On the use of cross-validation for local modeling in regression and time series prediction

Gianluca Bontempi

gbonte@ulb.ac.be

Machine Learning Group

Departement d'Informatique, ULB

Boulevard de Triomphe - CP 212

http://www.ulb.ac.be/di/mlg

Outline

- The Machine Learning Group
- A local learning algorithm: the Lazy Learning.
- Lazy Learning for multivariate regression modeling.
- Lazy Learning for multi-step-ahead time series prediction.
- Lazy Learning for feature selection.
- Applications.
- Future work.

Machine Learning: a definition

The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience. [35]

The Machine Learning Group (MLG)

- 7 researchers (1 prof, 6 PhD students), 4 graduate students).
- Research topics: Bioinformatics, Classification, Computational statistics, Data mining, Regression, Time series prediction, Sensor networks.
- Computing facilities: cluster of 16 processors, LEGO Robotics Lab.
- Website: `www.ulb.ac.be/di/mlg`.
- Scientific collaborations in ULB: IRIDIA (Sciences Appliquées), Physiologie Moléculaire de la Cellule (IBMM), Conformation des Macromolécules Biologiques et Bioinformatique (IBMM), CENOLI (Sciences), Microarray Unit (Hopital Jules Bordet), Service d’Anesthesie (ERASME).
- Scientific collaborations outside ULB: UCL Machine Learning Group (B), Politecnico di Milano (I), Universitá del Sannio (I), George Mason University (US).
- The MLG is part to the "Groupe de Contact FNRS" on Machine Learning.

MLG: running projects

1. "Integrating experimental and theoretical approaches to decipher the molecular networks of nitrogen utilisation in yeast": ARC (Action de Recherche Concertée) funded by the Communauté Française de Belgique (2004-2009). Partners: IBMM (Gosselies and La Plaine), CENOLI.

2. "COMP2SYS" (COMPutational intelligence methods for COMPlex SYStems) MARIE CURIE Early Stage Research Training funded by the European Union (2004-2008). Main contractor: IRIDIA (ULB).

3. "Predictive data mining techniques in anaesthesia": FIRST Europe Objectif 1 funded by the Région wallonne and the Fonds Social Européen (2004-2009). Partners: Service d'anesthesie (ERASME).

4. "AIDAR - Adressage et Indexation de Documents Multimédias Assistés par des techniques de Reconnaissance Vocale": funded by Région Bruxelles-Capitale (2004-2006). Partners: Voice Insight, RTBF, Titan.

Machine learning and applied statistics

Reductionist attitude: *ML is a modern buzzword which equates to statistics plus marketing*

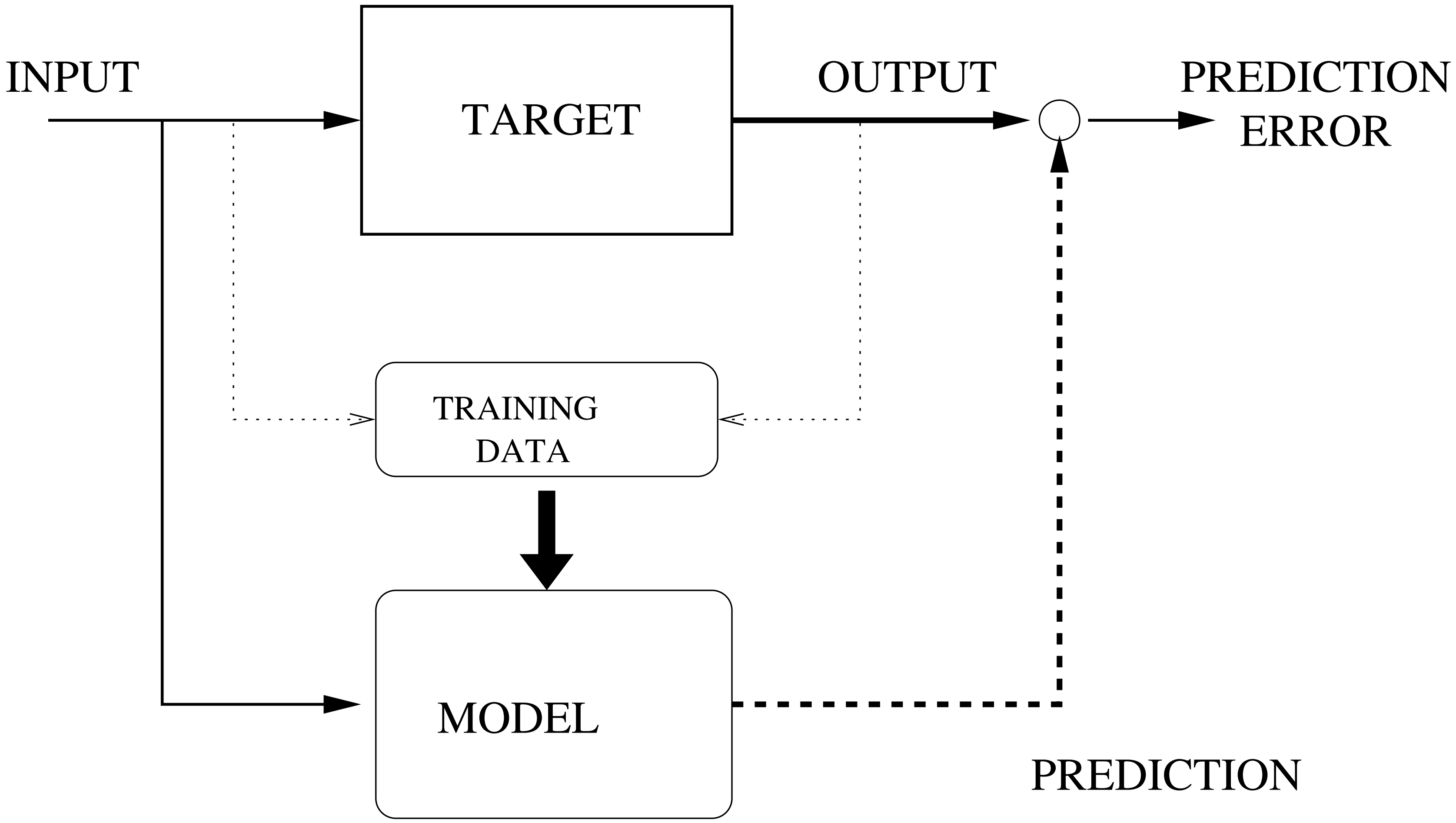
Positive attitude: ML paved the way to the treatment of real problems related to data analysis, sometimes overlooked by statisticians (nonlinearity, classification, pattern recognition, missing variables, adaptivity, optimization, massive datasets, data management, causality, representation of knowledge, parallelisation)

Interdisciplinary attitude: ML *should* have its roots on statistics and complements it by focusing on: algorithmic issues, computational efficiency, data engineering.

Motivations

- There exists a wide amount of theoretical and practical results for linear methods in statistics, forecasting and control.
- However, in real settings we encounter often nonlinear problems.
- Nonlinear methods are generally more difficult to analyze than linear ones, rarely produce closed-form or analytically tractable expressions, and are not easy to manipulate and implement.
- Local learning techniques are a powerful way of re-using linear techniques in a nonlinear setting.

Prediction models from data



Regression setting

- Multidimensional *input* $x \in \Re^n$ and scalar *output* $y \in \Re$

  $$y = f(x) + \varepsilon$$

  where f is the unknown regression function and ε is the random error term.

