Neural Procesing Letters, Vol.7, No.3, pp 139-149, Kluwer Academic Press (1998).

Local Adaptive Subspace Regression

Sethu Vijayakumar [*]
Department of Computer Science, Graduate School of Information Science, Tokyo Institute of Technology, Meguro-ku, Tokyo-152, Japan
Email: sethu@cs.titech.ac.jp

Stefan Schaal [†]
Department of Computer Science and Neurobiology, University of Southern California, Los Angeles, CA 90089-2520, USA.
Email: sschaal@erato.atr.co.jp

Abstract. Incremental learning of sensorimotor transformations in high dimensional spaces is one of the basic prerequisites for the success of autonomous robot devices as well as biological movement systems. So far, due to sparsity of data in high dimensional spaces, learning in such settings required a significant amount of prior knowledge about the learning task, usually provided by a human expert. In this paper we suggest a partial revision of the view. Based on empirical studies, we observed that, despite being *globally* high dimensional and sparse, data distributions from physical movement systems are *locally* low dimensional and dense. Under this assumption, we derive a learning algorithm, Locally Adaptive Subspace Regression, that exploits this property by combining a dynamically growing local dimensionality reduction technique as a preprocessing step with a nonparametric learning technique, locally weighted regression, that also learns the region of validity of the regression. The usefulness of the algorithm and the validity of its assumptions are illustrated for a synthetic data set, and for data of the inverse dynamics of human arm movements and an actual 7 degree-of-freedom anthropomorphic robot arm.

Keywords: sensorimotor map, locally weighted regression, dimensionality reduction, nonparametric learning

1. Introduction

One of the outstanding characteristics of biological systems is their ability to learn, in particular, to learn incrementally in real-time from a multitude of sensory inputs. Despite progress in artificial neural network learning, statistical learning, and machine learning, we are still far away from equipping an artificial system of even moderate complexity with a "black-box" learning system that can perform as autonomously and robustly as the biological counterpart. Among the most basic ingredients that are missing in most learning approaches are three critical

[*] Also with the ATR Human Information Processing Research Labs, Seika-cho, Kyoto 619-02, Japan

[†] Also with ERATO Kawato Dynamic Brain Project, 2-2, Hikaridai, Seika-cho, Kyoto-619-02, Japan

components. First, a learning system should possess the ability to learn continually from incrementally arriving data without the danger of forgetting useful knowledge from previously incorporated data, an effect called catastrophic interference. Second, the system has to automatically allocate the appropriate number of resources, e.g., hidden units in a neural network, to represent the learning problem at hand without the undesirable effects of overfitting or oversmoothing. And third, the learning system must be able to deal with a large number of inputs that are possibly redundant or irrelevant.

In this paper we will address these goals in the context of learning sensorimotor transformations, as needed, for instance, in the control of biological or robotic movement systems. From a statistical point of view, this involves approximating a functional relationship $f : R^N \rightarrow R^M$ from N inputs to M outputs. A typical example is to learn the inverse dynamics model of a robot, a highly nonlinear map that relates joint positions, velocities, and accelerations to appropriate joint torques. Previous research (Atkeson,1989; Atkeson, Moore & Schaal (in press)) has shown that nonparametric local learning techniques offer a favorable solution to learning such tasks with respect to the bias/variance dilemma of model selection (Geman, Bienenstock & Dursat,1992) and problems of negative interference. However, nonparametric learning techniques that depend on the notion of “neighborhood” generally scale unfavorably to high dimensions. The reason for this behavior comes from the non-intuitive effect that in high dimensional spaces, e.g., 20-dimensional, all data points are approximately the same distance away from each other (Scott,1992), thus destroying the discriminative power of neighborhood relations.

