Efficient Local Subspace Construction for Neural Data Modeling

J. Bruske, J. Tröster, G. Sommer
Computer Science Institute, Christian-Albrechts-Universität Kiel
Kiel, Germany

Abstract

In this article we show how optimally topology preserving maps (OTPMs) can be used for efficient local subspace construction and how existing local approximation networks can benefit from the local subspaces such constructed. The network models include radial basis function (RBF) networks, local linear maps (LLM) and local linear data modeling.

The local subspaces are constructed by local principal component analysis (PCA). Exploiting the OTPM, the local PCA can be shown to have only *linear time complexity* w.r.t. the dimensionality of the input space (in contrast to the prohibitive *cubic complexity of the conventional approach*), and hence the method becomes applicable even for very high dimensional input spaces as frequently encountered in computer vision, cf. [2]. Interesting on its own, we demonstrate the workability of the subspace approach by means of an appearance based robot grasping system.

1 Introduction

It has long been noticed that if input data stemming from some low d-dimensional manifold are embedded in a high n-dimensional input space, a local approximation scheme should work by projecting input data to the manifold and restrict approximation to within the manifold (hence eliminating noise orthogonal to the manifold). In a straightforward approach (e.g. [3]) a number of centers is distributed within the input manifold and an approximation to the manifold by local subspaces is obtained by local PCA of data in the Voronoi cells of the centers. Yet PCA has serial time complexity $O(n^3)$ independent of the intrinsic dimensionality d of manifold, and hence the approach becomes prohibitive for high dimensional input spaces.

On ICANN'97, [2], we have shown how local subspace analysis by PCA becomes possible in time $O(n + m(d)^3)$ using optimally topology preserving maps (OTPMs), where $m(d)$ is a function of the intrinsic dimensionality of the manifold only. Hence by using OTPMs, local subspace analysis scales only linear with the input dimensionality and suddenly a large range of local subspace approximation schemes becomes applicable. For example, in the local linear data modeling approach [3], data are modeled within the local linear subspace. In RBF networks, stimuli can be projected to the local subspace prior to calculation of the activation function, resulting in Hyper Basis Function (HBF) networks, [5]. Finally, in Ritter's Local Linear Map (LLM), [7], the subspaces can be used for both projection and reduction of storage requirements for the Jacobian matrices.

2 Efficient local subspace construction

For the convenience of the reader, we will now briefly review the basic procedure for efficient local subspace construction with optimally topology preserving maps as presentend in [2]. Given a training set $T \subset \mathbb{R}^n$, it proceeds in four stages (batch-variant).

1. Generate a set of N centers $S = \{c_1, \ldots, c_N\} \subset \mathbb{R}^n$ as the output of a vector quantization algorithm working on the training set T.
2. Calculate the graph $G = (V, E)$ by
   (a) associating each center in S with a node in V, i.e.
   $$|V| = |S| \text{ and } c_i \in S \Leftrightarrow i \in V$$
   (b) for each $x \in T$, connecting the nodes associated with the best and second best matching centers, i.e.
   $$E = \{(i,j) \mid \exists x \in T \, \forall k \in V \backslash \{i,j\} : \max\{\| c_i - x \|, \| c_j - x \|\} \leq \| c_k - x \|\}$$
   G is called the optimally topology preserving map, $OTPM_T(S)$, of S w.r.t. T, cf. [2].
3. For each node $i \in V$ perform a principal component analysis of the set of m_i difference vectors $\{c_{1_i} - c_i, \ldots, c_{m_i} - c_i\}$, with $(c_{j_i} - c_i)$ the difference vectors between c_i and c_{j_i}, the center of its j-th direct topological neighbor in G.
4. Exclude eigenvectors corresponding to very small eigenvalues.

As a result of the vector quantization stage (step 1) the centers are placed within the data manifold $M \subseteq \mathbb{R}^n$ and noise orthogonal to M is filtered out. In step 2, $OTPM_T(S)$ is constructed by simply connecting nodes corresponding to best and second best matching centers on presentation of T.

