Neural competition for motion segmentation

Jan Steffen[1], Michael Pardowitz[12], Jochen Steil[2], and Helge Ritter[12]

[1]Neuroinformatics Group, Faculty of Technology, Bielefeld University, Germany
[2]Research Institute for Cognition and Robotics (CoR-Lab), Bielefeld Univ., Germany

Abstract. We present a system for sensory classification and segmentation of motion trajectories. It consists of a combination of manifolds from *Unsupervised Kernel Regression (UKR)* and the recurrent neural *Competitive Layer Model (CLM)*. The UKR manifolds hold learned representations of a set of candidate motions and the CLM dynamics, working on features defined in the UKR domain, realises the segmentation of observed trajectory data according to the competing candidates. The evaluation on trajectories describing four different letters yields improved classification results compared to our previous, pure manifold approach.

1 Introduction

In previous work, we studied the use of UKR manifolds for the representation of motion capture data for motion production [1]. We have also shown how these representations can be used for the recognition of the represented motions [2].

The CLM is a layered recurrent neural network consisting of threshold units. It is able to segment features of processed data into *perceptual groups*. The segmentation is based on the evaluation of pairwise compatibilities of the input features (e.g. [3, 4, 5]). The fundamental requirements for the grouping, therefore, are the preprocessing into appropriate data features as inputs and a measure for their pairwise compatibilities.

We present how the preprocessing can be performed in the domain of UKR manifolds and how a corresponding general compatibility measure can be defined. We apply both on observed letter trajectories in order to segment them according to a set of previously trained candidates. The evaluation on letter trajectories yields promising results.

2 Competitive Layer Model (CLM)

The CLM is a recurrent neural network for grouping and segmentation of sensory input features. It has been introduced in [3] for spatial feature linking, further applied to image segmentation and perceptual grouping (e.g. [4, 5]) and recently transfered to action segmentation for robot task learning [6].

The fundamental basis of the CLM is an appropriate preprocessing that extracts input features from the processed data and a corresponding function that provides a pairwise measure for input compatibilities.

The CLM segments a set of N input features into groups of mutually compatible features. Each group is represented by one CLM layer. Each layer provides one neuron for every input feature. The activities of these neurons express the

assignments of the associated features to the corresponding layers. During the grouping process, each layer competes for the exclusive assignment of all features that are compatible with the layer. Originally, this compatibility association is usually not specified beforehand, but dynamically evolves from previous (partial) assignments of features to the same layer.

The grouping dynamics is driven by two main ingredients: (a) an *intra-layer* pairwise compatibility negotiation between the assigned features and (b) an *inter-layer* Winner-Takes-All mechanism between the neurons from all layers that correspond to the same feature. The CLM describes these two ingredients as terms of a *grouping energy* function, that provides a measure for the quality of a specific grouping result. Applying the CLM dynamics minimises this energy and it can be shown (cf. [4]) that it converges (under certain conditions) to stable fixed-points representing local minima of the grouping energy and thus to locally optimal groupings of the input features.

To sum up, the crucial issues in using the CLM are the data preprocessing and the definition of the compatibility function. In our approach, both issues are realised in the domain of the UKR manifolds, as described in Sec.4.

For further details on CLM, please refer to [3, 4].

3 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 $\mathbf{y}=\mathbf{f}(\mathbf{x})$. The method has been introduced in [7] as the unsupervised counterpart of the Nadaraya-Watson kernel regression estimator. In its basic form, UKR uses the Nadaraya-Watson estimator $\mathbf{f}(\mathbf{x})=\sum_{i=1}^{N}\mathbf{y}_i\frac{K_{\mathbf{H}}(\mathbf{x}-\mathbf{x}_i)}{\sum_j K_{\mathbf{H}}(\mathbf{x}-\mathbf{x}_j)}$ as smooth mapping $\mathbf{f}\colon\mathbf{x}\in\mathbb{R}^q\rightarrow\mathbf{y}\in\mathbb{R}^d$ from latent to observed data space ($K_{\mathbf{H}}$: density kernel with bandwidth $\mathbf{H}$). In UKR, $\mathbf{X}=\{\mathbf{x}_i\}$ now plays the role of input data to the regression function and is treated as set of *latent parameters* corresponding to $\mathbf{Y}$.

UKR training, i.e. finding optimal latent variables $\mathbf{X}$, is realised as gradient-based minimisation of the reconstruction error $R(\mathbf{X})=\frac{1}{N}\sum_i\parallel\mathbf{y}_i-\mathbf{f}(\mathbf{x_i};\mathbf{X})\parallel^2$. Most notably, UKR can perform leave-one-out cross-validation without additional computational cost. In addition, it can easily be initialised with the results of spectral embedding methods like Isomap in order to improve its robustness against poor local minima.