Given this “curse of dimensionality”, nonparametric learning systems – and actually all other general nonlinear learning systems – seem to have limited merits for sensorimotor control. However, when one examines data distributions of high dimensional data sets generated from real physical systems, one often notices that *locally* such data is not high dimensional at all and rarely exceed 5-8 dimensions. It is this observation which motivates the approach suggested in this paper. Despite the fact that previously developed learning techniques are theoretically able to exploit such low dimensional distributions, they quickly become computationally infeasible and also tend to be numerically less robust. This effect is due to only exploiting the low dimensional distributions *implicitly* by regularization techniques, for instance, ridge regression (Atkeson, Moore & Schaal (in press)). However, if we can exploit the low dimensional distributions explicitly by performing a *local* dimensionality reduction of the data before we apply our nonparametric learning techniques, we should be able to extend nonparametric learning and its favorable incremental learning properties to high dimensional spaces within acceptable computational costs.

To pursue this line of thought, Section 2.1 will first discuss local dimensionality reductions. Afterwards, Section 2.2 outlines incremental locally weighted regression, and Section 2.3 introduces a complete algorithm for local learning in high dimensional space, Locally Adaptive Sub-Space Regression (LASS). Finally, Section 3 demonstrates the properties of LASS using synthetic data, behavioral data from human psychophysical experiments, and data from an actual 7-degree-of-freedom anthropomorphic robot arm in order to perform function approximation in up to 21-dimensional spaces.

2. Locally Adaptive Subspace Regression

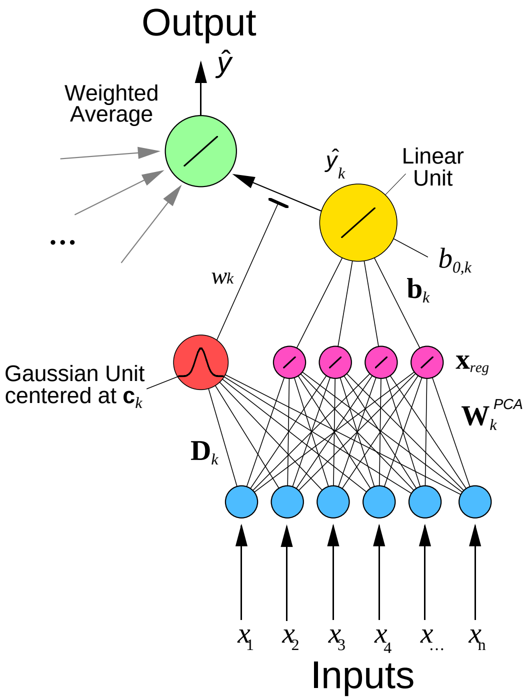


Figure 1. Illustration of the information processing stages of LASS.

The assumed underlying statistical model of our problems is the standard regression model $y = f(\boldsymbol{x}) + \epsilon$, where x denotes the N dimensional input vector, y, for the sake of clarity, a scalar output, and ϵ the additive mean-zero noise term. LASS consists of an automatically adjusting number of elements, each of which is processing data in the same way. The different stages and additional notation of the information flow of one element in the LASS system are shown in Fig.1. In essence, the input is transformed into a predicted output $\hat{y_k}$ by two linear transformations, $\mathbf{W}_k^{PCA}$ and $\mathbf{b}_k$. $\mathbf{W}_k^{PCA}$ performs the dimensionality reduction and $\mathbf{b}_k$ fits a (hyper)plane to the reduced data. Additionally, a weight w_k is calculated with the help of the connections $\mathbf{D}_k$. Based on a Gaussian activation function, the weight, w_k, is computed for every training data point $(\mathbf{x}, y)$ as

$$w_k \;=\; exp(-\frac{1}{2}(\mathbf{x}-\mathbf{c}_k)^T\mathbf{D}_k(\mathbf{x}-\mathbf{c}_k)), \tag{1}$$
$$\text{where } \mathbf{D}_k = \mathbf{M}_k^T\mathbf{M}_k \text{ for positive definiteness.}$$

w_k indicates how much this LASS element should contribute to the total prediction of the entire system. The center of the Gaussian $\mathbf{c}_k$ is assigned at the time of creation of the LASS unit and remains stationary. The matrix $\mathbf{D}_k$, referred to as the distance metric, determines the size and shape of the "receptive field" created by (1). The total prediction $\hat{y}$ of the entire LASS system results from the weighted average of the individual predictions $\hat{y}_k$ of all the K elements:

$$\hat{y} = (\sum_{k=1}^{K} w_k\hat{y}_k)/(\sum_{k=1}^{K} w_k). \tag{2}$$

From now on, we will drop the subscript k since every LASS elements learns independently of every other one and is updated by the same formulae.

