaller than the inverse of the average point distance in the corresponding dimension (here: $\Theta_1 = 12$). The results for varying $\lambda_{cseq}/\lambda_{pvar}$ are relatively robust. Here, they optimally effect the generation of the desired latent structures for $\lambda_{cseq}/\lambda_{pvar} = 1$.
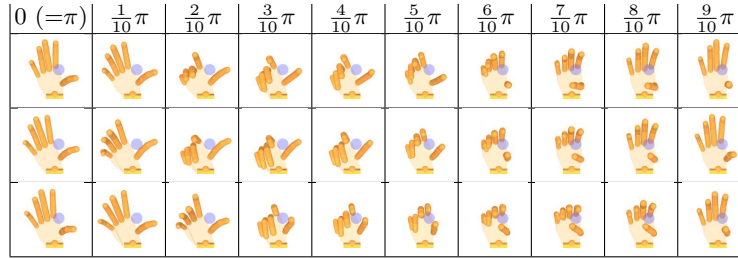
$0 \ (=\pi)$	$\frac{1}{10}\pi$	$\frac{2}{10}\pi$	$\frac{3}{10}\pi$	$\frac{4}{10}\pi$	$\frac{5}{10}\pi$	$\frac{6}{10}\pi$	$\frac{7}{10}\pi$	$\frac{8}{10}\pi$	$\frac{9}{10}\pi$

Fig. 5. Promising results for the targeted manipulation task. Horizontal dim.: time; vertical dim.: cap radius. Depicted are reprojections $\mathbf{f}(\mathbf{x};\mathbf{X})$ of regularly sampled latent positions $\mathbf{x}$ of the trained UKR. Please consider also the video available under `http://www.techfak.uni-bielefeld.de/∼jsteffen/mov/wsom2009/`.

7 Conclusion

We presented extensions to the unsupervised manifold learning method UKR, which now allow for semi-supervised learning of structured manifolds. We evaluated the new extensions on toy data in a general and manipulation relevant context as basis for future work on real manipulation data. First promising results using our new insights are visualised in Fig. 5: the targeted task of representing the periodic movement of turning a bottle cap has been successfully achieved.

References

1. G. Barreto, A. Araújo, and H.Ritter. Self-Organizing Feature Maps for Modeling and Control of Robotic Manipulators. *Intelligent and Robotic Systems*, 30(4), 2003.
2. C. Bishop and C. Legleye. Estimating conditional probability densities for periodic variables. *Advances in Neural Information Processing Systems*, 7:641–648, 1995.
3. I.T. Jolliffe. *Principal Component Analysis*. Springer, New York, 2nd edition, 2002.
4. S. Klanke. *Learning Manifolds with the Parametrized Self-Organizing Map and Unsupervised Kernel Regression*. PhD thesis, Bielefeld University, 2007.
5. K. Mardia. Statistics of Directional Data. *Academic Press, London*, 1972.
6. P. Meinicke, S. Klanke, R. Memisevic, and H. Ritter. Principal Surfaces from Unsupervised Kernel Regression. *IEEE Trans. on PAMI*, 27(9), 2005.
7. E. A. Nadaraya. On Estimating Regression. *Theory of Probability and Its Application, Vol.9*, 1964.
8. B. Schölkopf, A. Smola, and K. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5):1299–1319, 1998.
9. J. Steffen, R. Haschke, and H. Ritter. Towards Dextrous Manipulation Using Manifolds. In *Proc. IROS*, 2008.
10. J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, Dec. 2000.
11. J. Walter and H. Ritter. Rapid Learning with Parametrized Self-organizing Maps. *Neurocomputing*, 12:131–153, 1996.
12. G. S. Watson. Smooth Regression Analysis. *Sankhya, Ser.A*, 26, 1964.