The central "trick" in step 3 is to use the difference vectors $(c_{j_i} - c_i)$ for PCA of each local subspace and not the data in a local region itself. First, the difference vectors have very low noise component orthogonal to M (due to the noise reduction property of the vector quantizing stage), and second, the number of neighbors m of a node in an OTPM does only depend on the intrinsic dimensionality d and is small for small d. The latter property can be exploited to perform local PCA in time $O(m(d)^2 n + m(d)^3)$, [2], hence scaling only linearly (optimally) with the input dimensionality n.

Deciding in step 4 what size an eigenvalue μ_i as obtained by each local PCA must have to indicate an associated intra-manifold eigenvector (and not a noise component), amounts to determining a threshold α (significance level). In this work, we will regard an eigenvalue μ_i as significant if $\frac{\mu_i}{\max_j \mu_j} > \alpha$. If no prior knowledge is available, different values of α have to be tested.

3 Local subspaces for neural data modeling

Local subspace construction as described in section 2 supplies us with a set of (orthonormal) eigenvectors $e_1^i, \ldots, e_{l_i}^i$, $l_i \leq m_i$, spanning a local subspace for each center $c_i \in S$. In this section we show how knowledge of these subspaces can be used to improve existing local approximation schemes.

3.1 Local linear modeling

In [3], Kambhatla and Leen proposed an algorithm for local linear modeling of data based on clustering and straightforward PCA. They first generate a set of centers S by clustering the data with a vector quantizer, then they use local PCA in each Voronoi cell to obtain the local eigenvectors. New data is coded as the index of the best matching unit, bmu, together with the projection coefficients to the local eigenvectors $(x-c_{bmu})^T e_1^{bmu}, \ldots, (x-c_{bmu})^T e_{l_{bmu}}^{bmu}$. The reconstruction (decoding) is given by

$$\hat{x} = c_{bmu} + \sum_{i=1}^{l_{bmu}} ((x - c_{bmu})^T e_i^{bmu}) e_i^{bmu}. \tag{1}$$

Utilizing the efficient local subspace construction procedure (section 2) we were able to apply local linear modeling to a sequence of (64×64)-dimensional images in [2].

3.2 Subspace Local Linear Maps

The Local Linear Map (LLM) as introduced by Ritter et al., [7], has found widespread application for learning input - output mappings. The LLM rests on a locally linear (first order) approximation of the unknown function $f : R^n \to R^k$ and computes its output as (winner-take-all variant)

$$y(x) = A_{bmu}(x - c_{bmu}) + o_{bmu}. \tag{2}$$

Here $o_{bmu} \in R^k$ is an output vector attached to the best matching unit (zero order approximation) and $A_{bmu} \in R^{k \times n}$ is a local estimate of the Jacobian matrix (first order term). Centers are distributed by a clustering algorithm.

Due to the first order term, the method is very sensitive to noise in the input. With a noised version $x' = x + \eta$ the output differs by $A_{bmu}\eta$, and thus the LLM largely benefits from projecting to the local subspace, cancelling the orthogonal (w.r.t M) noise component of η. Equally important, instead of adapting and storing $k \times n$ parameters with each matrix A_i, by first projecting to the local l_i-dimensional subspaces only matrices $A'_i \in R^{k \times l_i}$ need to be stored. This results in much better scaling with the input dimension n and, because of the reduced number of free parameters, better learning and generalization properties. The Subspace LLM (SLLM) proposed in this article hence takes the form

$$y(x) = A'_{bmu} E_{bmu}(x - c_{bmu}) + o_{bmu}, \tag{3}$$

where $E_i = [e_1^i, \ldots, e_{l_i}^i]$ denotes the local projection matrix as calculated by the efficient subspace construction procedure.

