A model selection approach for local learning

Gianluca Bontempi, Mauro Birattari and
Hugues Bersini
Iridia – Université Libre de Bruxelles,
1050 Bruxelles, Belgium
E-mail: {gbonte, mbiro, bersini}@ulb.ac.be

Local learning techniques, for each query, extract a prediction interpolating locally the neighboring examples which are considered relevant according to a distance measure. As other learning approaches, the local learning procedure can be conveniently decomposed into a parametric identification and a structural identification. While parametric identification is reduced to a linear regression, structural identification requires that the designer perform a certain number of choices. In this paper we focus on an automatic query-by-query selection of the *bandwidth*, a structural parameter which plays a major role in the final performance. We propose a local method where, for each query, different model candidates are first generated, then assessed and finally selected. We introduce in the context of local learning the recursive least squares algorithm as an efficient way to generate local models. Moreover, local cross-validation is used as an economic way to validate different alternatives. As far as model selection is concerned, the *winner-takes-all* strategy and a local *combination* of the most promising models are explored. The method proposed is tested on six different datasets and compared with state-of-the-art approaches.

1. Introduction

Supervised learning can be conveniently decomposed into a *parametric identification* and a *structural identification* procedure. Once the model structure is given, the parametric identification selects the parameters which minimize on the training set the discrepancy between the target values and the predictions. On the other hand, structural identification aims to select in the space of possible model structures, the one which minimizes the generalization error. The search for the best parameters and the search for the best structure are instances of an optimization problem which is typically addressed in three stages: generation of different solutions, assessment of each solution and selection among the solutions assessed.

This paper will focus on the particular case of local methods for supervised learning. These methods perform function approximation by interpolating locally the training samples considered relevant according to a distance measure [1,3]. The parametric identification procedure is, therefore, quite simple and can be done through consolidated statistical methods. On the other side, it is commonly known that the performance of the local approximator is quite sensitive to the structural identification choices performed by the designer. Structural identification involves, among other things, the selection of a family of local approximators, the selection of a metric to evaluate which examples are more relevant, and the selection of the *bandwidth* which indicates the size of the region in which the data are correctly modeled by members of the chosen family of approximators. Although the prediction depends on the whole set of these structural parameters, it is common belief in local learning literature that the final performance is more sensitive to the bandwidth and to the distance metric [3]. As far as the problem of bandwidth selection is concerned, different approaches exist in literature. The choice of the bandwidth may be performed either based on some *a priori* assumption or on the data themselves. A further subclassification of data-driven approaches is of interest here. On the one hand, a constant bandwidth may be used; in this case it is set by a global optimization that minimizes an error criterion over the available dataset. On the other hand, the bandwidth may be selected locally and tailored for each query point.

In the present work, we propose a method that belongs to the latter class of local data-driven approaches and that, given a global distance metric, selects the bandwidth on a query-by-query basis. The main reason to favor a query-by-query bandwidth selection is that it allows better adaptation to the local characteristics of the problem at hand. Moreover, this approach is able to handle directly the case in which the database is updated on-line [7,8]. On the other hand, a globally optimized bandwidth approach would, in principle, require the global optimization to be repeated each time the distribution of the examples changes.

AI Communications 13 (2000) 41–47
ISSN 0921-7126 / $8.00 © 2000, IOS Press. All rights reserved

The method we propose is a local and query-based instance of the general structural identification procedure.

The model generation is based on the recursive least squares algorithm. This is an appealing and efficient solution to the intrinsically incremental problem of identifying and validating a sequence of local linear models centered in the query point, and each including a growing number of neighbors. The problem of bandwidth selection is reduced to the selection of the number k of neighboring examples which are given a non-zero weight in the local modeling procedure. Each time a prediction is required for a specific query point, a set of local models is identified, each including a different number of neighbors.

Once the candidate models have been generated, the generalization ability of each of them is assessed through a local cross-validation procedure. Here we use the PRESS statistic [2] which is a simple, well-founded and economical way to perform *leave-one-out* cross validation [9] and to assess the generalization performance of local linear models. It is worth noticing here that leave-one-out does not involve any significant computational overload, since the PRESS statistic uses partial results returned by the recursive least squares algorithm.

