LOCAL LEARNING IN LOCAL MODEL NETWORKS

Roderick Murray-Smith†, Tor Arne Johansen‡

†Daimler-Benz AG, Alt-Moabit 96a, D-10559 Berlin, Germany, murray@DBresearch-berlin.de
‡SINTEF Automatic Control, N-7034 Trondheim, Norway, Tor.Arne.Johansen@regtek.sintef.no

Abstract

Local Model Networks are hybrid models which allow the easy integration of *a priori* knowledge, as well as the ability to learn from data to represent complex, multidimensional dynamic systems from data. This paper points out problems with global learning methods in Local Model Networks. The bias/variance trade-offs for local and global learning are examined, and it is illustrated that local learning has a regularizing effect that can make it favorable compared to global learning in some cases.

1 Local Model Networks

The basic assumption underlying the use of learning systems for modelling purposes is that the behaviour of the system can be described in terms of a training set $\mathcal{D}_N = ((\psi(1), y(1)), ..., (\psi(N), y(N)))$ consisting of its observed input vector ψ and corresponding scalar output y. We assume the system output can therefore be modelled as

$$y = f(\psi) + \varepsilon. \tag{1}$$

where f is a function, and ε is independent random measurement noise with zero mean and variance σ^2.

The modelling problem, as seen in this paper is to robustly estimate the function f from observation data, having already used existing *a priori* information to pre-structure and parameterise the model structure $\hat{f}$. Typically, the model structure will not be able to exactly describe the system, so a bias $b(\psi)$ will naturally be associated with $\hat{f}$:

$$y = \hat{f}(\psi, \theta^\star) + b(\psi) + \varepsilon \tag{2}$$

We define the optimal p-dimensional parameter vector $\theta^\star$ for the learning problem in equation (2) such that the average bias is minimised over the domain of interesting inputs

$$\theta^\star \quad = \quad \arg\min_\theta \sum_{k=1}^{N} \left(f(\psi(k)) - \hat{f}(\psi(k), \theta) \right)^2$$

Unfortunately, since f is not known, we cannot find these optimal parameters, but must look for an estimate based on the data $\mathcal{D}_N$.

The model function $\hat{f}$ can in general be pre-structured and parameterized in a number of ways. With a linear model, the data, learning and validation are all considered to be globally (i.e. over the entire input domain) relevant. For non-linear models, however, it may be advantageous to partition the input domain into multiple subsets – a strategy inherent to local modelling techniques. Such local model representations have therefore seen an upsurge in interest in the form of Radial Basis Function (RBF) Nets, Spline Networks (Kavli, 1992), Modular Networks (Jacobs et al., 1991) and Fuzzy Systems (Takagi and Sugeno, 1985). This locality can be used to make models more interpretable and computationally efficient.

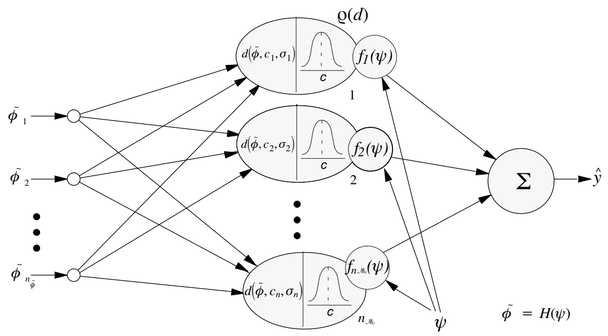


Figure 1. Local Model Network

The *Local Model Network*

$$\hat{f}(\psi, \theta) = \sum_{i=1}^{n_\mathcal{M}} \hat{f}_i(\psi, \theta_i)\rho_i(\tilde{\phi}), \tag{3}$$

can be viewed as an RBF network where the basis function coefficients have been generalised to allow not just a constant parameter to be associated with each basis function, but a more powerful function of the inputs ψ. This means that a smaller number of local models can cover larger areas of the input domain. The parameter vector is $\theta^T = [\theta_1^T, ..., \theta_{n_\mathcal{M}}^T]$, and $\tilde{\phi}$ defines the *operating point* of the system, usually given by a function $\tilde{\phi} = H(\psi)$. This is a vector which often can be defined on a lower dimensional subspace of the input space which is covered by the model's basis functions. These can be seen as *gating* or *weighting* functions for the local models (which are defined on the full input space). The basis functions used are defined,