3.3 Subspace RBF networks

RBF networks for learning input - output mappings approximate the unknown function $f : R^n \to R^k$ by

$$y(x) = \sum_{c_i \in S} o_i h_i(\| x - c_i \|), \tag{4}$$

where $h_i : R^+ \to [0,1]$ denote strictly monotonically decreasing activation functions, e.g. $h_i(z) = \exp(-z^2/\sigma_i^2)$. Their netto input is usually calculated as the squared Euclidean distance, i.e. $\| x - c_i \|^2 = (x - c_i)^2$. With the squared Euclidean distance, some noise component η orthogonal to the input manifold will result in a change of η^2 of all netto inputs and hence seriously affect approximation. Again, this noise component can be filtered out by projecting the input to the local subspaces. This amounts to setting $\| x - c_i \|^2$ to $(x-c_i)^T E_i E_i^T (x-c_i)$ where $E_i = [e_1^i, \ldots, e_{l_i}^i]$ is the local projection matrix. The basis functions now become Hyper Basis Functions (HBFs), cf. [5].

In general, the manifold can be curved and wrapped and hence we can not use all HBFs for approximation for a given stimulus (because they have lost any sensitivity orthogonal to the local eigenvectors). A solution to this problem is to use only the HBFs of the best matching unit and its direct topological neighbors w.r.t. the OTPM, $Nh^+(bmu)$, for output calculation:

$$y(x) = \sum_{i \in Nh^+(bmu)} o_i h_i((x - c_i)^T E_i E_i^T (x - c_i)), \tag{5}$$

Note that eq. (5) is just the Dynamic Cell Structure approximation scheme [1], except the additional projection to the local subspaces.

3.4 Normalized RBF networks and the effective projection property

In normalized RBF networks, the output of the RBF network is normalized with the total activation, i.e.

$$\bar{y}(x) = \sum o_i h_i / \sum h_i. \tag{6}$$

Interestingly, in case of Gaussian activation functions with equal widths σ_i and all centers c_i lying in a linear subspace $H \subset \mathbb{R}^n$, these nework possess an *effective projection property* [1]: They are insensitive to noise orthogonal to H without explicitly projecting to H, i.e.

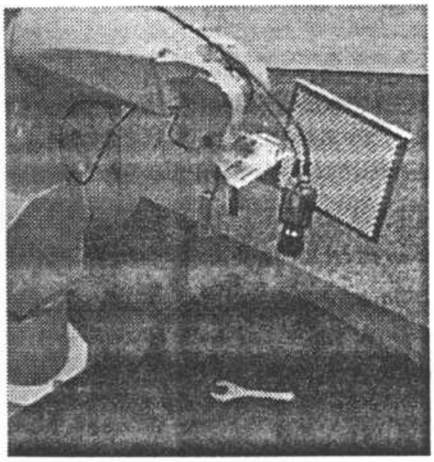
$$\bar{y}(x + \eta) = \bar{y}(x) \quad \text{for } x \in H \text{ and } \eta \perp H. \tag{7}$$

Hence in this particular case, explicit projection to H has little effect.

4 The robot grasping application

Our demonstration concerns appearance based pose estimation of objects, cf. [4]. Fig. 1 (left) shows the experimental setup. Given an image of the object, the robot has to estimate its pose and to grasp it. The flat object has only one degree of freedom, its rotation around the z-axis, and hence images under different rotations lie on a 1-dimensional trajectory in image space.

Preprocessing of the image involves segmentation, scale normalization, taking the logarithm of grey values and convolving the image with 75 DC-corrected Gabor filters. The latter are distributed on a 5×5 grid, with 3 orientations on each position. As bandpass filters the Gabor filters serve two purposes: In conjunction with the logarithmically graded grey values they first achieve brightness invariance (filter out the DC component) and, second, they filter out high frequencies which would lead to discontinuous trajectories in image space. The 75 filter responses serve as input to the networks. We use 180 images and their corresponding grasping angles as a training set, 180 as a test set (differing by 1° rotation from those in the test set).