Finally, the paper explores a *competitive* and a *co-operative* approach to model selection on the basis of some statistics of their cross-validation errors. In this local learning setting, we propose a comparison between a *winner-takes-all* strategy and a strategy based on the *combination of estimators* [17].

An experimental analysis of the recursive algorithm for local identification and validation is presented. The algorithm proposed is experimentally compared with other local bandwidth selection approaches, and with state-of-the-art methods as feedforward neural networks and Cubist, the rule-based tool developed by Ross Quinlan for generating piecewise-linear models.

It is worth saying that this paper will not focus on algorithmic computational issues. On this subject, the reader is invited to refer to the literature dealing with efficient implementations of local algorithms [11,12].

2. Local weighted regression

Given two variables $\mathbf{x} \in \Re^m$ and $y \in \Re$, let us consider the mapping $f\colon \Re^m \rightarrow \Re$, known only through a set of n examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ obtained as follows:

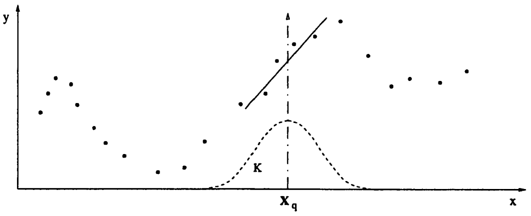

Fig. 1. Local weighted regression in the neighborhood of a quer point $\mathbf{x}_q$. A local linear model approximates the single-input sir gle-output mapping in a neighborhood of $\mathbf{x}_q$ characterized by a kei nel $K(\cdot)$.

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \qquad (1$$

where $\forall i$, ε_i is a random variable such that $E[\varepsilon_i] =$ (and $E[\varepsilon_i \varepsilon_j] = 0$, $\forall j \neq i$, and such that $E[\varepsilon_i^m] =$ $\mu_m(\mathbf{x}_i)$, $\forall m \geqslant 2$, where $\mu_m(\cdot)$ is the unknown mt moment of the distribution of ε_i and is defined as function of $\mathbf{x}_i$. In particular for $m = 2$, the last of th above mentioned properties implies that no assumptio of global *homoscedasticity* (equal variance of the nois over the input space) is made.

The problem of local regression can be stated a the problem of estimating the value that the regressio function $f(\mathbf{x}) = E[y|\mathbf{x}]$ assumes for a specific quer point $\mathbf{x}$, using information pertaining only to a neighborhood of $\mathbf{x}$.

Given a query point $\mathbf{x}_q$, and under the hypothesis o a local homoscedasticity of ε_i, the parameter β of a local linear approximation of $f(\cdot)$ in a neighborhood o $\mathbf{x}_q$ can be obtained solving the local polynomial regres sion (Fig. 1):

$$\sum_{i=1}^{n} \left\{ (y_i - \mathbf{x}_i'\beta)^2 K\left(\frac{d(\mathbf{x}_i, \mathbf{x}_q)}{h}\right) \right\}, \qquad (2$$

where, given a metric on the space $\Re^m$, $d(\mathbf{x}_i, \mathbf{x}_q)$ is th distance from the query point to the ith example, $K(\cdot$ is a weight function, h is the bandwidth, and where a constant value 1 has been appended to each vector $\mathbf{x}$ in order to consider a constant term in the regression.