2.1. Locally Weighted Dimensionality Reduction

Various candidates can be considered for dimensionality reduction in order to exploit locally low dimensional data distributions. From choices ranging from principal component analysis (PCA), independent component analysis (ICA), partial least squares (PLS) and factor analysis, we employ locally weighted PCA (LWPCA). Frank & Friedman (1993) have shown that although Principal Component Regression (PCR) performs dimensionality reduction only in the input space, it is quite competitive with more sophisticated statistical dimensionality reduction techniques. Moreover, LWPCA offers a good compromise in terms of computational feasibility and numerical robustness.

The goal of the LWPCA preprocessing stage is to locally project the original N dimensional input $\mathbf{x}$ into a L dimensional subspace – a subspace which accounts for the most local variance of the input data up to a user defined threshold θ_{PCA}:

$$\mathbf{x}_{reg} \;=\; \mathbf{W}^{PCA}\mathbf{x}_{mz}, \tag{3}$$

where $\mathbf{x}_{mz} = \mathbf{x} - \bar{\mathbf{x}}$ denotes the mean subtracted input data.

For a batch operation, this dimensionality reduction can be handled by performing a SVD decomposition on the weighted covariance matrix of the input data. For implementing an incremental LWPCA, we can minimize the following weighted cost criterion in the spirit of Minimum Description Length (MDL) (Rissanen,1989):

$$J_1 = \frac{1}{2}\sum_{i=1}^{p} w_i \|\mathbf{x}_{reconst,i} - \mathbf{x}_{mz,i}\|^2, \text{ where } \mathbf{x}_{reconst} = \mathbf{W}^{PCA^T}\mathbf{x}_{reg} \tag{4}$$

The minimization of (4) is achieved by gradient descent with learning rate η

$$W_{ij}^{PCA^{n+1}} = W_{ij}^{PCA^n} + \eta \frac{\delta J_1}{\delta W_{ij}^{PCA}}, \tag{5}$$

where

$$\frac{\delta J_1}{\delta W_{ij}^{PCA}} = w x_{reg,i}(\sum_{r=1}^{i} x_{reg,r} W_{ij}^{PCA^n} - x_{mz,j}). \tag{6}$$

and corresponds to a weighted version of the incremental PCA algorithm of Oja(1982) and Sanger(1989). In order to speed up learning, we use a second order gradient descent minimization of (4) based on Sutton (1992), explanation of which we defer due to space limitations.

2.2. Locally Weighted Regression and Distance Metric Adaptation

Each of the LASS elements performs a local regression on the projected reduced dimensional data as $\hat{y} = \mathbf{x}_{reg}^T\mathbf{b} + b_0 = \tilde{\mathbf{x}}^T\beta$, where $\tilde{\mathbf{x}} = (\mathbf{x}_{reg}^T, 1)^T$. An incremental estimate of β can be formed by recursive least squares (Schaal & Atkeson,1996). However, it should be noted that the LWPCA pre-processing step does not only yield a computational advantage in terms of providing a significantly reduced input to the regression; LWPCA also decorrelates the dimensions of $\mathbf{x}_{reg}$ Thus, the regression decomposes into $L + 1$ univariate additive regressions (Hastie & Tibshirani,1990), reducing the computational complexity of the regression from being quadratic in the number of regression inputs to being linear:

$$\beta_i^{n+1} = s_{xy,i}^{n+1}/s_{xx,i}^{n+1}, \text{ where } s_{xy,i}^{n+1} = \lambda s_{xy,i}^{n} + w\tilde{x}_i y \text{ and } s_{xx,i}^{n+1} = \lambda s_{xx,i}^{n} + w\tilde{x}_i{}^2. \tag{7}$$

The variable λ in (7) denotes a forgetting factor, a standard technique in recursive estimation (Ljung & Soederstroem,1986), such that initial input to the regression, stemming from a LWPCA which has not properly converged yet, will not negatively influence the regression result in the long run.