$$\rho_i(\tilde{\phi}) = \frac{\rho\left(d\left(\tilde{\phi}, c_i, \sigma_i\right)\right)}{\sum_{j=1}^{n_{\mathcal{M}}} \rho\left(d\left(\tilde{\phi}, c_j, \sigma_j\right)\right)} \quad (4)$$

where $\rho(\cdot)$ is the underlying *unnormalised* basis function, e.g. a Gaussian $\rho(d) = \exp(-d^2/2)$, and

$$d\left(\tilde{\phi}, c_i, \sigma_i\right) = \sqrt{(\tilde{\phi} - c_i)^T \sigma_i^{-2} (\tilde{\phi} - c_i)} \quad (5)$$

is a weighted Euclidean distance metric which measures the distance of the current operating point $\tilde{\phi}$ from the basis function's centre c_i. The *normalised* basis functions $\rho_i(\cdot)$ now sum to unity. The local models used are linear:

$$\hat{f}_i(\psi, \theta_i) = \left[1,\ \psi^T\right] \theta_i. \quad (6)$$

The basic local model network structure (3)-(6) was suggested in (Jones et al., 1989), followed up by (Stokbro et al., 1990), (Barnes et al., 1991) and (Johansen and Foss, 1993). Reviews of applications and research can be found in (Johansen, 1994, Murray-Smith, 1994). The Adaptive Expert networks in (Jacobs et al., 1991) are essentially also a probablistic interpretation of local model systems. (Priestley, 1988) describes State Dependent Models which are similar to Local Model Networks. Certain Fuzzy Systems can also be viewed as Local Model Networks (Foss and Johansen, 1993).

In this paper we consider the learning of the parameters of the local models for a given model structure, i.e. we are estimating θ for an a priori given set of c, σ. We recognize that the problem of pre-structuring or learning the structure of the local model network is perhaps the more important and challenging one, but as we shall see, the parameter learning problem has some interesting characteristics that are closely related to the structuring of the local model network. This justifies a closer look at parameter learning problem.

First, we briefly review the standard solution to this problem, using a criterion that penalizes mismatch between the training data and the global model output. Thereafter, we describe an alternative solution based on a number of weighted criteria that penalizes mismatch between the data and the local model outputs. Finally, the properties of these two different algorithms are discussed.

The general concept of local learning algorithms was suggested in the context of non-parametric models (Nadaraya, 1964),(Watson, 1964), (Benedetti, 1977) and (Cleveland et al., 1988). The parametric model form, as considered here, is seen in (Bottou and Vapnik, 1992). The contribution of this paper is an analysis of such an algorithm in the context of local model networks, summarising and extending some results in (Murray-Smith, 1994).

2 Local and Global Learning Algorithms

2.1 Global Learning

The local models are assumed linear in the parameters, as in equation (6), so the learning problem is a straightforward application of linear regression techniques to find the parameters θ which best matche the data. Stacking the data into matrices, we get the following regression model:

$$Y \quad = \quad \Phi\theta^{\star} + B + \mathcal{E}, \quad (7)$$

where Φ is the design matrix, the rows of which are defined by

$$\phi_k = \left[\rho_1(\tilde{\phi}(k))[1,\ \psi^T(k)], \ldots, \rho_{n_{\mathcal{M}}}(\tilde{\phi}(k))[1,\ \psi^T(k)]\right], \quad (8)$$

so that the design matrix Φ, vector of output measurements Y, vector of biases B, and errors $\mathcal{E}$ are

$$\Phi = \left(\ \phi_1^T, ..., \phi_N^T\ \right)^T, \qquad Y = \left(\ y(1), ..., y(N)\ \right)^T$$
$$B = \left(\ b(1), ..., b(N)\ \right)^T, \qquad \mathcal{E} = \left(\ \varepsilon(1), ..., \varepsilon(N)\ \right)^T$$

The standard least squares criterion for this estimation problem is

$$J(\theta) = \frac{1}{N}(Y - \Phi\theta)^T (Y - \Phi\theta). \quad (9)$$

and the Moore-Penrose pseudoinverse of Φ, Φ^+ is used to estimate the weights:

$$\hat{\theta}_{LS} = \Phi^+ Y = (\Phi^T\Phi)^{-1}\Phi^T Y. \quad (10)$$

The numerical algorithm used in the examples[1] to calculate the pseudoinverse is the *Singular Value Decomposition* (SVD). The SVD algorithm decomposes any $N \times p$ matrix Φ, such that $\Phi = USV^T$ and the pseudoinverse of Φ is:

$$\Phi^+ = VS^+U^T. \quad (11)$$

Once the singular values have been zeroed, the parameters solving the regression problem in equation (7) can be calculated.