The inverse mapping $\mathbf{x}=\mathbf{f}^{-1}(\mathbf{y};\mathbf{X})$ from latent to observation space is not directly supported. For our approach, we use the approximation $\hat{\mathbf{x}}=\mathbf{g}(\mathbf{y};\mathbf{X})=\arg\min_{\mathbf{x}}\|\mathbf{y}-\mathbf{f}(\mathbf{x};\mathbf{X})\|^2$ as it is proposed in [7].

In the original form, UKR is a purely unsupervised approach to continuous manifold learning. In order to enable to incorporate prior knowledge about the structure of the training data, we introduced a structured version of UKR training (e.g. [1]). With Structured UKR, it is possible to represent data with a temporal context, like trajectories of hand positions, in a very easy and robust

way. In particular, due to the specific training of Structured UKR, the order of the represented time series of training observations $\mathbf{y}_i$ is reflected in their latent parameters $\mathbf{x}_i$ and is captured by on specific latent time dimension. In order to represent periodic motions, we use the periodic kernel $K_{\circlearrowleft}(x_i - x_j; \Theta) = \exp\left[-\frac{1}{2}\Theta^2 \sin^2(x_i - x_j)\right]$. For further details on UKR, please refer to [7, 8, 1].

4 Combining UKR and CLM

In the combination of the two methods, the CLM performs the segmentation of observed trajectory samples according to a set of candidate patterns. These candidates are represented by UKR manifolds. The CLM input features of the observed trajectory, which are required by the CLM dynamics, are defined in the UKR manifold domains of these candidates.

Whereas the CLM has been designed in a general fashion, it usually uses global layer-independent features and therefore focusses on the pure pairwise mutual compatibilities of these features. On the contrary, in order to segment according to a known set of candidates, we associate every CLM layer with one specific UKR manifold and the layers' input features are computed on the basis of the layer-specific UKR manifolds. Every CLM layer therefore has its own input features and the CLM dynamics focusses on the segmentation of the features into coherent groups which are compatible with the assigned CLM layer.

The main issue of the UKR/CLM combination is to define both a preprocessing function and a compatibility measure *in the UKR manifold domain*. This procedure decouples the CLM from the structure and the characteristics of the data itself, and only focusses on their UKR representations. Since these representations are designed to be unified for a broad range of applications, we retrieve a more general approach for any data represented in UKR manifolds.

The manifold features are inspired by our initial, purely manifold-based approach which did not include competition of the candidates [2]. The features are based on (a) the order of the UKR latent representations $\hat{\mathbf{x}}_i = \mathbf{g}(\mathbf{y}_i, \cdot)$ of the observed trajectory samples $\{\mathbf{y}_i\}$ and (b) the normalised UKR self-reconstruction errors (sre) $e_i^{\text{sre}} = \boldsymbol{\Delta}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Delta}$ with $\boldsymbol{\Delta} = \| \mathbf{y}_i - \mathbf{f}(\hat{\mathbf{x}}_i, \cdot) \|$ and $\boldsymbol{\Sigma}$ being the diagonal matrix of the dimension-wise UKR training data variances (cf. Sec.3 for details on $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$). Due to the temporal context of the sequences, we usually restrict the mutual interactions to a limited time horizon by only considering a temporal neighbourhood $\mathcal{N}_i = \{\mathbf{y}_j \mid 0 < \| i - j \| < H\}$ of each input feature $\mathbf{y}_i$. H is the neighbourhood parameter of the method: large values of H focus on more global structures whereas small values are rather sensitive to local structures.

(a) can be directly used as compatibility indicator of observed and represented motion: since the latent space of a Structured UKR manifold reflects the temporal order of the represented pattern, the time components of the latent projections $\hat{\mathbf{x}}_i$ of compatible observations are in correct chronological order. Vice versa, observations whose latent projections are not correctly ordered are likely to be incompatible. Taking the period π of the latent time dimensions into account, we define the *latent order compatibility (loc)* of two observations

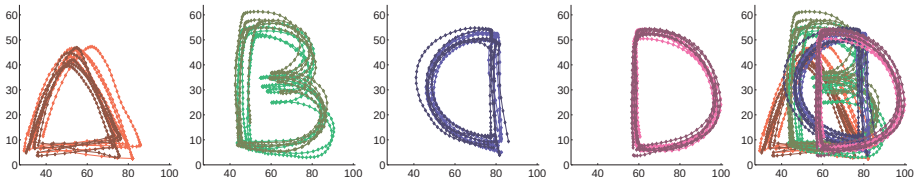


Fig. 1: 2D training (bright) and test (dark) data for A-,B-,C-,D-trajectories.