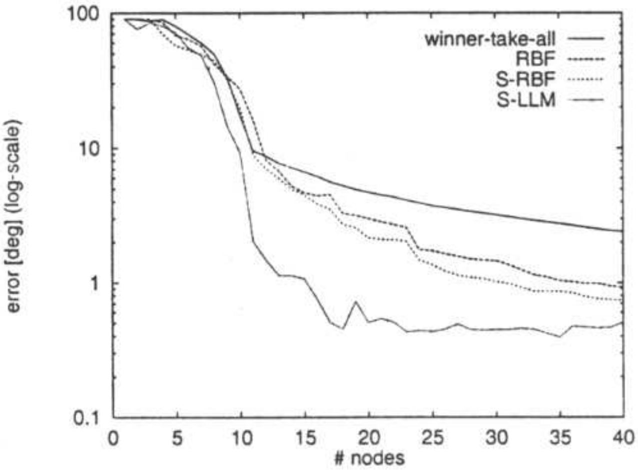


Figure 1: Left: Robot arm with attached camera for appearance based grasping of a spanner. Right: Pose error [°] for Subspace LLM on $\alpha = 0.2$ level, normalized RBF and Subspace RBF network and winner-take-all network.

Figure 1 (right) shows the grasping accuracy for a Subspace-LLM on the $\alpha = 0.2$ significance level, for a normalized Subspace-RBF network on the $\alpha = 0.2$ level, for a normalized RBF network and a winner take all scheme (averaged over 180 test images). All approximation schemes used the same center distribution (generated by an incremental LBG vector quantization algorithm). For the RBF networks we used the activation functions

$$h_i(z) = \exp(-z^2/\sigma_i^2) \text{ with } \sigma_i^2 \sim \frac{1}{|Nh(i)|} \sum_{j \in Nh(i)} (c_j - c_i)^2.$$

For each number of centers the output vectors o_i (and Jacobian matrices A_i in case of the Subspace-LLM) were optimized by singular value decomposition (SVD) [6]. Not surprisingly, the Subspace-LLM shows the best performance and achieves a pose estimation error of 1° with as few as 15 nodes in the network. The normalized Subspace-RBF network, however, performs only slightly better than the simple normalized RBF network. This can be attributed to the effective projection property of normalized RBF networks.

5 Conclusion

We have shown how to efficiently construct local subspaces utilizing OTPMs and how to use these subspaces for improved local approximation schemes. By projecting data to the relevant subspaces, othogonal noise is filtered out and, as in the case of the Subspace-LLM, the number of free parameters can be dramatically decreased.

Concerning our grasping application, we have succeeded in extending the demonstration presented in this article to an appearance based robot grasping system for multiple objects. In another project, we have equally successfully applied the local subspace approximation schemes for estimating the head posture in facial images. Based on these experiences we are confident that the presented subspace approach may become a valuable tool in pattern recognition applications characterized by high dimensional input spaces but low intrinsic dimensionality, in particular visual learning.

References

[1] J. Bruske. *Dynamische Zellstrukturen - Theorie und Anwendung eines KNN-Modells.* PhD thesis, Inst. f. Inf. u. Prakt. Math., Christian-Albrechts-Universitaet zu Kiel, 1998. (submitted).

[2] J. Bruske and G. Sommer. Topology representing networks for intrinsic dimensionality estimation. In *Proc. ICANN'97*, Springer LNCS, Nr. 1327, pages 595–600, 1997.

[3] N. Kambhatla and T.K. Leen. Fast non-linear dimension reduction. In *Advances in Neural Information Processing Systems, NIPS 6*, pages 152–159, 1994.

[4] H. Murase and S. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[5] T. Poggio and F. Girosi. Networks for approximation and learning. *Proc. of the IEEE*, 78(9):1481–1497, 1990.

[6] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C - The Art of Scientific Computing.* Cambridge University Press, 1988.

[7] H. Ritter, T. Martinetz, and K. Schulten. *Neuronale Netze.* Addison-Wesley, 1991.