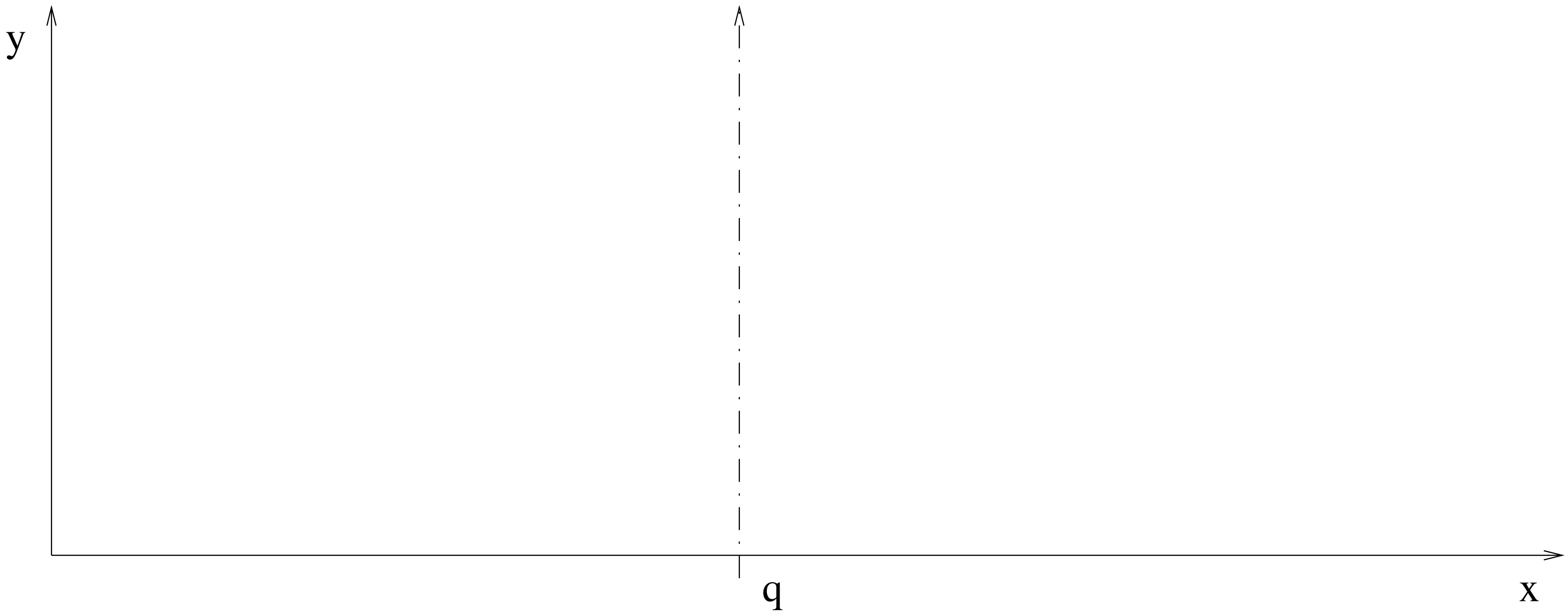
- A finite number of noisy input/output observations (*training set* D_N).

- A *test set* of input values for which an accurate *generalization* or *prediction* of the output is required.

- A *learning machine* which returns a input/output *model* on the basis of training set.

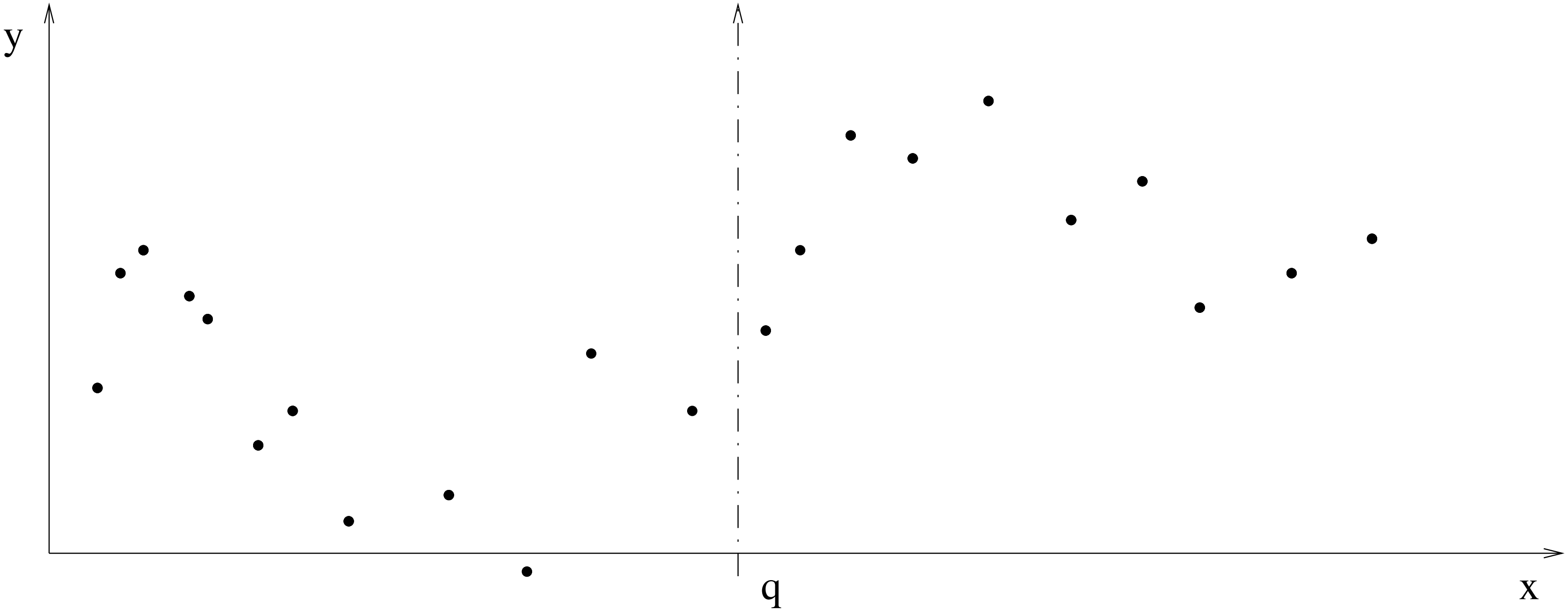
Assumption: No a priori knowledge on the process underlying the data.

The global modeling approach



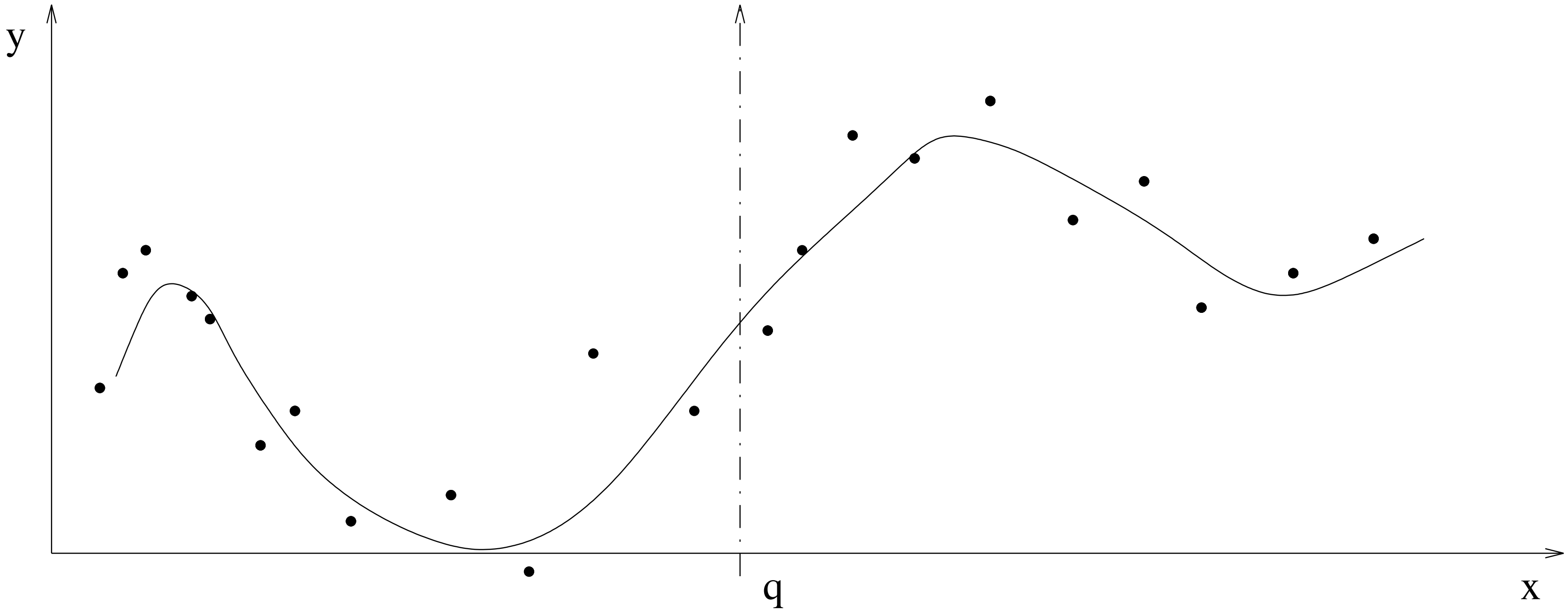
Input-output regression problem.

