A Manifold Representation as Common Basis for Action Production and Recognition

Jan Steffen, Michael Pardowitz, and Helge Ritter

Neuroinformatics Group, Faculty of Technology, Bielefeld University, Germany
email: {jsteffen,mpardowi,helge}@techfak.uni-bielefeld.de

Abstract. In this paper, we first review our work in the domain of dextrous manipulation, where we introduced *Manipulation Manifolds* – a highly structured manifold representation of hand postures which lends itself to simple and robust manipulation control schemes.
Coming from this scenario, we then present our idea of how this generative system can be naturally extended to the recognition and segmentation of the represented movements providing the core representation for a combined system for action production and recognition.

1 Introduction

In the field of humanoid robotics, two of the key challenges are the production of naturally looking movements on the one hand and the recognition of observed movements or their segmentation into several meaningful subparts on the other hand. Whereas these two problems are usually addressed independently from each other, we believe that they are indeed closely related and that is beneficial to base their handling on one and the same core representation of the underlying – observed or produced – movements. Whereas such common basis for action and perception could not be established yet in the field of robotics, it is widely known from neurophysiology where research on monkey brains reports from mirror neurons in the premotor cortex which not only show activity during the monkey's own excitations but as well during observations of the same actions performed by another monkey (e.g. [7]).

With our work, we approached this problem from the motion production side using motion capture data recorded from human demonstration. In this domain, many recent approaches focus on the Gaussian Processes Latent Variable Model (GPLVM, [4]) and variants. For example, Bitzer et al. [1] propose a methodology for learning and synthesising classes of movements using the GPLVM and identify robust latent space control policies which allow for generating novel movements. In another approach, Urtasun et al. [12] extend the GPLVM in order to learn interpretable latent directions and transitions between motion styles.

Whereas such approaches yield very promising results for reproducing and synthesising motion capture data, it is not clear how they can be extended for motion recognition. For our work, we were thus looking for a method that enables us to directly reinforce a clear and predefined structure of the latent variables

Final version in: Proc. of the 32nd German Conference on Artificial Intelligence (KI-2009)

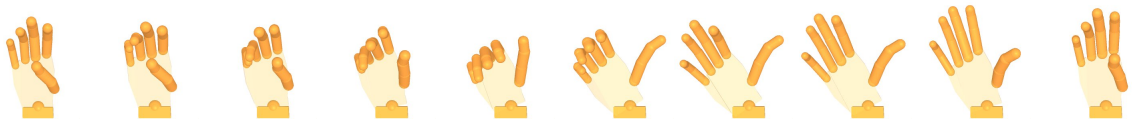

Fig. 1. Example of a hand posture sequence corresponding to a training manipulation of a bottle cap ($r = 2.0cm$). Remark the periodic nature of the movement.

which then lends itself to simple and robust control schemes. The knowledge about this clear structure then can be exploited as well for the use as recognition system afterwards. In the context of manifold generation, we presented modifications to a recent approach to non-linear manifold learning, namely *Unsupervised Kernel Regression* (UKR, [5]), which either allow for directly incorporating prior knowledge in a constructive manner [8] or in an automatic learning scenario [9]. As shown in [8, 9], the resulting *Manipulation Manifolds* then provide the desired highly structured latent spaces and can be used as basis for reproduction and synthesis of the represented movement class.

In this paper, we present our idea of how this generative system can be naturally extended to the recognition of the represented movements.

The paper is organised as follows: After a description of the training data (Sec. 2), we briefly review UKR (Sec.3) and the two methods to generate the *Manipulation Manifolds* (Sec. 4 and Sec. 5). Section 6 then describes the manifold characteristics which are exploited in Section 7 for the motion production. Section 8 finally presents our idea of a recognition system based on this representation followed by a conclusion in Section 9.

2 Manipulation Data

The training data consist of sequences of hand postures (each a 24D joint angle vectors) recorded with a data glove during cap turning movements for different cap radii ($r = 1.5cm,\ 2.0cm,\ 2.5cm,\ 3.0cm$ and $3.5cm$) in a physics-based computer simulation (e.g. Fig. 1). For each radius, we produced five to nine sequences of about 30 to 45 hand postures each – in total 1204 for all sequences.