$$\hat{\theta}_{LS} = VS^{-1}U^T Y. \quad (12)$$

2.2 Local learning

The global learning approach is based on the assumption that all of the parameters θ would be learned in a

[1] We used the MATLAB function `svd()`. For more details on SVD see the general treatment in books such as (Golub and van Loan, 1989), or (Press et al., 1988). The method is robust because it can, within limits, efficiently cope with singular or poorly conditioned matrices. Diagonal elements of S that are less than a preset tolerance, are zeroed, effectively reducing the degrees of freedom in the model.

single regression operation. This may not always be computationally feasible if a large number of training patterns or local models are needed for a particular problem (see Section 2.3). Perhaps more worrying, the global nature of the learning also means that the parameters of the local models cannot be interpreted independently of neighbouring local models, which means that they cannot be seen as local approximations to the underlying system. An alternative to global learning which is less prone to these disadvantages is to locally estimate the parameters of each of the local models (as defined in equation (6)) independently[2]. The parameters of the local models are then estimated using a set of local estimation criteria for the i-th local model

$$J_i(\theta_i) = \frac{1}{N}\left(Y - \Phi_i\theta_i\right)^T Q_i \left(Y - \Phi\theta_i\right), \qquad (13)$$

where $i = 1, ..., n_{\mathcal{M}}$. Q_i is an $N \times N$ diagonal weighting matrix, where the diagonal elements are weights $\alpha_i(\psi(1)), ..., \alpha_i(\psi(N))$, which are used to weight the importance of the different samples in the training set on the i-th local model. To achieve local learning it is necessary to define a set of local criteria[3]. Our confidence in a given observation regarding its relevance for the i-th local model is directly reflected in the i-th basis function. A plausible local weighting function is therefore $\alpha_i(\psi) = \rho_i(\tilde{\phi})$, which results in

$$Q_i = \mathrm{diag}\left(\rho_i(\tilde{\phi}(1)), ..., \rho_i(\tilde{\phi}(N))\right). \qquad (14)$$

In this case the locally weighted least squares estimate of the local model parameter vector θ_i is given by the minimum of J_i. In matrix terms the operation is now

$$\begin{aligned} \hat{\theta}_{LLS} &= \left(\hat{\theta}^T_{LLS,1}, ..., \hat{\theta}^T_{LLS,n_{\mathcal{M}}}\right)^T \\ \hat{\theta}_{LLS,i} &= \left(\Phi_i^T Q_i \Phi_i\right)^{-1} \Phi_i^T Q_i Y, \quad i = 1, 2, ..., n_{\mathcal{M}} \end{aligned}$$

where Φ_i is an $N \times (n_\psi + 1)$-submatrix of Φ corresponding to the ith local model. Often, a large number of the diagonal elements in Q_i are zero, or very close to zero. For practical purposes, the matrices and vectors involved in each local learning problem can therefore be replaced by lower-dimensional equivalent sub-matrices that only contain the training data samples that are relevant for each local model. The number of such observations is denoted N_i. The local learning method is therefore to compute $n_{\mathcal{M}}$ locally weighted least squares regressions, one for each local model, using only the subset of the training data within the model's receptive field, and with only the bases related to the given local model's parameters.

[2] This assumes that the basis functions achieve a partition of unity.

[3] A side-effect of local learning is that it also allows more flexibility in the use of learning algorithms, which will be especially useful with heterogeneous local model networks which use a variety of optimisation algorithms (possibly also not linear in the parameters) are locally applied, each suited to the individual local model type.

2.3 Computational effort

One important reason for using local learning methods is that the computational effort is dramatically reduced. The effort needed to find the pseudoinverse using SVD for a $(N \times p)$ matrix[4] is roughly (Noble and Daniel, 1988),

$$O_{gl} = O\left(N^2 p + N p^2 + \min(N, p)^3\right) \qquad (15)$$

where $p = n_{\mathcal{M}}(\psi + 1) = \dim(\theta)$ is the number of parameters in the model, which is clearly the crucial factor with regard to computational effort. The local effort O_{lo} is repeated $n_{\mathcal{M}}$ times,

$$O_{lo} = O\left(n_{\mathcal{M}}\left(N_i^2 n_\psi + N_i n_\psi^2 + \min(N_i, n_\psi)^3\right)\right), \qquad (16)$$

where N_i is usually significantly smaller than N.