In matrix notation, the solution of the above statec weighted least squares problem is given by:

$$\begin{aligned} \hat{\beta} &= (\mathbf{X}'\mathbf{W}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}'\mathbf{W}\mathbf{y} \\ &= (\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{v} = \mathbf{P}\mathbf{Z}'\mathbf{v}, \end{aligned} \qquad (3$$

where $\mathbf{X}$ is a matrix whose ith row is $\mathbf{x}_i'$, $\mathbf{y}$ is a vecto whose ith element is y_i, $\mathbf{W}$ is a diagonal matrix whos ith diagonal element is $w_{ii} = \sqrt{K(d(\mathbf{x}_i, \mathbf{x}_q)/h)}$, $\mathbf{Z} =$

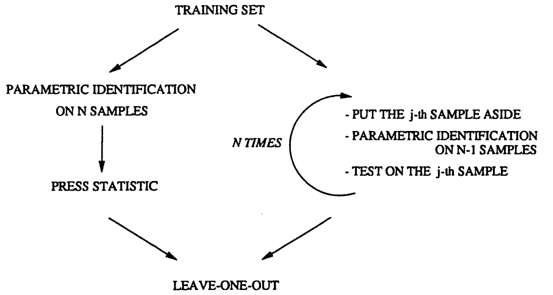
TRAINING SET
PARAMETRIC IDENTIFICATION
ON N SAMPLES
PRESS STATISTIC
N TIMES
- PUT THE j-th SAMPLE ASIDE
- PARAMETRIC IDENTIFICATION
ON N-1 SAMPLES
- TEST ON THE j-th SAMPLE
LEAVE-ONE-OUT

Fig. 2. Leave-one-out for linear models. The leave-one-out error can be computed in two equivalent ways: the slowest way (on the right) which repeats N times the training and the test procedure; the fastest way (on the left) which performs only once the parametric identification and the computation of the PRESS statistic.

$\mathbf{WX}$, $\mathbf{v} = \mathbf{Wy}$, and the matrix $\mathbf{X'W'WX} = \mathbf{Z'Z}$ is assumed to be non-singular so that its inverse $\mathbf{P} = (\mathbf{Z'Z})^{-1}$ is defined.

Once obtained the local linear polynomial approximation, a prediction of $y_q = f(\mathbf{x}_q)$, is finally given by:

$$\widehat{y}_q = \mathbf{x}'_q \widehat{\beta}. \tag{4}$$

A reliable assessment of the mean-squared-error

$$\mathrm{MSE} = E\left[(y_q - \hat{y}_q)^2\right], \tag{5}$$

is commonly provided by *cross-validation* [16]. A well-known disadvantage of a cross-validated approach is the large computational effort which is required to perform the series of training steps. However, in the case of linear models there exists a powerful statistical procedure to compute the leave-one-out cross-validation measure at a reduced computational cost (Fig. 2). It is the PRESS (Prediction Sum of Squares) statistic [2], a simple formula which returns the leave-one-out (l-o-o) error

$$e_j^{\mathrm{cv}} = y_j - \mathbf{x}'_j \widehat{\beta}_{-j}, \tag{6}$$

without explicitly identifying the parameters $\widehat{\beta}_{-j}$ from the examples available with the jth removed.

The formulation of the PRESS statistic for the case at hand is the following:

$$\begin{aligned} e_j^{\mathrm{cv}} = y_j - \mathbf{x}'_j \widehat{\beta}_{-j} &= \frac{y_j - \mathbf{x}'_j \mathbf{PZ'v}}{1 - \mathbf{z}'_j \mathbf{P} \mathbf{z}_j} \\ &= \frac{y_j - \mathbf{x}'_j \widehat{\beta}}{1 - h_{jj}}, \end{aligned} \tag{7}$$

where $\mathbf{z}'_j$ is the jth row of $\mathbf{Z}$ and therefore $\mathbf{z}_j = w_{jj}\mathbf{x}_j$, and where h_{jj} is the jth diagonal element of the *Hat matrix* $\mathbf{H} = \mathbf{ZPZ'} = \mathbf{Z}(\mathbf{Z'Z})^{-1}\mathbf{Z}'$.

3. Recursive model generation

In what follows, for the sake of simplicity, we will focus on first-degree linear approximators. An extension to generic polynomial approximators of any degree is straightforward. We will assume also that a metric on the space $\Re^m$ is given. All the attention will be thus centered on the problem of bandwidth selection.