3 Unsupervised Kernel Regression

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 [5]. Further development has lead to the inclusion of general loss functions, a landmark variant, and the generalisation to local polynomial regression [3]. In its basic form, UKR uses the Nadaraya-Watson estimator [6, 13] as smooth map-

ping from latent to observed data space ($K_{\mathbf{H}}$: Kernel with bandwidth $\mathbf{H}$):

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

$\mathbf{X} = \{\mathbf{x}_i\}, i = 1..N$ 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:

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

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 [2] or Isomap [11] 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 [3]. 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 $\mathbf{x}^\star = \mathbf{g}(\mathbf{y}; \mathbf{X}) = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{f}(\mathbf{x}; \mathbf{X})\|^2$.

4 Manifold construction

A simple but effective approach to generating a *Manipulation Manifold* is to construct the final manifold out of several sub-manifolds each realising a manipulation movement for one motion parameter (cp. [8]). In the example of turning a bottle cap, we incorporate the progress in time of the movement and the radius of the cap. The construction of the final manifold is performed iteratively starting with a sequence associated with the smallest radius. The latent parameters of the first 1D-UKR manifold are equidistantly distributed in a predefined interval according to the chronological order of the hand postures (Fig.2(a)). The second sequence of the same radius then is projected pointwise into the latent space of the previous 1D-manifold. By dint of this projection, we approximate a synchronisation of the temporal advance of the two movements. In the next step, we combine those data to a new UKR manifold by extending the sets of observed data and latent parameters by the new sequence data (cp. Sec.3: $\mathbf{Y}$ and $\mathbf{X}$).

Repeating this step for all sequences of one radius yields a radius-specific 1D manifold representing a generalised movement. Thus, by applying this method to all sets of radius-specific sequences, we generate one 1D manifold per radius. To promote the synchronisation of the temporal advances also between the different

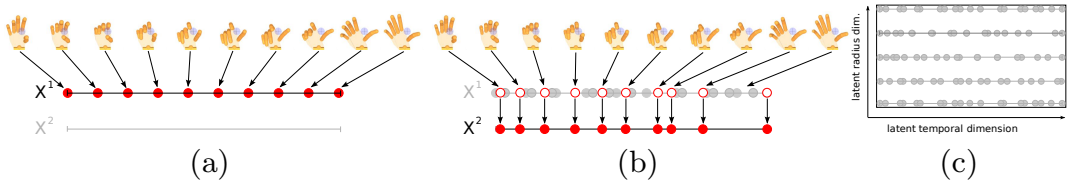

Fig. 2. Schematic description of different steps in the manifold construction process.

radius-specific manifolds, we initialise the manifolds of new radii with the projection of the observed data into the latent space of the previous radius (cp. 2(b)). The subsequent combination of all 1D-manifolds to one 2D-manifold representing the complete movements for all training radii then is realised by expanding each 1D latent parameter by a second dimension denoting the appropriate radius corresponding to the associated training sequence (2(c)).

Using the radius information automatically results in the correct ordering of the latent parameters in the new dimension. Whereas this last step always requires meta knowledge about the training data, at the same time, it provides a simple and effective way of incorporating prior knowledge into the manifold generation procedure. Another benefit of directly using observed or predefined meta data (like the cap size) as values of latent parameter dimensions is that new data recorded after the initial training can directly be added to the manifold in the same way and then serve to locally refine the manifold structure.

5 Manifold learning

In several cases, it is desirable to automatically learn the *Manipulation Manifolds* from training sequences instead of construct them in the described way. We thus presented extensions to original UKR training to provide implicit mechanisms for incorporating given knowledge about training data structures [10].

In addition, to take the periodic nature of the data sequences into account, we extended original UKR by allowing for different univariate kernels K_l (with dimension-specific parameters Θ_l) for different latent dimensions l (cp. Eq.2):

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

As kernel for the non-periodic dimensions, we use a standard Gaussian kernel with (inverse) bandwidth parameter Θ: $K_g(x_i - x_j; \Theta) = \exp\left[-\frac{1}{2}\Theta^2(x_i - x_j)^2\right]$, and for the periodic dimensions, we proposed a sin^2 kernel, periodic in $[0; \pi]$, again with parameter Θ: $K_\circlearrowleft(x_i - x_j; \Theta) = \exp\left[-\frac{1}{2}\Theta^2 \sin^2(x_i - x_j)\right]$.

The two key features provided by the construction described in the last section are that (a) the chronological order of the training sequences is reflected in the corresponding latent variables and (b) the latent representations of the training sequences have constant values in the latent radius dimension (assuming that the underlying movement parameters do not change within the sequences.)