The linear model represented by each LASS unit is valid only in the locality specified by the distance metric $\mathbf{D}$. In order to adapt to local differences in the spatial frequency of the outputs, we optimize the size and shape of the local models by adapting $\mathbf{D}$. This process can be accomplished by incrementally minimizing a local cost criterion:

$$\begin{aligned} J_2 &= \frac{1}{\sum_{i=1}^{p} w_i} \sum_{i=1}^{p} w_i \left\| y_i - \hat{y}_{i,-i} \right\|^2 + \gamma \sum_{n,m} D_{n,m}^{reg^2}, && (8) \\ & \quad \text{where} && (9) \\ \mathbf{D}^{reg} &= \mathbf{W}^{PCA} \mathbf{D} \mathbf{W}^{PCA^T} \end{aligned}$$

Here, the first term corresponds to a weighted mean squared cross validation measure, and the second term corresponds to a bias on the magnitude of the second derivatives of the output. The update of the distance metric $\mathbf{D}$ is a gradient descent in J_2, detailed in Schaal & Atkeson (1997). It should be noted, however, that the gradient computation for (8) is much simpler than in Schaal & Atkeson (1997) due to orthogonality of the outputs from the PCA pre-processing.

2.3. The LASS Algorithm

The LASS algorithm proceeds as following. The entire system is initialized with no processing element. Every piece of training data $(\mathbf{x}, y)$ is used to update all the existing elements. If no element is activated (cf. Equation 1) more than a threshold w_{gen}, a new LASS element is created with its receptive field center $\mathbf{c}$ in (1) initialized to $\mathbf{c} = \mathbf{x}$. The distance metric $\mathbf{D}$ is initialized to a user supplied value $\mathbf{D}_{def}$. The initial dimensionality of the LWPCA starts out with $L = 2$, although the regression will only use $L - 1$ inputs. An incrementally adapting mechanism increases the dimensionality of the regression stage based on a variance threshold criterion: if the condition $\mathbf{v}_{PCA,L} / \sum_{i=1}^{L} \mathbf{v}_{PCA,i} > \theta_{PCA}$ is satisfied, the dimensionality L is incremented by one and appropriate coefficients are added in $\mathbf{W}^{PCA}$ and $\mathbf{b}$. For this purpose, each LASS unit keeps an incremental record of the variances of its L LWPCA outputs:

$$\mathbf{v}_{PCA}^{n+1} = (\lambda W^n \mathbf{v}_{PCA}^n + w \mathbf{x}_{reg}^2) / (W^n + w) \qquad (10)$$

Note that the learning rule (5) guarantees that the variances $\mathbf{v}_{PCA}$ are in descending order (Sanger,1989) such that only the L-th variance has to be monitored. To avoid premature adding of dimensions, it is useful to also monitor the rate of change of the variances and add dimensions only if the rate of change is close to zero.

It is important to note that adding dimensions does not disturb the results obtained by the previously trained LASS parameters. A new dimension in the LWPCA adds a row to $\mathbf{W}^{PCA}$, but the learning rule (5) ensures that updates of coefficients of $\mathbf{W}^{PCA}$ are not affected by coefficients whose row index is larger. Thus, the new row is trained entirely independently. Similarly, adding a coefficient to $\mathbf{b}$ just adds a new element to an additive regression. As shown in (7), the regression updates for each dimension are independent of each other due to the decorrelation of $\mathbf{x}_{reg}$ in the LWPCA. Due to ordering of the variances, it can also be argued statistically that regression coefficients with a small $\mathrm{var}(\mathbf{x}_{reg})$ have a low prior probability of contributing significantly to the regression output, since the confidence in a regression coefficient depends inversely proportional on $\mathrm{var}(\mathbf{x}_{reg})$. All these facts contribute positively to an incremental mechanism with minimal interference.