The global modeling approach



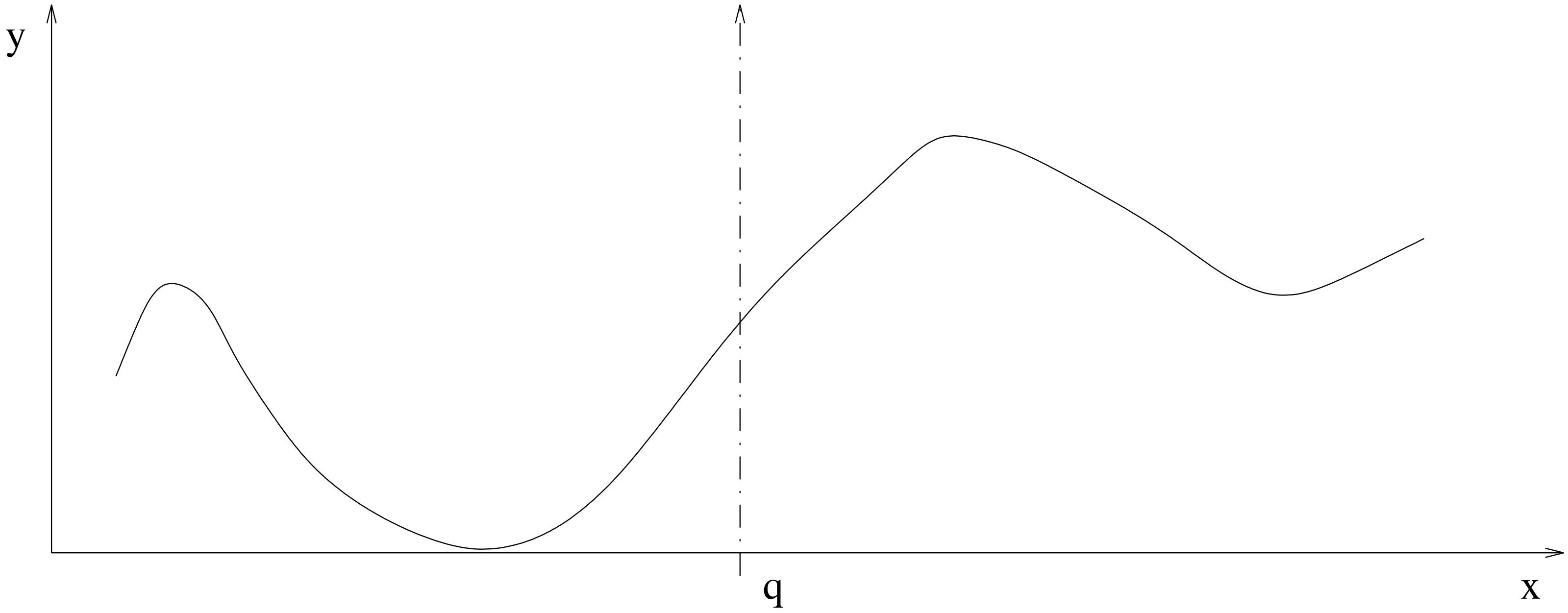
Training data set.

The global modeling approach



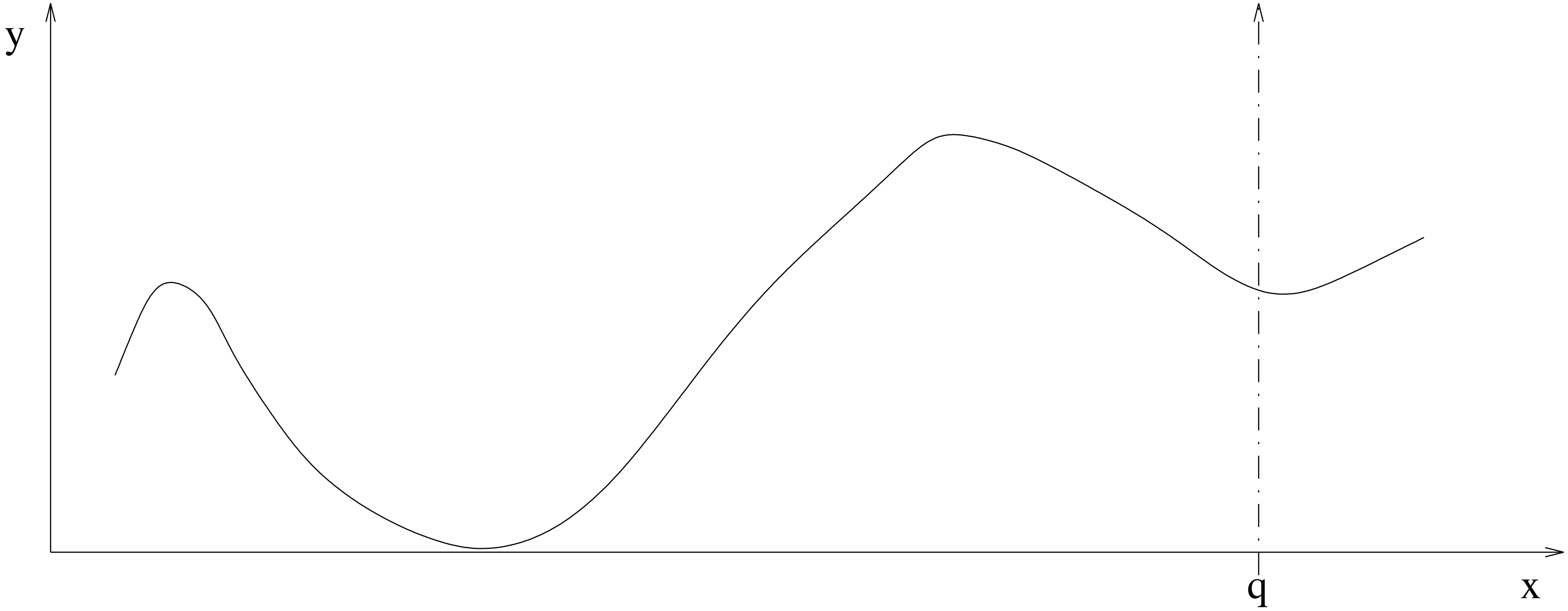
Global model fitting.

The global modeling approach



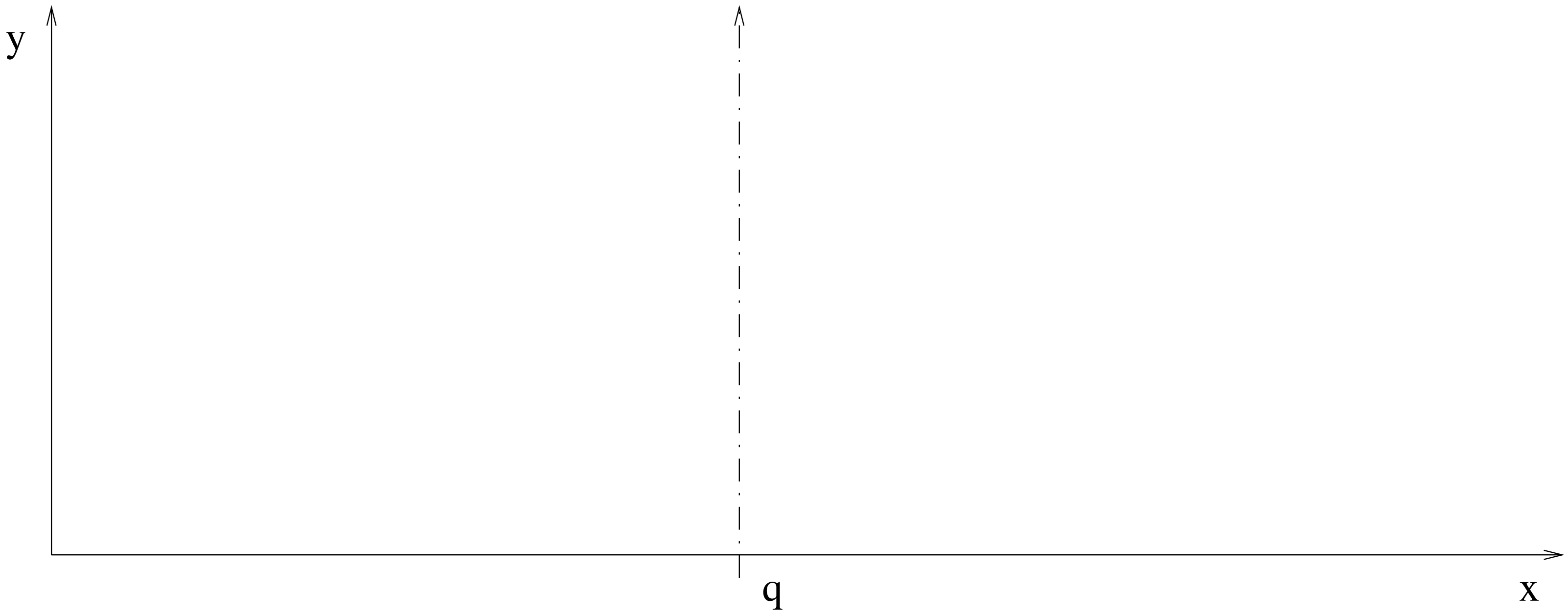
Prediction by using the fitted global model.