In the automatic learning case, we approximate these two features by means of penalty terms to the standard loss function (Eq.2):

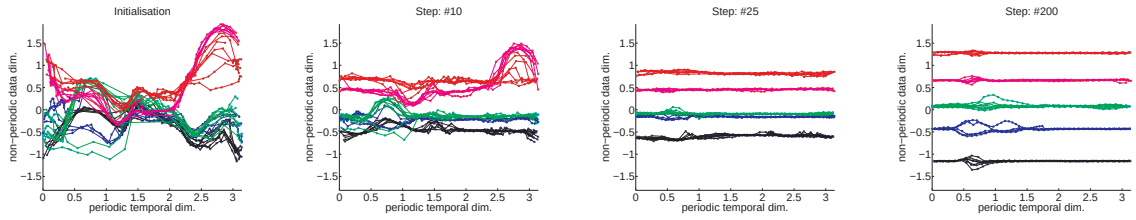

Fig. 3. Development of UKR latent variables after 0, 10, 25, and 200 steps. Connected points represent training sequences; different colours correspond to different cap radii.

(a) In the periodic case using *closed* sequences of training data ($\mathbf{x}_0^\sigma = \mathbf{x}_{N_\sigma}^\sigma$), we can express this as regularisation of the sum of successor distances:

$$E_{cseq}(\mathbf{X}) = \sum_{\sigma=1}^{N_\mathcal{S}} \sum_{i=1}^{N_\sigma} \sin^2(x_{i,d_t}^\sigma - x_{(i-1),d_t}^\sigma). \tag{4}$$

for sequences $\mathcal{S}_\sigma = (\mathbf{y}_1^\sigma, \mathbf{y}_2^\sigma, \ldots, \mathbf{y}_{N_\sigma}^\sigma)$, $\sigma = 1..N_\mathcal{S}$ with corresponding latent parameters $(\mathbf{x}_1^\sigma, \mathbf{x}_2^\sigma, \ldots, \mathbf{x}_{N_\sigma}^\sigma)$. d_t denotes the latent time dimension.

(b) is realised by penalising high variances in the parameter dimensions $k \neq d_t$:

$$E_{pvar}(\mathbf{X}) = \sum_{\sigma=1}^{N_\mathcal{S}} \sum_{k \neq d_t} \frac{1}{N_\sigma} \sum_{i=1}^{N_\sigma} \left(x_{i,k}^\sigma - \langle x_{\cdot,k}^\sigma \rangle\right)^2 \tag{5}$$

The resulting overall loss function then can be denoted as:

$$E(\mathbf{X}) = R(\mathbf{X}) + \lambda_{cseq} \cdot E_{cseq}(\mathbf{X}) + \lambda_{pvar} \cdot E_{pvar}(\mathbf{X}). \tag{6}$$

Fig.3 visualises an exemplary development of the latent variables using this method and the training data described in Sec.2. For further details see [10, 9].

6 Characteristics of the Manipulation Manifold

Figure 4 visualises an exemplary *Manipulation Manifold*. As result of the learning (or construction), the horizontal dimension corresponds to the temporal aspect of the movement and the vertical dimension describes the cap size as motion parameter (please consider the video referenced in Fig. 4). Like this, it forms a representation of the movement of turning a bottle cap for different cap sizes that fulfils our goal of fitting to the desired simple control strategy: the represented movement can be produced by projecting a linear trajectory that follows the time dimension in latent space into hand posture space.

This characteristics is realised by distorting the natural topology of the latent space such that those parts of the movement which are independent of the cap radius – and thus very similar for all radii (i.e. Fig. 4, the backward movement of the hand: columns 1-3) – are pushed away from each other to span the same latent radius range as the rest of the sequence. In contrast to this, with purely unsupervised learning, the similar parts would collapse to thin regions in latent space. This however would make the targeted control scheme impossible.