2.4 Experiments on a 1-D noisy function

Local and global learning may lead to considerably differing results. To show the effect of the two learning methods we use a simple one-dimensional example. The arbitrarily chosen nonlinear function is

$$y = \cos(6\psi^2) + \varepsilon(\psi), \qquad (17)$$

where the additive noise term $\varepsilon(\psi)$ is Gaussian with a varying standard deviation of $\sigma(\psi) = 0.4\exp\left(-4.6\left|\psi - \frac{1}{2}\right|\right)$. The unnormalized basis-function is the Gaussian, and $\tilde{\phi} = \psi$. The function is shown below in Figure 2. Both cases use the SVD algorithm, zeroing singular values smaller than 10^{-5}. For smaller training sets (Figure 2), the relative robustness of the local learning is immediately obvious, both in the smoother response, and in the lower error on the training data. As the amount of data increases, the global method's accuracy improves, but the 'oscillating' local models remain.

3 A closer look at Local and Global Learning

3.1 Causes for Ill-conditioning

In local model networks using global learning, even with robust identification algorithms, we sometimes observe, as in Figure 2, that the local models do not change behaviour smoothly as a function of the operating point. Also, they cannot be viewed as local approximations of the systems. This phenomenon of "oscillating" local models, where the negative contribution of one local model is compensated for by the positive contribution of the neighbouring local

[4] $p = n_{\mathcal{M}}(n_\psi + 1)$ is equivalent to the cost of a homogeneous local model net with linear local models, where n_ψ represents the dimension of the model's input space

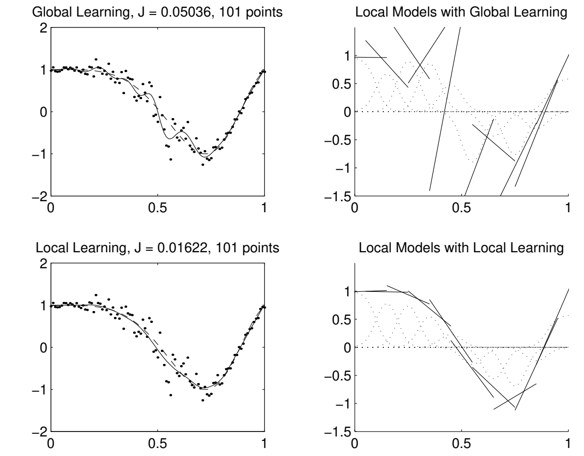


Figure 2: Experimental comparison of Global and Local Learning for 101 training points. The left hand side shows the target function, the noisy training data and the trained network's response. Local models used deviation inputs $\psi_i = \frac{\psi - c_i}{\sigma_i}$. The validation criterion in the figure titles, is the standard weighted mean square error, where the weighting function $\alpha(\psi) = \frac{1}{\sigma(\psi)}$. The validation criterion is evaluated on the model's deviation from the true function. The right hand side of each figure shows the normalised basis functions and the associated local linear models.

models, leads to a delicate balance which may minimise the global error on the training set, but need not necessarily be robust when confronted with new samples, i.e. the model generalises poorly, and is certainly more difficult to interpret. The use of such non-smooth models can also be highly disadvantageous in many applications[5]. We have identified two potential causes of such ill-conditioned behavior:

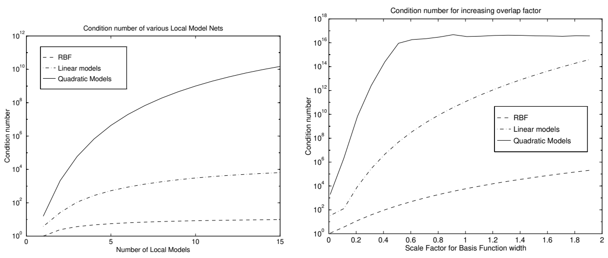
1. Even the optimal parameters $\theta^\star$ may give such behaviour, when there is a *fundamental structural mismatch* between model $\hat{f}(\cdot, \theta^\star)$ and the true system f. One example of such a structural mismatch is a discontinuous function f together with smooth basis-functions. Another example of structural mismatch is a concentration of local models in operating regions where the system has low complexity, rather than in operating regions with high complexity. The local model parameters in regions corresponding to low system complexity are effectively utilized to improve the fit in regions with high system complexity and low model complexity, which causes ill-conditioning because the corresponding basis-functions are close to zero.