The global modeling approach



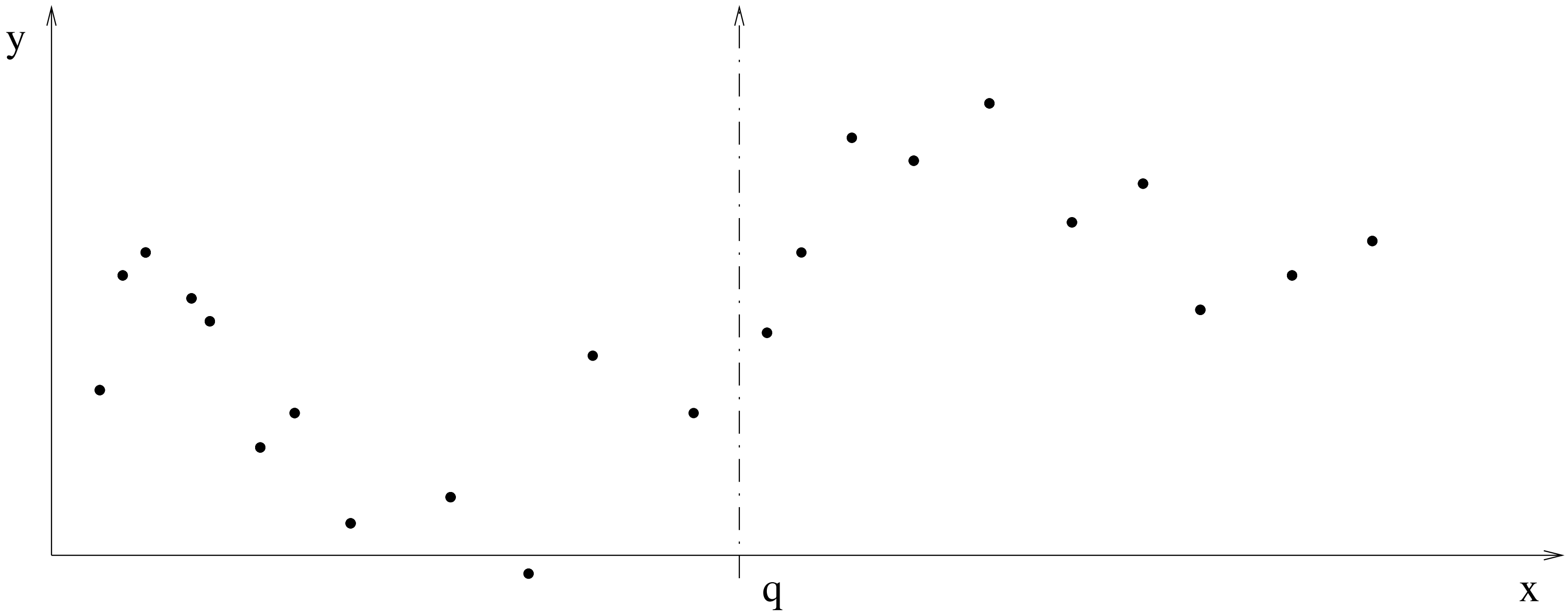
Another prediction by using the fitted global model.

The local modeling approach



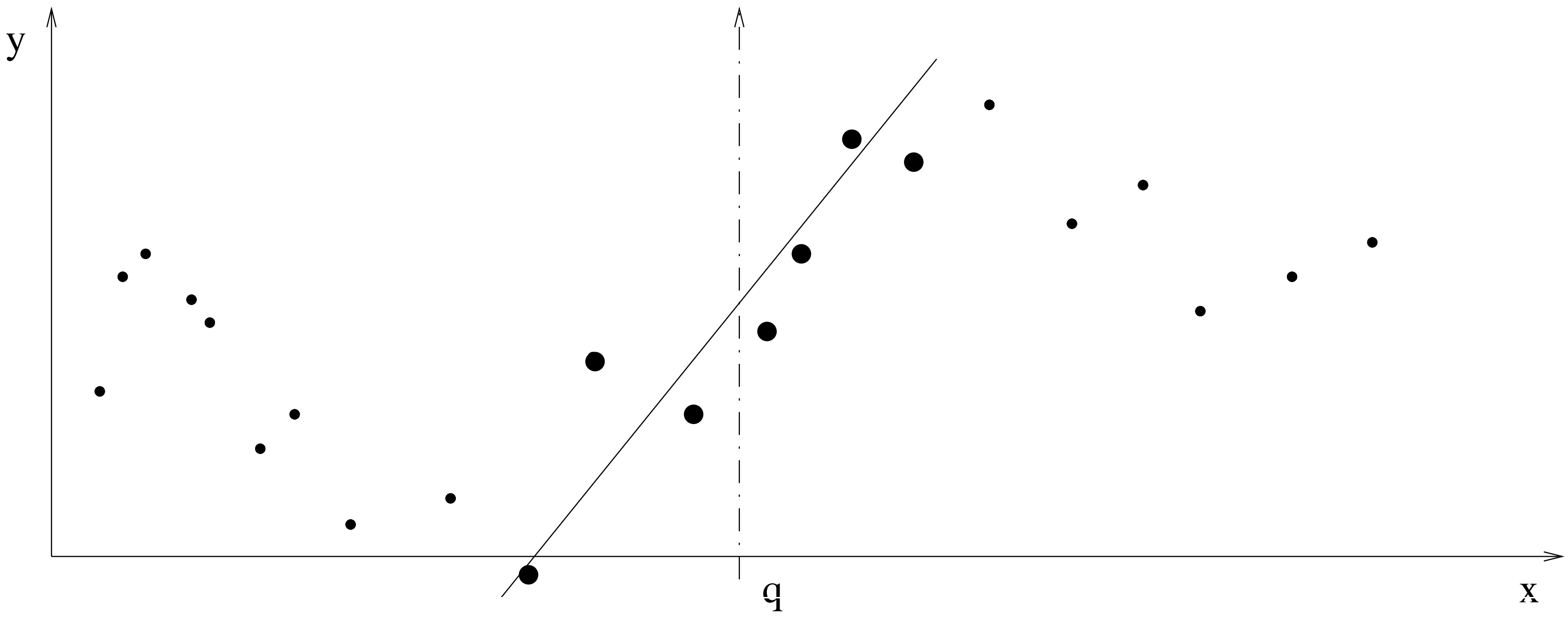
Input-output regression problem.

The local modeling approach



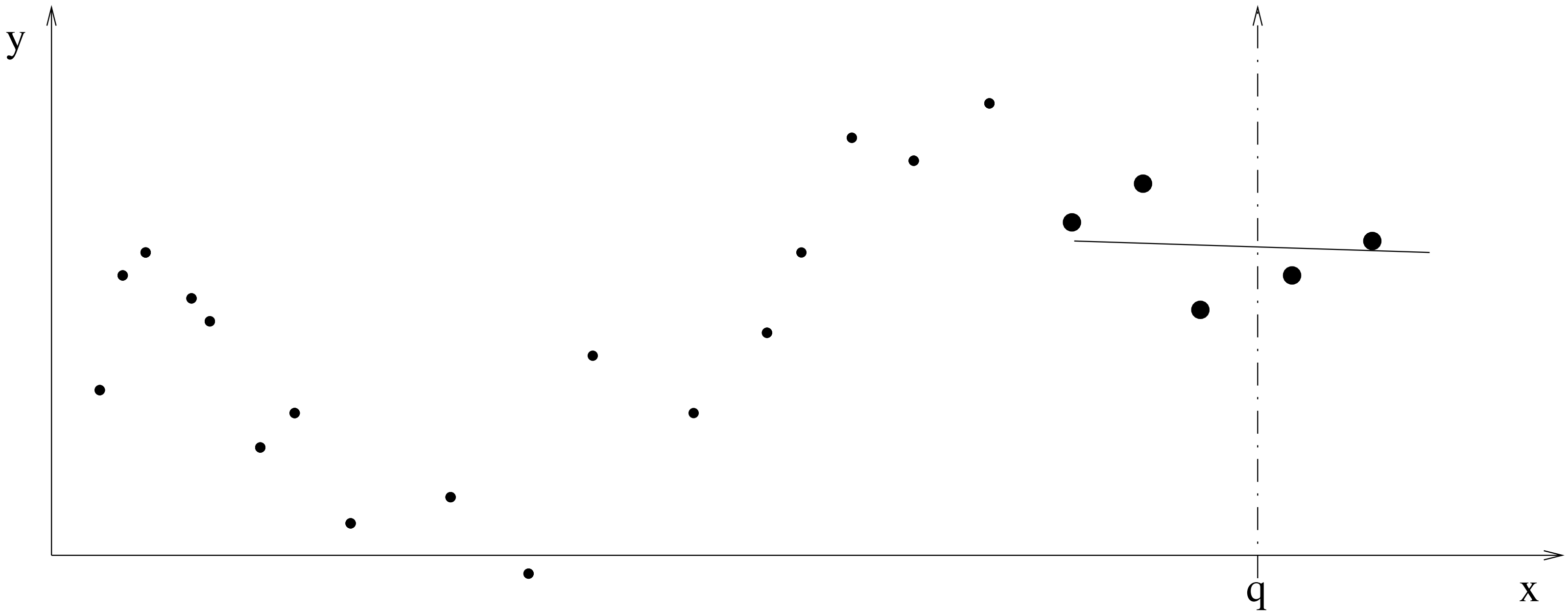
Training data set.