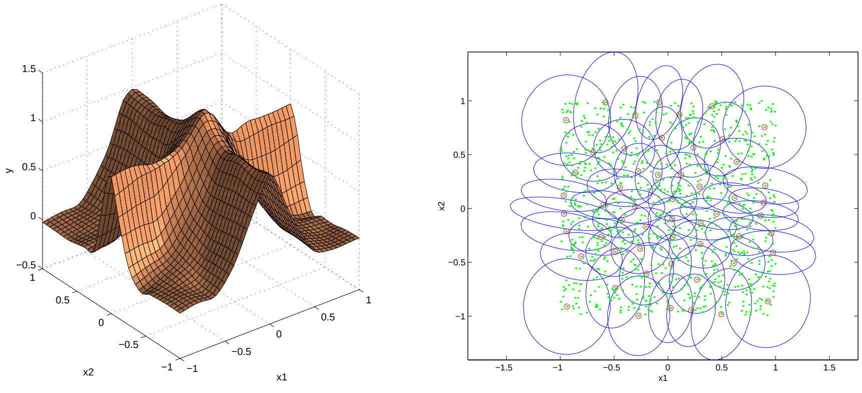
In sum, LASS is a constructive learning algorithm in two different senses. First, LASS elements are added whenever a training point in input space does not sufficiently activate any existent LASS element. This process will guarantee that the entire input distribution of the training data is quickly covered by LASS. Second, within each LASS element the dimensionality of the regression stage can grow until the LWPCA models a user specified fraction of the local variance of the inputs. The size and shape of the local region of validity of a LASS element, however, is determined by a goodness of fit criterion in regression space, thus coupling the choice of a local subspace to the regression stage, not unlike the classification algorithm of (Kambhatla & Leen, 1995). These features ensures the quality of the regression result. Since The adjustable parameters in each LASS element are the LWPCA weights $\mathbf{W}^{PCA}$, the regression parameters β and the distance metric $\mathbf{D}$, all of which are trained with second order learning techniques.

3. Empirical Results

In the first example we will use a synthetic data set that allows to illustrate function fitting results with LASS graphically. The task is to approximate

$$y = \max\{\exp(-10x_1^2), \exp(-50x_2^2, 1.25\exp(-5(x_1^2 + x_2^2)))\} + \mathrm{N}(0, 0.01)$$

from noisy data set of 500 samples, drawn uniformly from the unit square. This function consists of a narrow and a wide ridge which are



(a) (b)

Figure 2. (a) LASS approximation results for 10-dimensional input data set; (b) Contour lines of 0.1 iso-activation of each expert in input space.

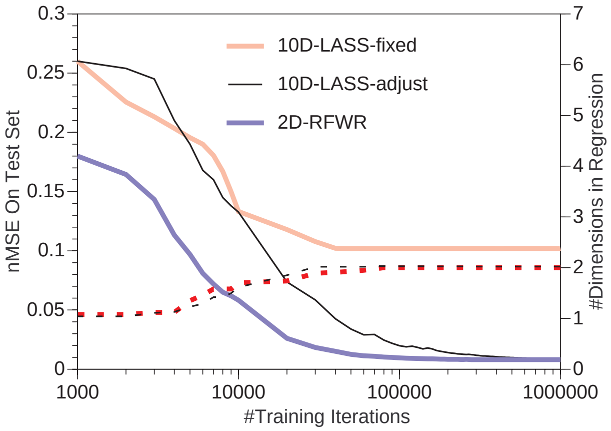


Figure 3. $nMSE$(solid lines) and dimensionality of regression(dashed lines) as function of training iterations for 10-D LASS approximation with fixed and adjusted distance metric and a baseline comparison with 2-D RFWR. One training iteration corresponds to one incremental presentation of a training sample.