2. Alternatively, the cause may be *overparameterization*, which gives an *ill-conditioned* $\Phi^T \Phi$ and a larger parameter estimate variance, which obviously may appear as large variability amoung neighbouring local models' parameters. For example, an increasing number of local models (increasing model structure complexity) will typically lead to higher condition number of $\Phi^T \Phi$ (see below), and hence, larger variance on the parameter estimate. On the other hand, the effect of an increasing number of local models on the prediction performance depends on whether the decrease in bias is more significant than the increase in variance, as we shall see later.

[5]For example, in model based predictive control, the control input is often found using gradient search methods, which, if the model is not smooth, would then be subject to many local minima and would lead to unreliable control.

3.2 Effect of Overlap factor on condition

Is the local model network structure particularly prone to ill-conditioning due to overparameterisation? The fact that the bases of a local model net are often effectively the same inputs weighted differently (see eqn. (8)), means that the higher the level of overlap in local model nets, the higher level of correlation of any model's inputs with its neighbours. The level of overlap thus becomes a major factor in the poor conditioning of the design matrix in local model nets. Reducing the overlap, however, leads to non-smooth transitions between models. Figure 3(a) shows the increase in condition number with increasing numbers of local models, when the relative overlap remains identical. To better understand the role of overlap, Figure 3(b) shows the increase in condition number in the example used earlier with a fixed number of local models (seven) when the overlap is increased.



(a) Constant relative overlap, with uniform basis functions evenly spaced, $\sigma = |c_1 - c_2|$.

(b) Varying Overlap. The x-axis shows the scaling factor for the basis function width. 1 represents $\sigma = |c_1 - c_2|$.

Figure 3: Condition number increasing with number of local models or with overlap.

The major effect of local learning on the cost functional is to remove the interaction of neighbouring local models' parameters from the design matrix.

3.3 A statistical view on ill-conditioning

In this section we investigate from a statistical perspective the observed ill-conditioning using global learning, and show that any ill-conditioning is expected to be reduced with local learning.

In order to make the following analysis simple and transparent, we assume that the inputs $\psi(k)$ in the design matrix Φ are deterministic[6]. With global learning (the least squares estimator), we have (e.g. (Ljung, 1987))

$$E\hat{\theta}_{LS} = \theta^\star + (\Phi^T\Phi)^{-1}\Phi^T B \tag{18}$$

$$E(\hat{\theta}_{LS} - E\hat{\theta}_{LS})(\hat{\theta}_{LS} - E\hat{\theta}_{LS})^T = (\Phi^T\Phi)^{-1}\sigma^2 \tag{19}$$

where E is expectation with respect to the probability distribution of ε. Moreover, we get the following bias/variance decomposition of the expected squared prediction error (e.g. (Hastie and Tibshirani, 1990))

$$\begin{aligned}
\mathrm{PSE}_{LS} &= \frac{1}{N}E\left(Y - \Phi\hat{\theta}_{LS}\right)^T\left(Y - \Phi\hat{\theta}_{LS}\right) \\
&= \frac{1}{N}E(Y - \Phi E\hat{\theta}_{LS})^T(Y - \Phi E\hat{\theta}_{LS}) \\
&\quad + \frac{1}{N}\Phi E\left(\hat{\theta}_{LS} - E\hat{\theta}_{LS}\right)\left(\hat{\theta}_{LS} - E\hat{\theta}_{LS}\right)^T\Phi^T \\
&= \frac{1}{N}B^T(I - \Phi(\Phi^T\Phi)^{-1}\Phi^T)B + \sigma^2 \\
&\quad + \frac{1}{N}\mathrm{tr}\left(E(\hat{\theta}_{LS} - E\hat{\theta}_{LS})(\hat{\theta}_{LS} - E\hat{\theta}_{LS})^T\Phi^T\Phi\right)
\end{aligned}$$

Substituting (19) into the last term, and observing that the trace of the resulting identity matrix simply reduces to its dimension, $p = \dim(\theta)$, it follows that with the least squares estimator we get, e.g. (Hastie and Tibshirani, 1990)

$$\mathrm{PSE}_{LS} = \beta^2 + \sigma^2 + \sigma^2 p/N. \tag{20}$$

The bias term

$$\beta^2 = \frac{1}{N}\mathrm{tr}\left((I - \Phi(\Phi^T\Phi)^{-1}\Phi^T)BB^T\right)$$

is the average bias, while $\sigma^2 p/N$ is the effect of the parameter estimator on the prediction. Hence, we have the well known bias/variance trade-off, which is essentially that the model variance can be reduced at the cost of increased bias by reducing the degrees of freedom in the model.