If as a weight function $K(\cdot)$ the indicator function

$$K\left(\frac{d(\mathbf{x}_i, \mathbf{x}_q)}{h}\right) = \begin{cases} 1 & \text{if } d(\mathbf{x}_i, \mathbf{x}_q) \leqslant h, \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

is adopted, the optimization of the parameter h can be conveniently reduced to the optimization of the number k of neighbors to which a unitary weight is assigned in the local regression evaluation. In other words, we reduce the problem of bandwidth selection to a search in the space of $h(k) = d(\mathbf{x}(k), \mathbf{x}_q)$, where $\mathbf{x}(k)$ is the kth nearest neighbor of the query point.

The main advantage deriving from the adoption of the weight function defined in Eq. (8), is that, simply by updating the parameter $\widehat{\beta}(k)$ of the model identified using the k nearest neighbors, it is straightforward and inexpensive to obtain $\widehat{\beta}(k+1)$. In fact, performing a step of the standard recursive least squares algorithm [4], we have:

$$\begin{cases} \mathbf{P}(k+1) = \mathbf{P}(k) \\ \qquad - \dfrac{\mathbf{P}(k)\mathbf{x}(k+1)\mathbf{x}'(k+1)\mathbf{P}(k)}{1 + \mathbf{x}'(k+1)\mathbf{P}(k)\mathbf{x}(k+1)}, \\ \gamma(k+1) = \mathbf{P}(k+1)\mathbf{x}(k+1), \\ e(k+1) = y(k+1) - \mathbf{x}'(k+1)\widehat{\beta}(k), \\ \widehat{\beta}(k+1) = \widehat{\beta}(k) + \gamma(k+1)e(k+1), \end{cases} \tag{9}$$

where $\mathbf{P}(k) = (\mathbf{Z'Z})^{-1}$ when $h = h(k)$, and where $\mathbf{x}(k+1)$ is the $(k+1)$th nearest neighbor of the query point.

Once an initialization $\widehat{\beta}(0) = \widetilde{\beta}$ and $\mathbf{P}(0) = \widetilde{\mathbf{P}}$ is given, Eq. (9) and Eq. (10) recursively evaluate for different values of k a local approximation of the regression function $f(\cdot)$, a prediction of the value of the regression function in the query point, and the vector of leave-one-out errors from which it is possible to extract an estimate of the variance of the prediction error.

Notice that $\widetilde{\beta}$ is an *a priori* estimate of the parameter and $\widetilde{\mathbf{P}}$ is the covariance matrix that reflects the reliability of $\widetilde{\beta}$ [4]. For non-reliable initialization, the following is usually adopted: $\widetilde{\mathbf{P}} = \lambda\mathbf{I}$ with λ large, and where $\mathbf{I}$ is the identity matrix.

4. Local model validation

In the previous section we introduced recursive least-squares as an effective method for generating local model candidates. These models have now to be validated in order to proceed to the final model selection. Once the leave-one-out is adopted as validation criterion, it follows that the model generation procedure returns as a by-product all the elements necessary for the PRESS computation.

Matrix $\mathbf{P}(k+1)$ is returned by Eq. (9) and thus the leave-one-out cross-validation errors can be directly calculated without the need of any further model identification:

$$e_j^{\mathrm{cv}}(k+1) = \frac{y_j - \mathbf{x}_j'\widehat{\beta}(k+1)}{1 - \mathbf{x}_j'\mathbf{P}(k+1)\mathbf{x}_j}, \qquad \forall j\colon d(\mathbf{x}_j, \mathbf{x}_q) \leqslant h(k+1). \tag{10}$$

It will be useful in the following to define for each value of k the $[k \times 1]$ vector $\mathbf{e}^{\mathrm{cv}}(k)$ that contains all the leave-one-out errors associated to the model $\widehat{\beta}(k)$.

5. Local model selection and combination

Once a set of candidate models have been generated through Eq. (9) and validated through Eq. (10), we proceed to model selection. The recursive algorithm described by Eq. (9) and Eq. (10) returns for a given query point $\mathbf{x}_q$, a set of predictions $\widehat{y}_q(k) = \mathbf{x}_q'\widehat{\beta}(k)$, together with a set of associated leave-one-out error vectors $\mathbf{e}^{\mathrm{cv}}(k)$.