The local modeling approach



Local fitting and prediction.

The local modeling approach



Another local fitting and prediction.

Global vs. local modeling

- The traditional approach to supervised learning is *global* modeling which describes the relationship between the input and the output with an analytical function over the whole input domain.
- Even for huge datasets, a parametric model can be stored in a small memory. Also, the evaluation of the parametric model requires a short program that can be executed in a reduced amount of time.
- Modeling complex input/output relations often requires the adoption of global nonlinear models, whose learning procedures are typically slow and analytically intractable. In particular, *validation* methods, which address the problem of assessing a global model on the basis of a finite amount of noisy samples, are computationally prohibitive.
- For these reasons, in recent years, interest has grown in pursuing alternatives (*divide-and-conquer*) to global modeling techniques.

Global vs. local modeling

- The divide-and-conquer strategy consists in attacking a complex problem by dividing it into simpler problems whose solutions can be combined to yield a solution to the original problem.
- Instances of the divide-and-conquer approach are *modular* techniques (e.g. local model networks [36], regression trees [19], splines [45]) and *local modeling* (aka smoothing) techniques.
- The principle underlying local modeling is that a smooth function can be well approximated by a low degree polynomial in the neighborhood of any query point.
- Local modeling techniques do not return a global fit of the available dataset but perform the prediction of the output for specific test input values, also called *queries*.
- The talk presents our contribution to local modeling techniques and their application to a number of experimental problems.

Lazy vs. eager modeling

- Eager techniques perform a wide amount of computation for tuning the model before observing the new query.
- An eager technique must then commit to a specific hypothesis that covers all the future queries.
- Lazy techniques [1] wait for the query to be defined before starting the learning procedure.
- For that purpose, the database of observed input/output data is always kept in memory and the output prediction is obtained by interpolating the samples in the neighborhood of the query point.
- Lazy methods will generally require less computation during training but more computation when they must predict the target value for a new query.

Examples

- The classical linear regression is an example of global, eager, and linear approach.
- Neural networks (NN) are instances of the global, eager, and nonlinear approach: NN are global in the sense that a single representation covers the whole input space. They are eager in the sense that the examples are used for tuning the network and then they are discarded without waiting for any query. Finally, NN are nonlinear in the sense that the relation between the weights and the output is nonlinear.
- The technique we are going to discuss here is a lazy and local approach.
- Remark: we can imagine a local technique (e.g. a K-nearest neighbor) where the most important parameter (i.e. the number of neighbors) is defined in an eager fashion.

Some history

- Local regression estimation was independently introduced in several different fields in the late nineteenth [42] and early twentieth century [28].
- In the statistical literature, the method was independently introduced from different viewpoints in the late 1970's [20, 31, 43].
- Reference books are Fan and Gijbels [26] and Loader [32].
- In the machine learning literature, work on local techniques for classification dates back to 1967 [24]. A more recent reference is the special issue on Lazy Learning [1].

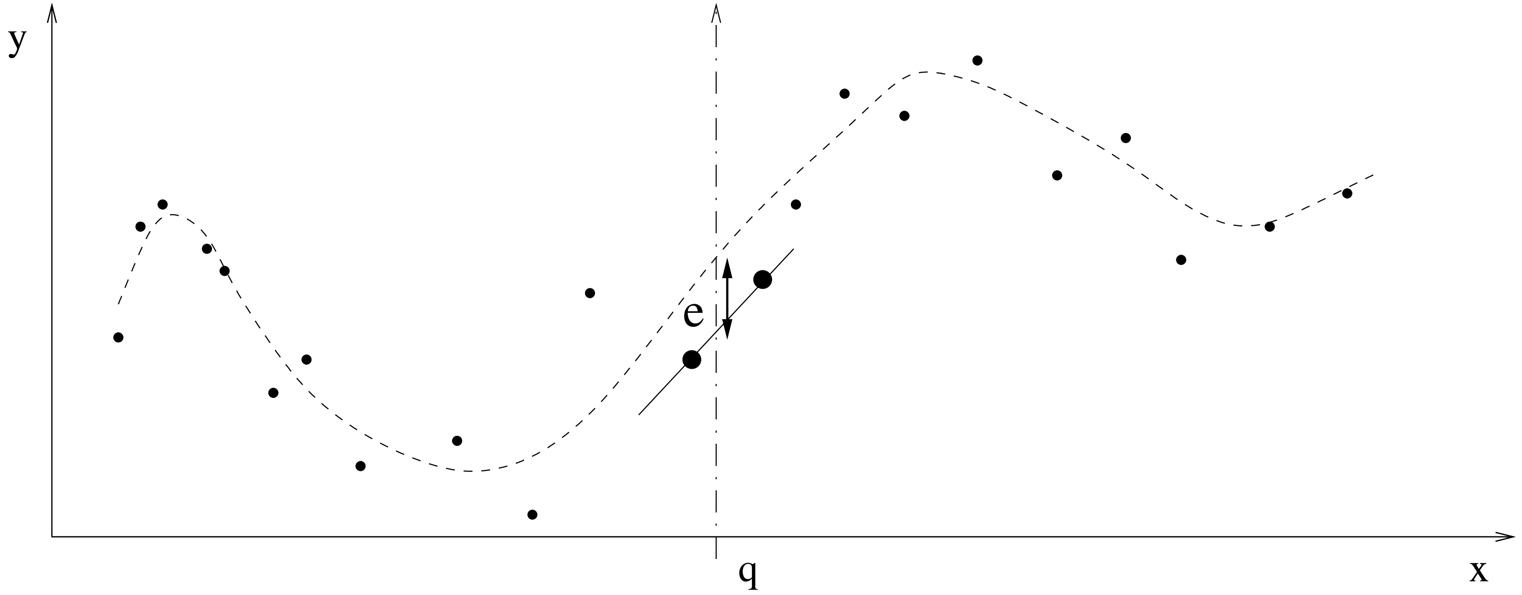
Local modeling procedure

The identification of a local model [3] can be summarized in these steps:

1. Compute the distance between the query and the training samples according to a predefined *metric*.
2. Rank the neighbors on the basis of their distance to the query.
3. Select a subset of the nearest neighbors according to the *bandwidth* which measures the size of the neighborhood.
4. Fit a *local model* (e.g. constant, linear,...).

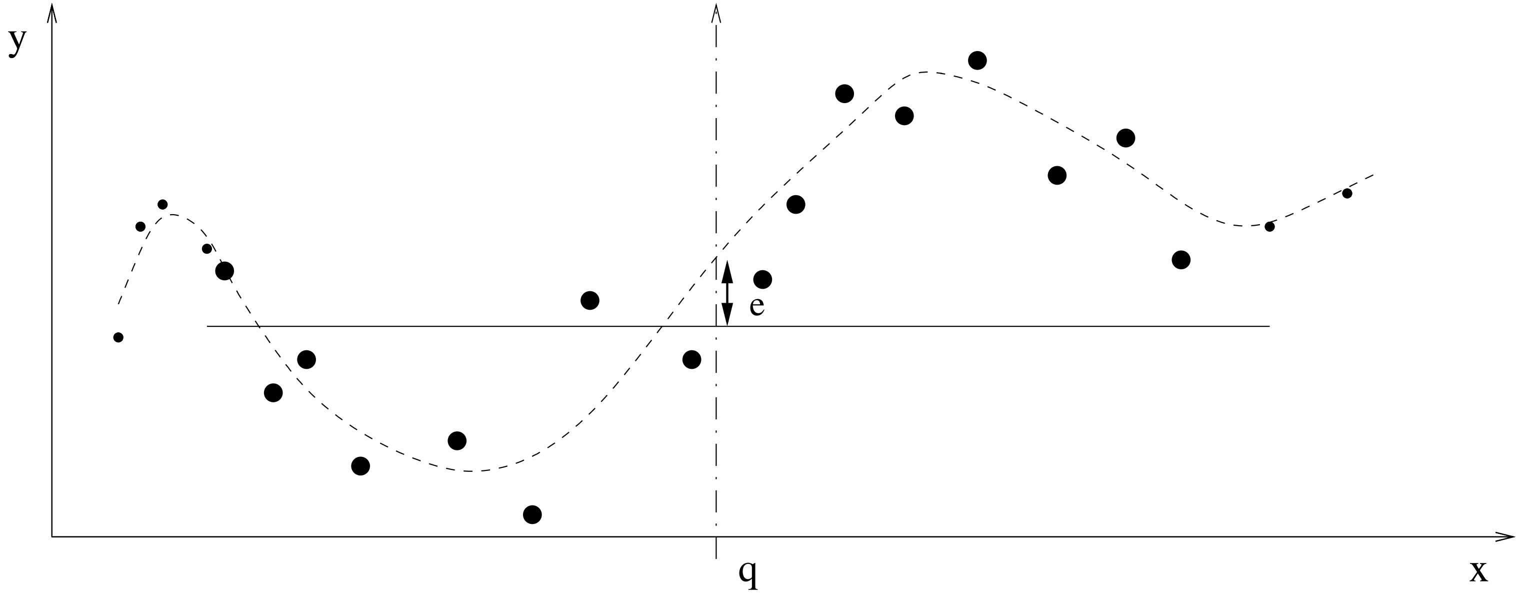
Each of the local approaches has one or more structural (or smoothing) parameters that control the amount of smoothing performed.
In this talk we will focus on the *bandwidth selection*.