$\mathbf{y}_i$ and $\mathbf{y}_j$ $(i < j; j \in \mathcal{N}_i)$ as:

$$c_{ij}^{\text{loc}} = c_{ji}^{\text{loc}} = H_{\circlearrowleft}^{\pi}(\hat{\mathbf{x}}_j - \hat{\mathbf{x}}_i) \tag{1}$$

and $\forall i\colon c_{ii}^{\text{loc}} = 0$. $H_{\circlearrowleft}^{\pi}(\cdot) = \text{sgn}(\cos(\text{mod}_{\pi}(\cdot)))$ serves as a Signum function for periodic data. The mod_{π} realises a direction sensitivity in the periodic dimension.

(b) can also be used directly as compatibility measure (cf. [2]). Indeed, considering the average self-reconstruction error in the temporal neighbourhood $\mathcal{N}_j$ of the evaluated $\mathbf{y}_j$ turned out to be more robust (again for observations $\mathbf{y}_i$ and $\mathbf{y}_j$, whereas $i < j$ and $j \in \mathcal{N}_i$):

$$c_{ij}^{\text{sre}} = c_{ji}^{\text{sre}} = -1 + 2\exp\left(-\frac{1}{\|\mathcal{N}_j\|}\sum_{k\in\mathcal{N}_j}\left\|\frac{\Delta_e(k)}{\Delta_{\hat{\mathbf{x}}}(k)}\right\|\right) \tag{2}$$

where $\Delta_e(k) = e_k^{\text{sre}} - e_{k-1}^{\text{sre}}$ and $\Delta_{\hat{\mathbf{x}}}(k) = \hat{x}_{k,d_t} - \hat{x}_{k-1,d_t}$ with d_t being the latent time dimension and $\Delta_e(1) = \Delta_{\hat{\mathbf{x}}}(1) = 0$.

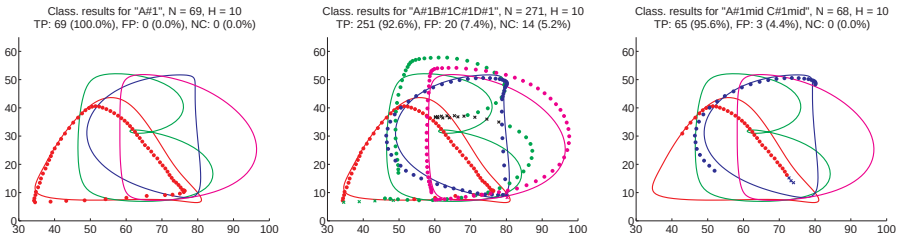
The matrix $\mathbf{F}$ of CLM compatibilities can be denoted as a balanced combination of both measures, with components $f_{ij} = \frac{1}{2}(c_{ij}^{\text{loc}} + c_{ij}^{\text{sre}}) \in [-1; 1]$ describing the compatibility of the observations $\mathbf{y}_i$ and $\mathbf{y}_j$ with each other and with the underlying layer-corresponding UKR manifold.

5 Evaluation

The evaluation focusses on the recognition abilities of the combined CLM/UKR system. The training and test data (cf. Fig. 1) each consist of five trajectories for each of the letters 'A', 'B', 'C', and 'D'. Each trajectory corresponds to the path of a tracked hand drawing a letter in the air and is extracted from monocular camera pictures from an orthogonal view onto the virtual drawing plane. The tracking has been done with the ARToolkit system (http://artoolkit.sourceforge.net) only utilising the x/y-components corresponding to the 2D projections onto the drawing plane.

In preparation of the evaluation, one Structured UKR manifold for each of the letters is trained, each on the basis of five training sequences of the corresponding letter and each with a one-dimensional periodic latent space (cf. Sec.3). The trained manifolds then are associated with one of the four CLM layers.

The evaluation of a test sequence consisting of observed trajectory samples $\{\mathbf{y}_i\}, i = 1 \ldots N$ is performed as follows: (a) Each CLM layer is equipped with an



(a) (b) (c)

Fig. 2: Exemplary evaluation results. Best viewed in colour. See text for details.

ordered set of N neurons corresponding to the N observation samples. Each observation $\mathbf{y}_i$ therefore has a four-dimensional neural representation in the CLM. The components constitute the neural activities of the candidate letters represented by the layer-specific UKR manifolds. All neurons are initialised with small positive random activations. (b) For each CLM layer, the layer-specific $N \times N$ interaction matrix $\mathbf{F}$ is computed on the basis of the layer-associated manifold (see last section). (c) The CLM dynamics is applied until convergence, i.e. until only one non-zero neural activity remains in the 4D neural representation of any $\mathbf{y}_i$. Afterwards, all observations are uniquely assigned to one CLM layer and to one UKR representation of a candidate letter, respectively. These assignments constitute the final pointwise classification results.