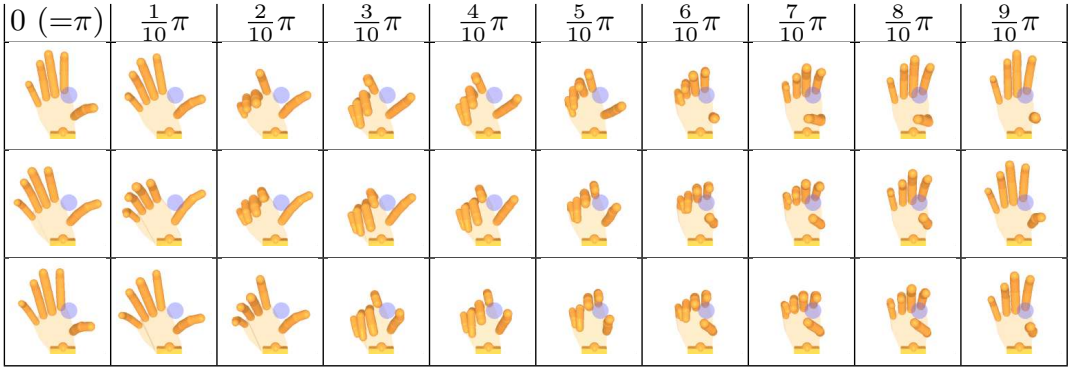

Fig. 4. Visualisation of the training result in the hand posture space. The depicted postures correspond to the reprojections $\mathbf{f}(\mathbf{x};\mathbf{X})$ of regularly sampled positions $\mathbf{x}$ in the trained latent space. Please consider also the video available under `http://www.techfak.uni-bielefeld.de/~jsteffen/mov/ki2009/upkrturn/`.

Indeed, whereas the distortion is beneficial for the production of motions, it poses some problems for the inverse direction, i.e. projecting hand posture sequences into latent space. In that case, whereas the temporal information is robust, the projection is strongly non-robust in the parameter (radius) dimension in the described radius-independent parts as the corresponding hand postures are fairly similar for the whole range of latent radii for a specific point in time and the result of $\mathbf{g}(\mathbf{Y})$ can heavily vary for small changes of $\mathbf{y}$.

7 Motion Production and Synthesis

The clear structure of the latent space enables the use of a very simple controller in order to synthesise the represented movements: The algorithm starts in an initial hand posture corresponding to a fixed latent position on the 'maximum radius' border of the latent space in a temporal position where the fingers have contact with the cap. The motion controller then is subdivided into two different phases of orthogonal, straight navigations through the latent space: (a) Grasping the cap is realised by a straight navigation in direction of decreasing radii following the radius dimension until thumb, fore finger and middle finger have contact. (b) The manipulation – during which the adapted radius is fixed – is performed by navigating through the latent space following the temporal dimension. Please consider also the corresponding video[1] and [9] for further details.

8 Towards Motion Recognition and Segmentation

The recognition approach takes the inverse direction to the motion production described in the last section: instead of projecting latent trajectories into hand posture space in order to determine a sequence of intermediate target hand postures, we now observe such sequences and use it as input. By projecting them into latent space ($\mathbf{g}(\cdot)$) and back to hand posture space ($\mathbf{f}(\mathbf{g}(\cdot))$, cp. Sec.

[1] http://www.techfak.uni-bielefeld.de/~jsteffen/mov/ki2009/upkrmanip/

3), we obtain means to define features which express the degree of compatibility of the observed postures and the manifold:

a) The compatibility of single observations with the manifold can be expressed with the self-reconstruction error of the observations yielding a measure for the similarity of observed and best-matching represented posture in the manifold:

$$C_{rec}(\mathbf{y}^\star; \mathbf{Y}) = -1 + 2 \cdot exp\left(-\Delta^T \Delta\right) \in [-1; 1] \tag{7}$$

where $\Delta(\mathbf{y}^\star) = \mathbf{y}^\star - \mathbf{f}(\mathbf{g}(\mathbf{y}^\star))$ is the self-reconstruction error of observation $\mathbf{y}^\star$.

b) The *temporal* compatibility of a single observation with its preceding observations (*history*) can be expressed by the relative positions of the representations of the current and preceding observations in latent space: if the single observations of the input sequence are compatible with the manifold (in the sense of (a)), then, the compatibility of the chronological order of their latent representations with the manifold can be expressed as the sum over the distances between projections of successive data. As a measure for the compatibility of the observation $\mathbf{y}_{t-h}$ in the history of $\mathbf{y}_t$, we thus define:

$$c_{hist}(h, t) = \frac{1}{2}\cos(\delta_{h,t}) + \frac{1}{2} C_{rec}(\mathbf{y}_{t-h}; \mathbf{Y}) \tag{8}$$

where $\delta_{h,t} = mod_\pi(\mathbf{g}(\mathbf{y_{t-h-1}}) - \mathbf{g}(\mathbf{y_{t-h}}))$ is the *directed* temporal difference of the latent space projections $\mathbf{g}(\cdot)$ of the historic observation $\mathbf{y}_{t-h}$ and its predecessor $\mathbf{y}_{t-h-1}$ (taking the period π of the dimension into account). C_{rec} again is the self-reconstruction error described in (a).