The bandwidth trade-off: overfit



Too narrow bandwidth ⇒ overfitting ⇒ large prediction error e.
In terms of bias/variance trade-off, this is typically a situation of high variance.

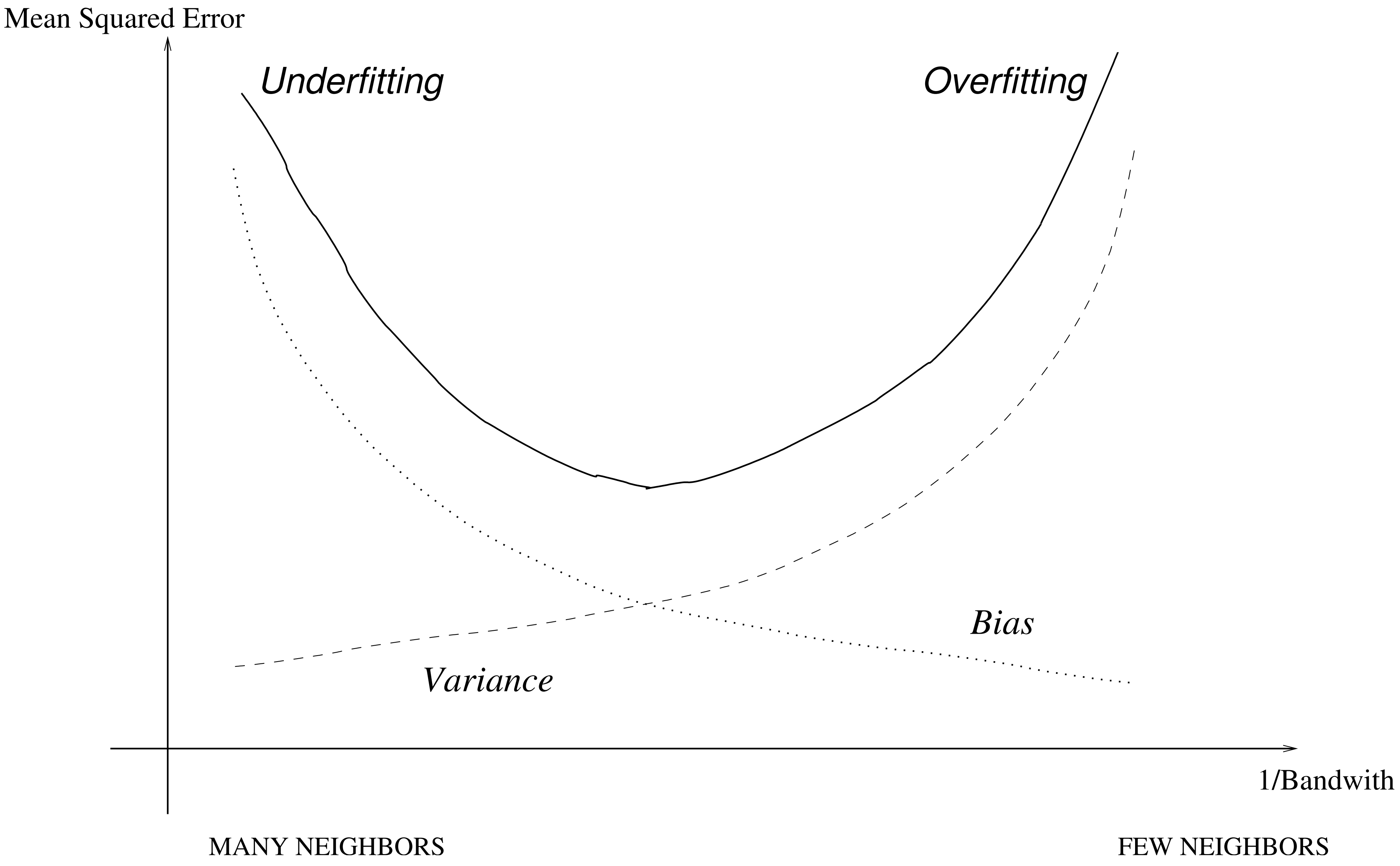
The bandwidth trade-off: underfit



Too large bandwidth ⇒ underfitting ⇒ large prediction error e

In terms of bias/variance trade-off, this is typically a situation of high bias.

Bandwidth and bias/variance trade-off



Existing work on bandwidth selection

Rule of thumb methods. They provide a crude bandwidth selection which in some situations may result sufficient. Examples of rule of thumb are in [25],[27].

Plug-in techniques. The exact expression of optimal bandwidth can be obtained from the asymptotic expressions of bias and variance, which unfortunately depends on unknown terms. The idea of the direct plug-in method is to replace these terms with estimates. This method was first introduced by Woodrofe [47] in density estimation. Examples of plug-in methods for non parametric regression are reported in Ruppert et al. [41].

Data-driven estimation. It is a selection procedure which estimates the generalization error directly from data. Unlike the previous approach, this method does not rely on the asymptotic expression but it estimates the values directly from the finite data set. To this group belong methods like cross-validation, Mallow's C_p, Akaike's AIC and other extensions of methods used in classical parametric modeling.

Existing work (II)

- Debate on the superiority of plug-in methods over data-driven methods is still open and the experimental evidences are contrasting. Results on behalf of plug-in methods come from [47, 41, 38].

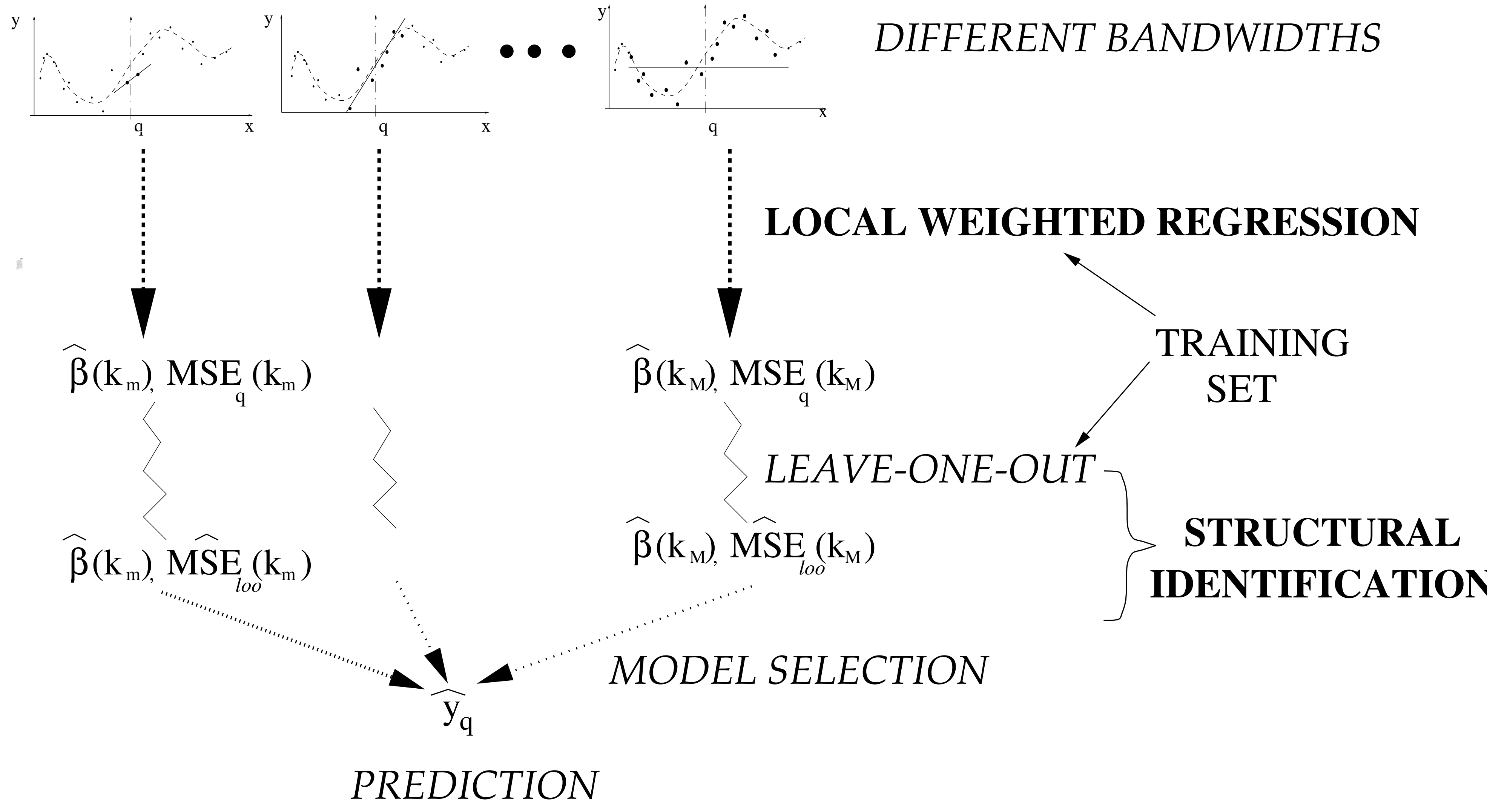
- Loader [33] showed how the supposed superior performance of plug-in approaches is a complete myth. The use of cross-validation for bandwidth selection has been investigated in several papers, mainly in the case of density estimation [30].