perpendicular to each other, and a Gaussian bump at the origin. The test data set consists of 1681 data points corresponding to the vertices of a 41x41 grid over the unit square; the corresponding output values are the exact function values. The approximation error is measured as a normalized mean squared error, $nMSE$, i.e, the MSE on the test set normalized by the variance of the outputs of the test set. The initial parameters of LASS are set to $\mathbf{D}_{def} = 20\mathbf{I}$ ($\mathbf{I}$ is the identity matrix), $w_{gen} = 0.2$, $\theta_{PCA} = 0.03$. The PCA learning rate was $\eta = 1$, the regression learning rate $\theta = 100$ and the forgetting factor was set to $\lambda = 0.9995$. In the test for LASS, we augmented the input space by 8 additional dimensions whose values were zero, and transformed this input space by a 10 dimensional randomly chosen rotation matrix. Thus, the task of LASS was to recover this low dimensional function now embedded in a high dimensional space. As a comparison, LASS was trained with and without the adjustment of the distance metric D. Fig.3 illustrates the course of learning for both tests. It takes about 50,000 iterations until the LWPCA converges initially. For a fixed distance metric, the algorithm converged with an $nMSE = 0.1$ and the actual dimensionality of the data was correctly detected as 2. On the other hand, when we allowed the adjustment of the distance metric $\mathbf{D}$ based on the regression, an $nMSE = 0.01$ was achieved, with the LWPCA again saturating the regression dimensions employed at 2. Fig.2a shows a typical example of the excellent reconstruction of the function after rotating the results back into the original low dimensional space. Fig.2b shows the size and orientation of the receptive fields that the system created during learning. As can be noticed, each LASS element modifies it‘s region of locality based on the function's curvature, hence, shrinking along high frequency directions while stretching over largely linear areas. The $nMSE$ achieved by LASS was indistinguishable to the results achieved in 2-dimensional approximation of the same function by RFWR (see Fig.3), a robust local learning technique for low dimensional non-parametric regression (Schaal & Atkeson, 1996). It should also be noted that the $nMSE$ starts at a fairly low value after only 1,000 iterations, despite the system only employing one dimensional regressions at this point.

In the second evaluation, we use real movement data collected from human arm movements in random point-to-point reaching and rhythmic scribbling tasks. Kinematic arm data was recorded with optical recording equipment. From this data, the 7 joint positions of the arm and their associated velocities and accelerations were recovered. Under the assumption of a rigid body dynamics model, corresponding joint torques were computed using biologically plausible values for inertia and mass parameters. LASS was trained to fit the inverse dynamics model of this data, a mapping from the joint positions, velocities, and accelerations to the joint torques (a 21 to 7 dimensional function). This test was intended to determine whether a successful nonparametric approximation of the rigid body inverse dynamics, given the measured input distribution from the human subjects, could be obtained in locally reduced subspaces. Results of this evaluation showed that LASS achieved an $nMSE$ of 0.04 on test sets by employing only