For the compatibility of the whole history of $\mathbf{y}_t$ of length H, we define:

$$C_{hist}(H, t) = \frac{\sum_{h=1}^{H} \gamma^h c_{hist}(h, t)}{\sum_{h=1}^{H} \gamma^h} \tag{9}$$

where $\gamma \in [0; 1]$ is the discount factor for historic observations. As C_{rec}, C_{hist} can take values in $[-1; +1]$ whereas -1 corresponds to maximally incompatible and $+1$ to maximally compatible with the underlying UKR manifold.

The combination of (a) and (b) with $\lambda \in [0; 1]$ to an overall compatibility measures yields:

$$C = \lambda C_{rec} + (1 - \lambda) C_{hist} \in [-1; +1] \tag{10}$$

Like this, C gives a measure for the compatibility of the observation together with its history with the underlying manifold. In other words, C realises a measure to quantify the appropriateness of the candidate manifold to reproduce the observation and the history. The classification of the observation to one of several candidate classes then can be realised as a winner-takes-all mechanism that works on the results of all UKR manifolds.

The special strength of this approach is that the compatibility with the represented motion is computed for each point separately (incorporating few historic hand postures) and is thus independent of a fixed data window. In addition, this enables the method to work on inhomogeneous data sequences which consist of more than one movement and hence enables the use for a segmentation of such

sequences into several candidate motions (each represented as Structured UKR manifold) or even only motion parts.

9 Conclusion

In the field of humanoid robotics, two of the key challenges are the production of naturally looking movements on the one hand and the recognition of observed movements or their segmentation into several meaningful subparts on the other hand. In this paper, we presented our idea of how these two problems can be based on one and the same core representation – namely the *Manipulation Manifolds* consisting of Structured UKR manifolds.

After a short revision of the basic method and the generation and use of the manifolds for the action production part of the system, we presented our basic plan to perform recognition and segmentation tasks on the basis of the same representation. For this part, only initial evaluation has been done. Indeed, the results are very promising and we are convinced that an elaborate evaluation will help us to further refine our approach.

ACKNOWLEDGEMENT This work has been carried out with support from the German Collaborative Research Centre "SFB 673 - Alignment in Communication" granted by the DFG and from the German Cluster of Excellence 277 "Cognitive Interaction Technology" (CITEC).

References

1. S. Bitzer, I. Havoutis, and S. Vijayakumar. Synthesising Novel Movements through Latent Space Modulation of Scalable Control Policies. In *Proc. SAB*, 2008.
2. I.T. Jolliffe. *Principal Component Analysis*. Springer, New York, 2nd edition, 2002.
3. S. Klanke. *Learning Manifolds with the Parametrized Self-Organizing Map and Unsupervised Kernel Regression*. PhD thesis, Bielefeld University, 2007.
4. N. Lawrence. Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models. *Machine Learning Research*, 6, 2005.
5. P. Meinicke, S. Klanke, R. Memisevic, and H. Ritter. Principal Surfaces from Unsupervised Kernel Regression. *IEEE Trans. PAMI*, 27(9), 2005.
6. E. A. Nadaraya. On Estimating Regression. *Theory of Probability and Its Appl. (9)*, 1964.
7. G. Rizzolatti, M. Fabbri-Destro, and L. Cattaneo. Mirror neurons and their clinical relevance. *Nat Clin Pract Neuro*, 5(1):24–34, 2009.
8. J. Steffen, R. Haschke, and H. Ritter. Towards Dextrous Manipulation Using Manifolds. In *Proc. Int. Conf. on Intelligent Robots and Systems (IROS)*, 2008.
9. J. Steffen, S. Klanke, S. Vijayakumar, and H. Ritter. Realising Dextrous Manipulation with Structured Manifolds using Unsupervised Kernel Regression with Structural Hints. In *ICRA Workshop: Approaches to Sensorimotor Learning on Humanoid Robots*, May 2009. (to appear).
10. J. Steffen, S. Klanke, S. Vijayakumar, and H. Ritter. Towards Semi-supervised Manifold Learning: UKR with Structural Hints. In *Proc. WSOM*, June 2009. (ta).
11. 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.
12. R. Urtasun, D. Fleet, A. Geiger, J. Popovic, T. Darrell, and N. Lawrence. Topologically-Constrained Latent Variable Models. In *Proc. ICML*, 2008.
13. G. S. Watson. Smooth Regression Analysis. *Sankhya, Ser.A*, 26, 1964.