From the information available, a final prediction $\hat{y}_q$ of the value of the regression function can be obtained in different ways. Two main paradigms deserve to be considered: the first is based on the selection of the *best* approximator according to a given criterion, while the second returns a prediction as a combination of more local models.

If the selection paradigm, frequently called *winner-takes-all*, is adopted, the most natural way to extract a final prediction $\hat{y}_q$, consists in comparing the prediction obtained for each value of k on the basis of the classical *mean square error* criterion:

$$\hat{y}_q = \mathbf{x}_q'\widehat{\beta}(\hat{k}), \tag{11}$$

with

$$\hat{k} = \arg\min_k \mathrm{MSE}(k) = \arg\min_k \frac{\sum_{i=1}^k \omega_i (\mathbf{e}_i^{\mathrm{cv}}(k))^2}{\sum_{i=1}^k \omega_i}, \tag{12}$$

where ω_i are weights than can be conveniently used to discount each error according to the distance from the query point to the point to which the error corresponds [3].

As an alternative to the *winner-takes-all* paradigm, we explored also the effectiveness of local combinations of estimates [15,17]. Adopting also in this case the *mean square error* criterion, the final prediction of the value y_q is obtained as a weighted average of the best b models, where b is a parameter of the algorithm. Suppose the predictions $\hat{y}_q(k)$ and the error vectors $\mathbf{e}^{\mathrm{cv}}(k)$ have been ordered creating a sequence of integers $\{k_i\}$ so that $\mathrm{MSE}(k_i) \leqslant \mathrm{MSE}(k_j)$, $\forall i < j$. The prediction of $\hat{y}_q$ is given by

$$\hat{y}_q = \frac{\sum_{i=1}^b \zeta_i \hat{y}_q(k_i)}{\sum_{i=1}^b \zeta_i}, \tag{13}$$

where the weights are the inverse of the mean square errors: $\zeta_i = 1/\mathrm{MSE}(k_i)$. This is an example of the *generalized ensemble method* [13].

6. Experiments and results

The experimental evaluation of the incremental local identification and validation algorithm was performed on six datasets. The first five, described by Quinlan [14], were obtained from the UCI Repository of machine learning databases [5], while the last one was provided by Leo Breiman. A summary of the characteristics of each dataset is presented in Table 1.

Table 1
A summary of the characteristics of the datasets considered

Dataset	Housing	Cpu	Prices	Mpg	Servo	Ozone
Number of examples	506	209	159	392	167	330
Number of regressors	13	6	16	7	8	8

The methods compared adopt the recursive identification and validation algorithm, combined with different strategies for model selection or combination. We considered also two approaches in which k is selected globally:

lb1: Local bandwidth selection for linear local models. The number of neighbors is selected on a query-by-query basis and the prediction returned is the one of the best model according to the mean square error criterion.

lb0: Local bandwidth selection for constant local models. The algorithm for constant models is derived directly from the recursive method described in Eq. (9) and Eq. (10). The best model is selected according to the mean square error criterion.

lbC: Local combination of estimators. This is an example of the method described in Eq. (13). On the datasets proposed, for each query the best 2 linear local models and the best 2 constant models are combined.

gb1: Global bandwidth selection for linear local models. The value of k is obtained minimizing the prediction error in 20-fold cross-validation on the dataset available. This value is then used for all the query points.

gb0: Global bandwidth selection for constant local models. As in **gb1**, the value of k is optimized globally and kept constant for all the queries.