It is clear from (19) that an ill-conditioned matrix $\Phi^T\Phi$ will lead to a large variability in some directions in the parameter space[7]. However, whether the matrix $\Phi^T\Phi$ is ill-conditioned or not has no impact on the predicted squared error (PSE), cf. (20) where only the number of parameters relative to the number of training samples p/N is of importance[8].

[6] Notice that qualitatively similar results also hold with a random design matrix, although the analysis is considerably more complicated.

[7] The *condition* of the $\Phi^T\Phi$ is in general important for the robustness of the learning process. The *condition number* of a square matrix A is defined to be the ratio between its largest and smalles singular values. The larger the condition number, the larger the effect of slight changes in the matrix A on the solution of the pseudoinverse A^+. As the weights are dependent on A^+, a slight change in data would lead to different weights, so generalisation is likely to be poor.

[8] Notice that when the inputs ψ are viewed as stochastic variables, the condition of $\Phi^T\Phi$ may have some effect on the PSE.

It also follows from (20) that the amount of overlap does not affect the variance part of the PSE, while it does greatly affect the variance of the parameter estimate.

3.4 Regularising effect of local learning

The classic regularisation method as defined in *regularisation theory* (Tikhonov and Arsenin, 1977) is to extend the standard quadratic error criterion to become a cost-complexity operator, including a non-negative penalty functional which includes *a priori* information such as smoothness which makes the learning problem better conditioned[9]. Local learning is an alternative method that can be applied to eliminate, or at least strongly reduce the ill-conditioning and oscillating local model phenomena that may be caused by the reasons mentioned above. To illustrate the effect of local learning we observe that

$$E\hat{\theta}_{LLS} = \theta^\star + \tilde{\theta}_{LLS} \tag{21}$$

where the bias is defined by

$$\tilde{\theta}_{LLS} = \begin{pmatrix} (\Phi_1^T Q_1 \Phi_1)^{-1}\Phi_1 Q_1 B_{LLS,1} \\ \vdots \\ (\Phi_{n_\mathcal{M}}^T Q_{n_\mathcal{M}} \Phi_{n_\mathcal{M}})^{-1}\Phi_{n_\mathcal{M}} Q_{n_\mathcal{M}} B_{LLS,n_\mathcal{M}} \end{pmatrix} \tag{22}$$

and $B_{LLS,i}$ is defined as

$$B_{LLS,i} = B + (Q_i - I)\Phi_i\theta_i^\star + \sum_{j\neq i} Q_j\Phi_j\theta_j^\star. \tag{23}$$

Notice that the bias may tend to be larger with local learning than with global learning. To understand the effect of local learning on the variance, we examine the change in effective degrees of freedom in the model. Notice that

$$\hat{Y} = \sum_{i=1}^{n_\mathcal{M}} Q_i\Phi_i\hat{\theta}_{LLS,i} = SY$$

where the smoothing matrix S is defined by

$$S = \sum_{i=1}^{n_\mathcal{M}} Q_i\Phi_i(\Phi_i^T Q_i\Phi_i)^{-1}\Phi_i^T Q_i$$

It is straightforward to show that, (Hastie and Tibshirani, 1990)

$$\mathrm{PSE}_{LLS} = \tilde{\beta}^2 + \sigma^2 + \sigma^2\tilde{p}/N \tag{24}$$

where the average bias is redefined by

$$\tilde{\beta}^2 = \frac{1}{N}\mathrm{tr}\left((I - S)(I - S)BB^T\right)$$

[9] Many neural network learning algorithms have implicitly (often unplanned!) had a regularisation effect, in that they do not find the 'optimal' (in the least squares sense) solution to the posed optimisation problem. Methods such as weight decay, stopping learning early (Sjöberg and Ljung, 1992), network pruning, learning with noise (Bishop, 1994) are all examples of *ad hoc* attempts to produce a regularisation effect.