Figure 2 exemplarily depicts such classification outcome. The solid lines visualise the layer-specific UKR representations of the candidate letters. The points (correct classification) and crosses (false classification) correspond to the processed observations of the test trajectories. The colours correspond to the classified letters 'A' (red), 'B' (green), 'C' (blue), and 'D' (magenta).

Three different applications of the recognition system are illustrated: In Fig. 2(a), a previously segmented whole letter trajectory is processed. In Fig. 2(b), the system is applied to non-segmented observations consisting of letter concatenations and is used to segment the trajectories according to the set of candidates. Figure 2(c) shows that even concatenated trajectory *parts* can be processed and segmented.

Table 1 gives an overview over several classification/segmentation results. Listed are the mean percentages of true positive point classifications over four evaluation runs. The sequences 'A_1' to 'D_1' correspond to trajectories of whole single letters (cf. Fig. 2(a)), '$A_1B_1C_1D_1$' and '$D_1C_1B_1A_1$' to concatenations of whole letter trajectories (cf. Fig. 2(b)) and '$A_1^{mid}B_1^{mid}$' to '$A_1^{mid}D_1^{mid}$' to concatenations of the middle parts (half of the points) of the corresponding letter trajectories (cf. Fig. 2(c)). In order to analyse different *sample rates* and parametrisations, the evaluations of the sequences consider *every* trajectory sample ($N^{th} = 1$) or only *every second* ($N^{th} = 2$), each for the neighbourhood sizes $H = 5, 10, 15$, and 20 (cf. Eq. 2). Supplemental results can be found under `http://www.techfak.uni-bielefeld.de/~jsteffen/esann2010/`.

The method achieves to correctly classify a high percentage of the evalu-

	$N^{th} = 1$				$N^{th} = 2$			
Sequence	H=5	H=10	H=15	H=20	H=5	H=10	H=15	H=20
A_1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
B_1	70.3	83.8	83.8	90.9	93.2	100.0	100.0	100.0
C_1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
D_1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
$A_1B_1C_1D_1$	93.4	92.6	91.9	83.0	99.3	98.5	89.7	88.2
$D_1C_1B_1A_1$	93.7	83.0	83.8	86.3	88.2	89.7	93.4	86.6
$A_1^{mid}B_1^{mid}$	80.3	76.3	75.0	71.1	87.2	83.3	92.3	92.3
$A_1^{mid}C_1^{mid}$	98.5	95.6	92.6	88.2	97.1	88.6	80.0	77.1
$A_1^{mid}D_1^{mid}$	100.0	97.2	94.8	91.7	99.3	91.9	83.1	73.0

Table 1: List of evaluation results. See text for details.

ated points. If errors occur, they are not distributed over the whole trajectory, and thus, the sequences of correctly classified points are rarely interrupted as shown in Fig. 2. This effect results from the 'gap-filling' characteristics of the CLM dynamics and constitutes a large advantage in the classification robustness compared to the approach without CLM described in [2].

6 Conclusion

We presented a new approach to competitive sensory segmentation of motion trajectories. It extends our previous approach of using UKR manifolds for motion recognition by embedding them into a Competitive Layer Model. Through the CLM neural competition of the candidate motions, previously learned patterns can be recognised very robustly, as we have shown in the evaluation on letter trajectories.

ACKNOWLEDGEMENT This work has been carried out with support from the German Collaborative Research Centre "SFB 673: Alignment in Communication" granted by the DFG.

References

[1] J. Steffen, S. Klanke, S. Vijayakumar, and H. Ritter. Towards Semisupervised Manifold Learning: UKR with Structural Hints. In *Proc. WSOM*, 2009.

[2] J. Steffen, M. Pardowitz, and H. Ritter. Using Structured UKR Manifolds for Motion Classification and Segmentation. In *Proc. IROS*, 2009.

[3] H. Ritter. A Spatial Approach to Feature Linking. In *Proc. INNC*, 1990.

[4] H. Wersing, J. Steil, and H. Ritter. A Competitive Layer Model for Feature Binding and Sensory Segmentation. *Neural Computation*, 13(2):357–387, 2001.

[5] J. Ontrup, H. Wersing, and H. Ritter. A computational feature binding model of human texture perception. *Cognitive Processing*, 5(1):32–44, 2004.

[6] M. Pardowitz, R. Haschke, J. Steil, and H. Ritter. Gestalt-Based Action Segmentation for Robot Task Learning. In *Proc. HUMANOIDS*, 2008.

[7] P. Meinicke, S. Klanke, R. Memisevic, and H. Ritter. Principal Surfaces from Unsupervised Kernel Regression. *Trans. on PAMI*, 27(9):1379 – 1391, 2005.

[8] S. Klanke. *Learning Manifolds with the Parametrized Self-Organizing Map and Unsupervised Kernel Regression*. PhD Thesis, Bielefeld University, Bielefeld, Germany, 2007.