The distance between a generic query $\mathbf{x}_q$ and an input $\mathbf{x}_i$ in the training set is computed by using a weighted Euclidean distance

$$d(\mathbf{x}_i, \mathbf{x}_q) = \sqrt{(\mathbf{x}_i - \mathbf{x}_q)'\mathbf{M}'\mathbf{M}(\mathbf{x}_i - \mathbf{x}_q)}, \tag{14}$$

where **M** is a global parameter and given by the relative linear influence (*relevance*) of the regressors [10]. This means that the metric M in (14) is a diagonal matrix with

$$M_{jj} = \sqrt{\frac{\widehat{\beta}_j^2}{\sum_{j=1}^{n} \widehat{\beta}_j^2}}, \tag{15}$$

where $\widehat{\beta}_j$ is the jth term of the least-squares vector estimated on the whole training set. We are confident that the adoption of a local metric could improve the performance of our local learning method.

The local learning results are compared with those we obtained, in the same experimental settings, both using feedforward neural networks and Cubist, the rule-based tool developed by Quinlan for generating piecewise-linear models. While Cubist is an integrated tool which performs automatically model selection and returns the best expected prediction, a fair comparison with feedforward neural networks should require a state-of-the-art neural selection procedure. In order to avoid possible criticism on this subject, we decided to perform no neural structural identification but to compute the predictions for several different structures and to return the best *a posteriori* result on the test set. This quite *optimistic* result is by definition better than any other result obtainable by any structural identification method for neural networks. We focused in particular on two-layer architectures with a first sigmoid layer and a second linear layer, trained with the Levenberg-Marquardt algorithm. We chose as structural parameter the number of neurons in the first layer, and we made it vary over a range between 2 and 12. In the table we report only the result obtained by the best neural structure which is not necessarily the same among different datasets.

Each approach was tested on each dataset using the same 10-fold cross-validation strategy. Each dataset was divided randomly into 10 groups of nearly equal size. In turn, each of these groups was used as a testing set while the remaining ones together were providing the examples. Thus all the methods performed a prediction on the same unseen cases, using for each of them the same set of examples. In Table 2 we present the results obtained by all the methods, and averaged on the 10 cross-validation groups. Since the methods were compared on the same examples in exactly the same conditions, the sensitive one-tailed paired test of significance can be used. In what follows, by 'significantly better' we mean better at least at a 5% significance level.

The first consideration about the results concerns the local combination of estimators. According to Table 2, the method **lbC** performs in average always better than the *winner-takes-all* linear and constant. On

Table 2

Mean absolute error on unseen cases

Method	Housing	Cpu	Prices	Mpg	Servo	Ozone
lb1	2.21	28.38	1509	1.94	0.48	3.52
lb0	2.60	31.54	1627	1.97	0.32	3.33
lbC	2.12	26.79	1488	1.83	0.29	3.31
gb1	2.30	28.69	1492	1.92	0.52	3.46
gb0	2.59	32.19	1639	1.99	0.34	3.19
Cubist	2.17	28.37	1331	1.90	0.36	3.15
Nnet	2.33	31.18	2092	2.05	0.38	3.32

Table 3

Relative error (%) on unseen cases

Method	Housing	Cpu	Prices	Mpg	Servo	Ozone
lb1	12.63	9.20	15.87	12.65	28.66	35.25
lb0	18.06	20.37	22.19	12.64	22.04	31.11
lbC	12.35	9.29	17.62	11.82	19.72	30.28
gb1	13.47	9.93	15.95	12.83	30.46	32.58
gb0	17.99	21.43	22.29	13.48	24.30	28.21
Cubist	16.02	12.71	11.67	12.57	18.53	26.59
Nnet	14.06	14.40	32.17	12.65	22.47	30.06

two dataset **lbC** is significantly better than both **lb1** and **lb0**; and on three dataset it is significantly better than one of the two, and better in average than the other.

The second consideration is about the comparison between our query-by-query bandwidth selection and a global optimization of the number of neighbors: in average **lb1** and **lb0** performs better than their counterparts **gb1** and **gb0**. On two datasets **lb1** is significantly better than **gb1**, while is about the same on the other four. On one dataset **lb0** is significantly better than **gb0**.

As far as the comparison with Cubist is concerned, the recursive local identification and validation proposed obtains results comparable with those obtained by the state-of-the-art method implemented in Cubist. On the six datasets, **lbC** performs one time significantly better than Cubist, and one time significantly worse.