- In regression an adaptation of Mallow's C_p was introduced by Rice [40] for constant fitting and by Cleveland and Devlin [21] in local polynomial regression. Cleveland and Loader [22] suggested local C_p and local PRESS for choosing both the degree of local polynomial mixing and the bandwidth.

- We believe that plug-in methods are built on a series of assumptions about the statistical process underlying the data set and on theoretical results which are more reliable more the number of points tends to infinity.

- In a common black-box situation where no a priori information is available, the adoption of data driven techniques can result a promising approach to the problem.

Data-driven bandwidth selection



DIFFERENT BANDWIDTHS

LOCAL WEIGHTED REGRESSION

TRAINING SET

$\widehat{\beta}(k_m)$, $MSE_q(k_m)$

$\widehat{\beta}(k_M)$, $MSE_q(k_M)$

LEAVE-ONE-OUT

$\widehat{\beta}(k_m)$, $\widehat{MSE}_{loo}(k_m)$

$\widehat{\beta}(k_M)$, $\widehat{MSE}_{loo}(k_M)$

STRUCTURAL IDENTIFICATION

MODEL SELECTION

$\widehat{y}_q$

PREDICTION

Original contributions

Problem1: identifying a sequence of local models is expensive.

Solution1: we propose *recursive-least-squares* (RLS) to speed up the identification of sequence of models with increasing number of neighbors [6, 13].

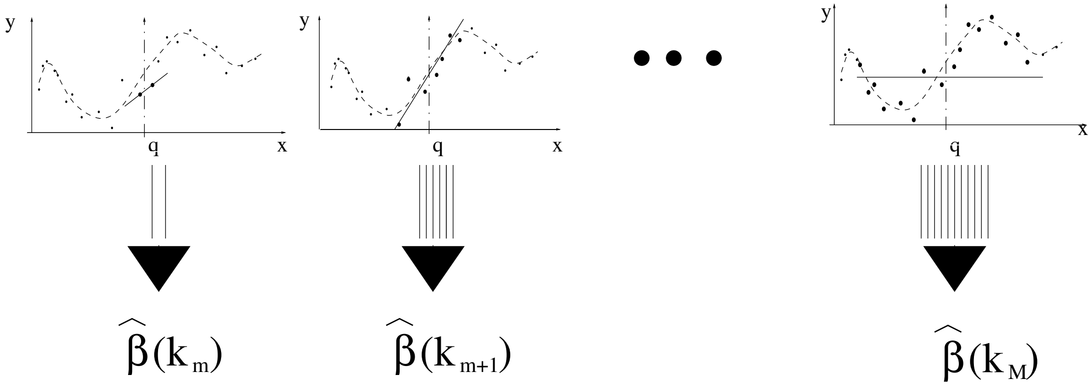
Problem 2: validating a local model by cross-validation is expensive.

Solution 2: we compute the leave-one-out cross-validation by obtaining the *PRESS statistic* through the terms of RLS [9].

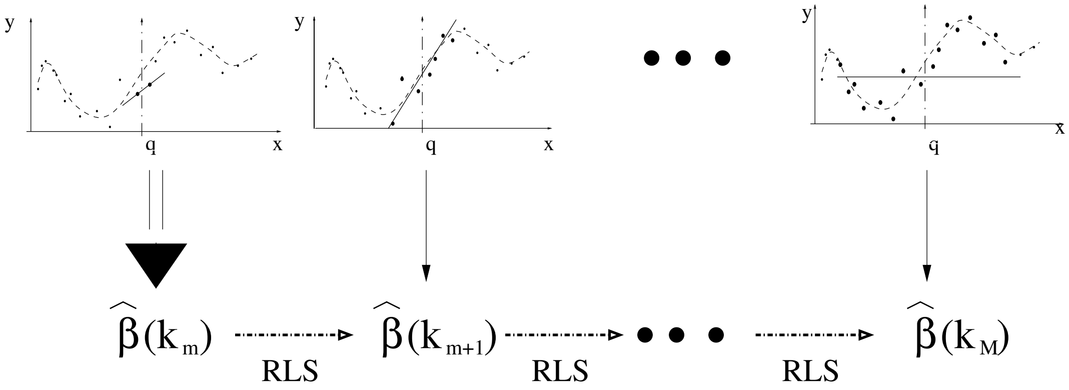
Problem 3: choosing the best model is prone to errors.

Solution 3: we *combine* the best models [7].

Recursive-least-squares in space

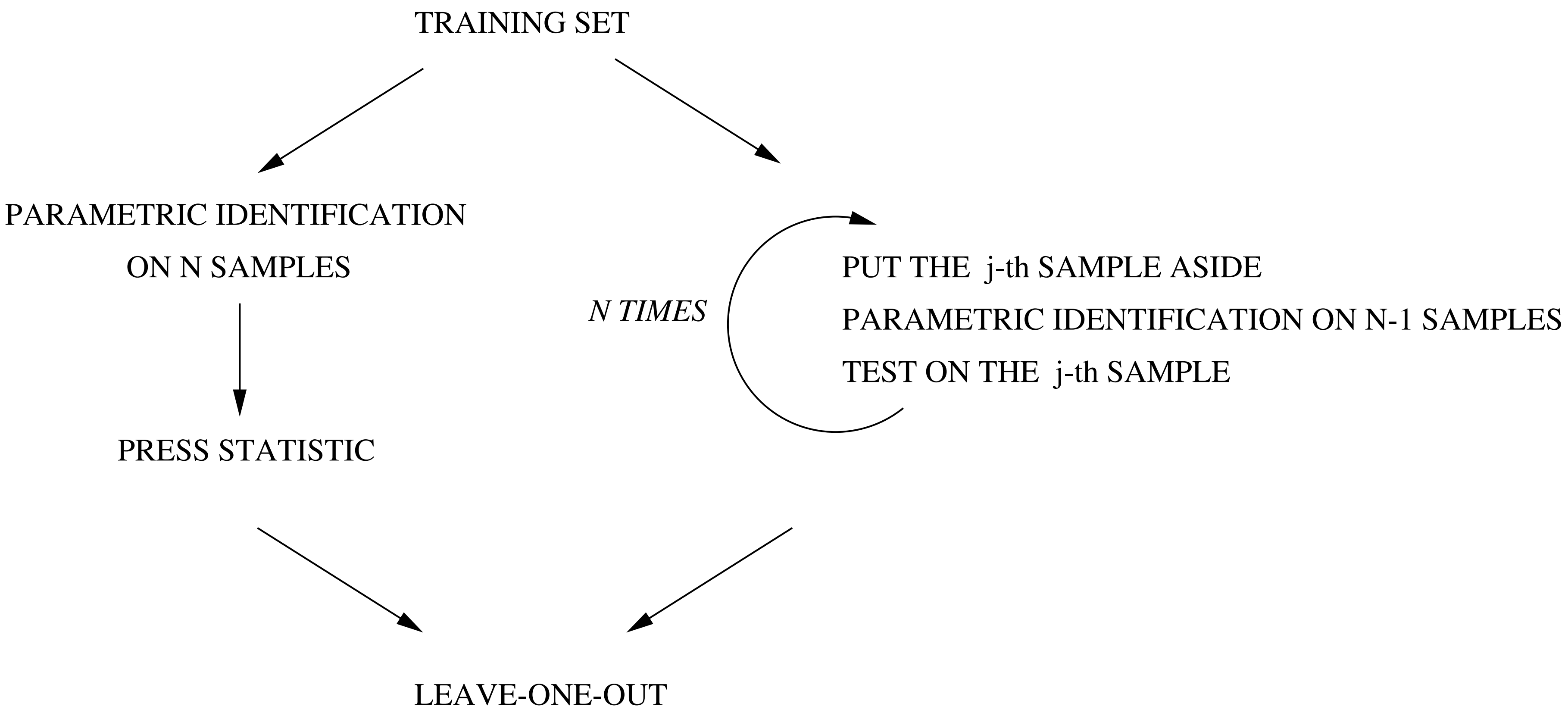


SLOW IDENTIFICATION



FAST IDENTIFICATION

PRESS statistic and leave-one-out



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

PRESS was first introduced by Allen [2].