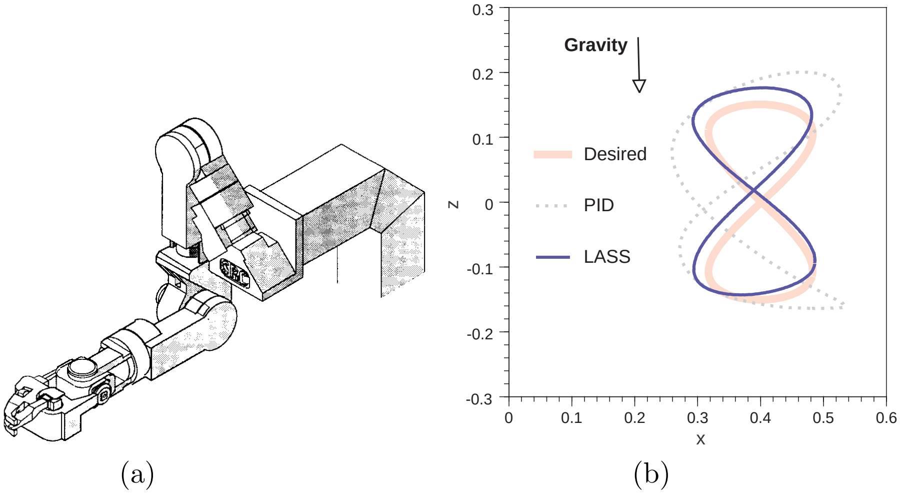


Figure 4. (a) Sketch of SARCOS Dexterous Arm (b) Trajectory tracking comparison of the SARCOS arm using a PID controller against using a non-parametric inverse dynamics model learned by LASS.

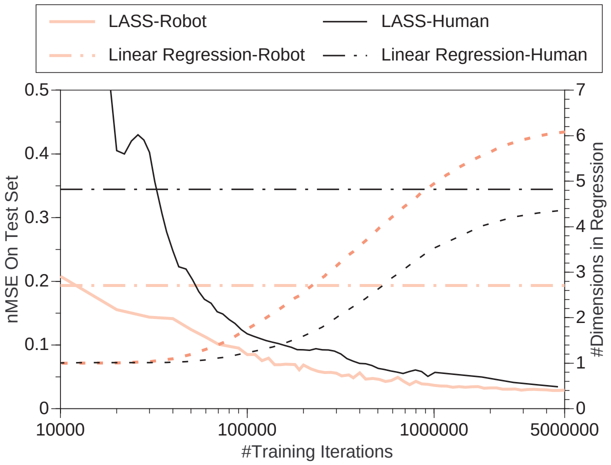


Figure 5. Average dimensionality and typical learning curves for LASS in real movement data of humans and robots.

between 4-5 dimensions on the average for the regression (see Fig.5). This strongly supports our assumption that real movement data are *locally* low dimensional.

However, an authoritative statement about having successfully learned an inverse dynamics model should not solely be made based on $nMSE$ values alone; the true test is, of course, success in motion synthesis and performance on actual trajectory tracking. Therefore, in our final evaluation, we approximated the inverse dynamics model of a 7-degree-of-freedom anthropomorphic robot arm (Sarcos Dexterous Arm, Fig.4a) from a data set consisting of 45,000 data points, collected at 100Hz from the actual robot performing various rhythmic and discrete movement tasks. The data was randomly split into half to obtain a training set and testing set. Fig.5 shows the learning results in comparison to a linear regression model. Already after 10,000 iteration, LASS performed as good as the linear regression model despite employing only 1 regression dimension at this stage. After about 100,000 iterations (roughly 4 passes through the training set or 15 minutes of real-time data), LASS already accomplished a $nMSE$ (averaged over all 7 output dimensions) of under 0.05. The system converges at $nMSE = 0.03$ while employing an average of 6 dimensions locally, once again confirming our hypothesis that physical systems realize locally low dimensional data distributions. Finally, we used the inverse dynamics model learned by LASS in a trajectory following task of drawing an 'eight' in Cartesian coordinates. In comparison to a low gain, biologically inspired PID controller on the same task, the LASS based controller improved tracking significantly (see Fig.4b) although a slight constant offset due to incomplete gravity compensation can be noticed.

4. Discussion

The goal of this paper is to emphasize one major point, i.e, local learning for regression in high dimensional spaces may not be as complicated as previously thought. The rationale for this statement is based on the assumption that data distributions, despite being *globally* high dimensional, are *locally* often of only low dimensional structure. Evaluations of the real movement data provided strong support for this assumption. Based on this, we developed a nonparametric learning algorithm which is targeted to make use of such locally low dimensional distributions. Our learning system, Locally Adaptive SubSpace regression (LASS), preprocesses data by a local principal component analysis (LWPCA) which is capable of adapting dynamically to the input data distribution. Besides learning a locally weighted regression model(LWR) based on the preprocessed data, LASS also automatically adjusts the region of locality where the local linear model is valid. For a synthetic and an actual human and robot data sets, we illustrated that LASS achieved the expected performance: in all cases, locally low dimensional data distributions were detected and exploited appropriately. In contrast to our previously developed learning methods whose computational complexity is more than quadratic, LASS scales linearly to the number of input dimensions.