As far as the comparison with feedforward neural networks is concerned, the proposed local method obtains results significantly better. On the six datasets, **lbC** performs five times significantly better than the best neural network.

The second index of performance we investigated is the *relative error*, defined as the mean square error on unseen cases, normalized by the variance of the test set. The relative errors are presented in Table 3 and show a similar picture to Table 2, although the mean square errors considered here penalize larger absolute errors.

We suggest the interested reader to refer to [6] for a larger experimental setting (more datasets and more referential approaches).

7. Conclusion and future work

The paper presented a bandwidth selection approach for local learning method. Despite the trivial metric adopted the experimental results confirm that the approach is able to compete with a state-of-the-art approaches and can be effectively used in a local context for multivariate regression problems.

Future work will focus on the problem of local metric selection. Moreover, we will explore more sophisticated ways to combine local estimators and we will extend this work to polynomial approximators of higher degree.

Acknowledgements

The work of Gianluca Bontempi was supported by the European Union TMR Grant FMBICT960692. The work of Mauro Birattari was supported by the FIRST program of the Région Wallonne, Belgium. The authors thank Ross Quinlan and gratefully acknowledge using his software Cubist. For more details on Cubist see http://www.rulequest.com. We also thank Leo Breiman for the dataset *ozone* and the UCI Repository for the other datasets used in this paper.

References

[1] D.W. Aha, Editorial, *Artificial Intelligence Review* **11**(1–5) (1997), 1–6.

[2] D.M. Allen, The relationship between variable and data augmentation and a method of prediction, *Technometrics* **16** (1974), 125–127.

[3] C.G. Atkeson, A.W. Moore and S. Schaal, Locally weighted learning, *Artificial Intelligence Review* **11**(1–5) (1997), 11–73.

[4] G.J. Bierman, *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York, NY, 1977.

[5] C. Blake, E. Keogh and C.J. Merz, *UCI Repository of Machine Learning Databases*, 1998.

[6] G. Bontempi, Local learning techniques for modeling, prediction and control, PhD thesis, IRIDIA – Université Libre de Bruxelles, 1999.

[7] G. Bontempi, M. Birattari and H. Bersini, Lazy learning for modeling and control design, *International Journal of Control* **72**(7/8) (1999), 643–658.

[8] G. Bontempi, H. Bersini and M. Birattari, The local paradigm for modeling and control: From neuro-fuzzy to lazy learning. *Fuzzy Sets and Systems* (1999) (in press).

[9] B. Efron and R.J. Tibshirani, *An Introduction to the Bootstrap*, Chapman and Hall, New York, NY, 1993.

[10] J.H. Friedman, Flexible metric nearest neighbor classification, Tech. rept. Stanford University, 1994.

[11] A.W. Moore, J. Schneider and K. Deng, Efficient locally weighted polynomial regression predictions, in: *Proceedings of the 1997 International Machine Learning Conference*, Morgan Kaufmann Publishers, 1997.

[12] S.A. Nene and S.K. Nayar, A simple algorithm for nearest neighbor search in high dimensions, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(9) (1997), 989–1003.

[13] M.P. Perrone and L.N. Cooper, When networks disagree: Ensemble methods for hybrid neural networks, in: *Artificial Neural Networks for Speech and Vision*, R.J. Mammone, ed., Chapman and Hall, 1993, pp. 126–142.

[14] J.R. Quinlan, Combining instance-based and model-based learning, in: *Machine Learning. Proceedings of the Tenth International Conference*, Morgan Kaufmann, 1993, pp. 236–243.

[15] A.J.C. Sharkey, *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems (Perspectives in Neural Computing)*, Springer Verlag, 1999.

[16] M. Stone, Cross-validatory choice and assessment of statistical predictions, *Journal of the Royal Statistical Society B* **36**(1) (1974), 111–147.

[17] D. Wolpert, Stacked generalization, *Neural Networks* **5** (1992), 241–259.