and the degrees of freedom are defined by

$$
\begin{aligned}
\tilde{p} &= \mathrm{tr}(SS) \\
&= \sum_{i=1}^{n_{\mathcal{M}}} \mathrm{tr}\left((\Phi_i^T Q_i \Phi_i)^{-1}\Phi_i^T Q_i Q_i \Phi_i (\Phi_i^T Q_i \Phi_i)^{-1}\Phi_i^T Q_i Q_i \Phi_i\right) \\
&= \sum_{i=1}^{n_{\mathcal{M}}} \mathrm{tr}\left((\Phi_i^T Q_i \Phi_i)^{-1}\Phi_i^T Q_i \Phi_i (\Phi_i^T Q_i \Phi_i)^{-1}\Phi_i^T Q_i \Phi_i \right.\\
&\quad -(\Phi_i^T Q_i \Phi_i)^{-1}\Phi_i^T D_i \Phi_i (\Phi_i^T Q_i \Phi_i)^{-1}\Phi_i^T Q_i Q_i \Phi_i \\
&\quad \left. -(\Phi_i^T Q_i \Phi_i)^{-1}\Phi_i^T Q_i \Phi_i (\Phi_i^T Q_i \Phi_i)^{-1}\Phi_i^T D_i \Phi_i\right) \\
&= p - \sum_{i=1}^{n_{\mathcal{M}}} \mathrm{tr}\left((\Phi_i^T Q_i \Phi_i)^{-1}\Phi_i^T D_i \Phi_i (\Phi_i^T Q_i \Phi_i)^{-1}\Phi_i^T Q_i Q_i \Phi_i \right.\\
&\quad \left. + (\Phi_i^T Q_i \Phi_i)^{-1}\Phi_i^T D_i \Phi_i\right)
\end{aligned}
$$

where $D_i = Q_i - Q_i Q_i$, which will always be positive semi-definite, and $\tilde{p} \leq p$. The degrees of freedom in the model are therefore less with local learning than with global learning. This gives a reduced variance $\sigma^2 \tilde{p}/N$, at the possible cost of an increased bias $\tilde{\beta}^2$, compared to global learning.

In the case when the overlap becomes very small, it is clear that the $\rho_k(\cdot)$ functions approach step-functions, and $Q_i Q_i \rightarrow Q_i$. Hence, it follows from the expression for $\tilde{p}$ that $\tilde{p} \rightarrow p$ as the overlap parameter σ_i goes to zero. This can also be seen intuitively, since there will be no interactions between the parameters of the different local models. In the opposite case, when the overlap parameter σ_i approaches infinity, it is easily seen that all the local models are given the same weight uniformly over the input domain, i.e. $Q_i \rightarrow I/N$, where I is the $N \times N$ identity matrix. We see from the equation for $\tilde{p}$ that $\tilde{p} \rightarrow p/N = 1 + n_\psi$. Again, this can be seen intuitively from the fact that the uniform weighting of the $n_{\mathcal{M}}$ local models effectively corresponds to only one local model that covers the whole range.

4 Interaction between model structure & learning method

In summary, the simple analysis in the previous section has shown that local learning has a regularizing effect, characterised by

1. A reduction of the degrees of freedom in the model structure.
2. Reduced variance at the cost of increased bias in the parameter estimate, compare (18) with (21).
3. Reduced variance at the cost of increased bias in the squared prediction error PSE, compare (20) with (24).

With some model structures there is a trade-off between smoothly changing local model behaviour, and small expected squared prediction error (PSE). These model structures are typically characterised by over-parameterisation or fundamental structural mismatch. If the cause of the ill-conditioning in a trained local model net is due to *overparameterisation*, local learning has the beneficial effect of reducing the variance and therefore minimising the PSE (generalisation error), since the degrees of freedom in the model structure have effectively been reduced by the implicit regularisation in local learning. If, on the other hand, there is a *fundamental structural mismatch* between the model and the underlying process, the use of local learning will lead to extra bias being introduced and if this is more significant than the reduction in variance, then local learning will lead to an increase in PSE. However, local learning *will* reduce the variability of the parameter estimate, avoiding the oscillations and leading to more transparent local models.

5 Conclusions

Global parameter optimisation methods were found to be computationally expensive, and non-robust with over-parameterised or poorly structured local model networks, leading to non-transparent local models with sometimes poor prediction performance. On the other hand, it should be stressed that when the model structure is well chosen, global learning is more accurate. Local learning is very relevant for the practical application of the architecture, as the development of a model structure based on training data will often lead to non-optimal structures, and it is important that the parameter estimation methods should robust with regards to poor model strucure. The trade-off is therefore between the use of the less variable local learning with a necessarily extended model structure, and a more powerful, expensive and variable global learning method, with a reduced model structure.