An open point of research concerns how the regression analysis could influence the LWPCA. At the current stage of LASS, LWPCA proceeds independently of LWR, which, from a statistical point of view, is not satisfying as the quality of the regression depends on the distribution of the input data (Schaal & Atkeson,1997). Empirical evaluations will provide insight into how much this shortcoming affects the quality of learning. Performing LWPCA in joint data space, or employing alternative techniques, like partial least squares regression, have shown promising results in some pilot studies.

Acknowledgements

Support for S.Vijayakumar and S.Schaal was partially provided by the ATR Human Information Processing Research Laboratories and the ERATO Dynamic Brain Project. This work was funded in parts by grants to S.Vijayakumar provided by the Japanese Ministry of Education, Science and Culture (Monbusho) and to S.Schaal provided by the German Research Association, the Alexander von Humboldt Foundation and the German Scholarship Foundation.

References

An,C.H., Atkeson.C.G. & Hollerbach.J.M.(1988), *Model based control of a robot manipulator*, MIT Press, Cambridge, MA.

Atkeson.C.G.(1989), Using local models to control movement, In:Touretzky,D.(Eds.), *Advances in Neural Information Processing Systems 1,* San Mateo, CA:Morgan Kauffman.

Atkeson,C.G., Moore,A.W. & Schaal,S.(in press)a, Locally weighted learning for control, *Artificial Intelligence Review.*

Atkeson,C.G., Moore,A.W. & Schaal,S.(in press)b, Locally weighted learning, *Artificial Intelligence Review.*

Cleveland,W.S.(1979), Robust locally weighted regression and smoothing scatterplots, *Journal of the American Statistical Association* vol.74, pp.829-836.

Geman,S., Bienenstock,E. & Doursat,R.(1992), Neural networks and the bias-variance dilemma, *Neural Computation*, No.4, pp.1-58.

Hastie,T.J. & Tibshirani,R.J.(1990) *Generalized additive models*, London:Chapman-Hall.

Kambhatla,N. & Leen,T.K.(1994), In: Touretzky,D.S., Mozer,M.C. & Hasselmo,M.E.(Eds.) *Advances in Neural Information Processing Systems 6,* San Fransico, CA:Morgan Kaufmann Publishers.

Ljung,L. & Soederstroem,T.(1986) *Theory and Practice of Recursive Identification*, Cambridge, MIT Press.

Oja,E.(1982) A simplified neuron model as a principal component analyzer, *Journal of Mathematical Biology* Vol.15, pp.267-273.

Rissanen,J.(1989) *Stochastic complexity in statistical enquiry*, Singapore:World Scientific.

Sanger,T.D.(1989), Optimal unsupervised learning in a single layer linear feedforward neural network, *Neural Networks*, Vol.2, pp.459-473.

Schaal,S. & Atkeson,C.G.(1996) From isolation to cooperation : An alternative view of a system of experts, In: Touretzky,D.S., Mozer,M.C. & Hasselmo,M.E.(Eds.) *Advances in Neural Information Processing Systems 8,* Cambridge, MA: MIT Press.

Schaal,S. & Atkeson,C.G.(1997), Receptive field weighted regression, *Technical Report TR-H-209, ATR Human Information Processing Labs., Kyoto 619-02, Japan.*

Schaal,S.(in press), Learning from demonstration, *Advances in Neural Information Processing Systems 9.*

Scott,D.W.(1992), *Multivariate Density Estimation*, New York:Wiley.

Sutton,R.S.(1992), Adapting bias by gradient descent: An incremental version of Delta-Bar-Delta, *Proc. Tenth National Conf. Artificial Intelligence* pp.171-176.

Witten,I.H., Neal, R.M. & Cleary,J.G.(1997), Arithmetic coding for data compression, *Communications of the ACM* , Vol.30, pp.520-540.