Analysis of local learning showed that it can be seen as a simple form of regularisation, meaning that the local methods often produce models with higher accuracy, and greater robustness than global learning methods. The level of overlap between local models was found to play a major role in the ill-conditioning of the learning problem, as it is the parameter that determines the amount of regularization. The analysis of individual local models as local approximations to the underlying process is in general only valid with local learning.

The analysis of the computational complexity shows that local learning is faster than global learning, where the effort increases cubically with number of models. Even ignoring the speed-up gained by the reduced number of points (as $N_i \leq N$), the local variant will be faster. The effort for local learning also increases linearly in $n_{\mathcal{M}}$ as opposed to the cube of $(n_{\mathcal{M}}(n_\psi + 1))$, which makes local learning more suitable for larger problems.

References

Barnes, C. et al. (1991). Applications of neural networks to process control and modelling. In *Artificial Neural Networks, Proceedings of 1991 Internat. Conf. Artif. Neur. Nets*, volume 1, pages p321–326.

Benedetti, J. K. (1977). On the nonparmetric estimation of regression functions. J. Royal Stat. Soc., Ser B **39**, 248–253.

Bishop, C. M. (1994). Training with noise is equivalent to Tikhonov Regularization. Submitted for publication.

Bottou, L. and Vapnik, V. (1992). Local learning algorithms. Neural Computation **4**, 888–900.

Cleveland, W. S., Devlin, S. J., and Grosse, E. (1988). Regression by local fitting. Journal of Econometrics **37**, 87–114.

Foss, B. A. and Johansen, T. A. (1993). On local and fuzzy modelling. In *3rd Int. Conf. on Industrial Fuzzy Control and Intelligent Systems*, Houston, Texas.

Golub, G. H. and van Loan, C. F. (1989). *Matrix Computations*. Johns Hopkins University Press.

Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized Additive Models*. Monographs on Statistics and Applied Probability 43. Chapman and Hall, London.

Jacobs, R. A. et al. (1991). Adaptive mixtures of local experts. Neural Computation **3**, 79–87.

Johansen, T. A. (1994). *Operating Regime Based Process Modelling and Identification*. Ph.D. Thesis, Norges Tekniske Høgskole, Trondheim, Norway.

Johansen, T. A. and Foss, B. A. (1993). Constructing NARMAX models using ARMAX models. Int. J. Control **58**, 1125–1153.

Jones, R. D. et al. (1989). Function approximation and time series prediction with neural networks. Technical Report 90-21, Los Alamos National Lab., New Mexico.

Kavli, T. (1992). *Learning Principles in Dynamic Control*. PhD thesis, University of Oslo.

Ljung, L. (1987). *System Identification — Theory for the User*. Prentice-Hall, Englewood cliffs, New Jersey, USA.

Murray-Smith, R. (1994). *A Local Model Network Approach to Nonlinear Modelling*. Ph.D. Thesis, Department of Computer Science, University of Strathclyde, Glasgow, Scotland. E-mail:murray@DBresearch-berlin.de.

Nadaraya, E. A. (1964). On estimating regression. Theory of Probability and Its Applications **9**, 141–132.

Noble, B. and Daniel, J. W. (1988). *Applied linear algebra*. Prentice–Hall Int., 3rd edition.

Press, W. H. et al. (1988). *Numerical Recipes (C): The Art of Scientific Computing*. Cambridge Press, UK.

Priestley, M. B. (1988). *Non-linear and Non-stationary Time Series Analysis*. Academic Press.

Sjöberg, J. and Ljung, L. (1992). Overtraining, regularization, and searching for minimum in neural networks. In *Proc. IFAC Symposium on Adaptivve Systems in Control and Signal Processing, Grenoble, France.*, pages 669–674.

Stokbro, K., Umberger, D. K., and Hertz, J. A. (1990). Exploiting neurons with localized receptive fields to learn chaos. Complex Systems **4**, 603–622.

Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its applications for modeling and control. IEEE Trans. on Systems, Man and Cybernetics **15**, 116–132.

Tikhonov, A. N. and Arsenin, V. Y. (1977). *Solutions of Ill-posed problems*. Winston, Washington DC.

Watson, G. S. (1964). Smooth regression analysis. Sankhya, Ser. A **26**, 359–372.