The regression task

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

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

where $\forall i$,

- ε_i is a random variable such that $E[\varepsilon_i] = 0$ and $E[\varepsilon_i \varepsilon_j] = 0$, $\forall j \neq i$,
- $E[\varepsilon_i^m] = \mu_m(\mathbf{x}_i)$, $\forall m \geq 2$, where $\mu_m(\cdot)$ is the unknown m^{th} moment of the distribution of ε_i and is defined as a function of $\mathbf{x}_i$.

In particular for $m = 2$, the last of the above mentioned properties implies that no assumption of global homoscedasticity is made.

Local Weighted Regression

- The problem of local regression can be stated as the problem of estimating the value that the regression function $f(\mathbf{x}) = E[y|\mathbf{x}]$ assumes for a specific query point $\mathbf{x}$, using information pertaining only to a neighborhood of $\mathbf{x}$.
- Given a query point $\mathbf{x}_q$, and under the hypothesis of a local homoscedasticity of ε_i, the parameter $\boldsymbol{\beta}$ of a local linear approximation of $f(\cdot)$ in a neighborhood of $\mathbf{x}_q$ can be obtained solving the local polynomial regression:

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

  where, given a metric on the space $\Re^n$,

  - $d(\mathbf{x}_i, \mathbf{x}_q)$ is the distance from the query point to the i^{th} example, $i = 1, \ldots, N$,
  - $K(\cdot)$ is a weight (aka kernel) function,
  - h is the bandwidth

Local Weighted Regression (II)

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

  $$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{W}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}'\mathbf{W}\mathbf{y} = (\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{v} = \mathbf{P}\mathbf{Z}'\mathbf{v},$$

  where $\mathbf{X}$ is a matrix whose i^{th} row is $\mathbf{x}_i'$, $\mathbf{y}$ is a vector whose i^{th} element is y_i, $\mathbf{W}$ is a diagonal matrix whose i^{th} diagonal element is $w_{ii} = \sqrt{K\left(d(\mathbf{x}_i, \mathbf{x}_q)/h\right)}$, $\mathbf{Z} = \mathbf{W}\mathbf{X}$, $\mathbf{v} = \mathbf{W}\mathbf{y}$, and the matrix $\mathbf{X}'\mathbf{W}'\mathbf{W}\mathbf{X} = \mathbf{Z}'\mathbf{Z}$ is assumed to be non-singular so that its inverse $\mathbf{P} = (\mathbf{Z}'\mathbf{Z})^{-1}$ is defined.

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

  $$\hat{y}_q = \mathbf{x}_q'\hat{\boldsymbol{\beta}}.$$

Linear Leave-one-out

- By exploiting the linearity of the local approximator, a leave-one-out cross-validation estimation of the mean squared error $E[(f(x_q) - \hat{y}_q)^2]$ can be obtained without any significant overload.

- In fact, using the PRESS statistic [2, 37], it is possible to calculate the error $e_j^{\text{cv}} = y_j - \mathbf{x}_j' \hat{\boldsymbol{\beta}}_{-j}$, without explicitly identifying the parameters $\hat{\boldsymbol{\beta}}_{-j}$ from the examples available with the j^{th} removed.

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

$$e_j^{\text{cv}} = y_j - \mathbf{x}_j' \hat{\boldsymbol{\beta}}_{-j} = \frac{y_j - \mathbf{x}_j' \mathbf{P} \mathbf{Z}' \mathbf{v}}{1 - \mathbf{z}_j' \mathbf{P} \mathbf{z}_j} = \frac{y_j - \mathbf{x}_j' \hat{\boldsymbol{\beta}}}{1 - h_{jj}},$$

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

Rectangular weight function

- In what follows, for the sake of simplicity, we will focus on linear approximator. An extension to generic polynomial approximators of any degree is straightforward. We will assume also that a metric on the space $\Re^n$ 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

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

  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 k^{th} nearest neighbor of the query point.

Recursive local regression

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

$$\begin{cases} \mathbf{P}(k+1) = \mathbf{P}(k) - \dfrac{\mathbf{P}(k)\mathbf{x}(k+1)\mathbf{x}'(k+1)\mathbf{P}(k)}{1+\mathbf{x}'(k+1)\mathbf{P}(k)\mathbf{x}(k+1)} \\ \gamma(k+1) = \mathbf{P}(k+1)\mathbf{x}(k+1) \\ e(k+1) = y(k+1) - \mathbf{x}'(k+1)\hat{\beta}(k) \\ \hat{\beta}(k+1) = \hat{\beta}(k) + \gamma(k+1)e(k+1) \end{cases}$$

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

Recursive PRESS computation

Moreover, once the matrix $\mathbf{P}(k+1)$ is available, the leave-one-out cross-validation errors can be directly calculated without the need of any further model identification:

$$e_j^{\text{cv}}(k+1) = \frac{y_j - \mathbf{x}_j' \hat{\beta}(k+1)}{1 - \mathbf{x}_j' \mathbf{P}(k+1) \mathbf{x}_j}, \qquad \forall j: \ d(\mathbf{x}_j, \mathbf{x}_q) \leq h(k+1).$$

Let us define for each value of k the $[k \times 1]$ vector $\mathbf{e}^{\text{cv}}(k)$ that contains all the leave-one-out errors associated to the model $\hat{\beta}(k)$.

Model selection

- The recursive algorithm returns for a given query point $\mathbf{x}_q$, a set of predictions $\hat{y}_q(k) = \mathbf{x}_q' \hat{\boldsymbol{\beta}}(k)$, together with a set of associated leave-one-out error vectors $\mathbf{e}^{\text{cv}}(k)$.
- 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:

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

$$\text{with } \hat{k} = \arg\min_k \widehat{\text{MSE}}(k) = \arg\min_k \frac{\sum_{i=1}^{k} \omega_i \left(\mathbf{e}_i^{\text{cv}}(k)\right)^2}{\sum_{i=1}^{k} \omega_i};$$

Local Model combination

- As an alternative to the *winner-takes-all* paradigm, we explored also the effectiveness of local combinations of estimates [46].
- 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 $\mathbf{e}^{\text{cv}}(k)$ have been ordered creating a sequence of integers $\{k_i\}$ so that $\widehat{\text{MSE}}(k_i) \leq \widehat{\text{MSE}}(k_j)$, $\forall i < j$. The prediction of $\hat{y}_q$ is given by

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

where the weights are the inverse of the mean square errors: $\zeta_i = 1/\widehat{\text{MSE}}(k_i)$. This is an example of the *generalized ensemble method* [39].

From local learning to Lazy Learning (LL)

- By speeding up the local learning procedure, we can delay the learning procedure to the moment when a prediction in a query point is required (*query-by-query* learning).
- The combination approach makes possible to integrate local models of different order (e.g. constant and linear) and different bandwidths.
- This method is called *lazy* since the whole learning procedure (i.e. the parametric and the structural identification) is deferred until a prediction is required.

Experimental setup for regression

Datasets: 23 real and artificial datasets from the ML repository.

Methods: Lazy Learning, Local modeling, Feed Forward Neural Networks, Mixtures of Experts, Neuro Fuzzy, Regression Trees (Cubist).

Experimental methodology: 10-fold cross-validation.

Results: Mean absolute error (Table 7.2), relative error (Table 7.3) and paired t-test (Appendix C) [7].

Regression datasets

Dataset	Number of examples	Number of regressors
Housing	330	8
Cpu	506	13
Prices	209	6
Mpg	159	16
Servo	392	7
Ozone	167	8
Bodyfat	252	13
Pool	253	3
Energy	2444	5
Breast	699	9
Abalone	4177	10
Sonar	208	60
Bupa	345	6
Iono	351	34
Pima	768	8
Kin_8fh	8192	8
Kin_8nh	8192	8
Kin_8fm	8192	8
Kin_8nm	8192	8
Kin_32fh	8192	32
Kin_32nh	8192	32
Kin_32fm	8192	32
Kin_32nm	8192	32

Experimental results: paired comparison

Each method is statistically compared with all the others (9 * 23 =207 comparisons).

Method	Number of times the method was significantly worse than another
LL linear	74
LL constant	96
LL combination	23
Local modeling linear	58
Local modeling constant	81
Cubist	40
Feed Forward NN	53
Mixtures of Experts	80
Local Model Network (fuzzy)	132
Local Model Network (k-mean)	145

The less, the best !!

Award in EUFIT competition

Data analysis competition on regression: awarded as a runner-up among 21 participants at the Third International Erudit competition on *Protecting rivers and streams by monitoring chemical concentrations and algae communities* [10].

Lazy Learning for dynamic tasks

Multi-step-ahead prediction: [12]

long horizon forecasting based on the iteration of a LL one-step-ahead predictor.

Nonlinear control: [11]

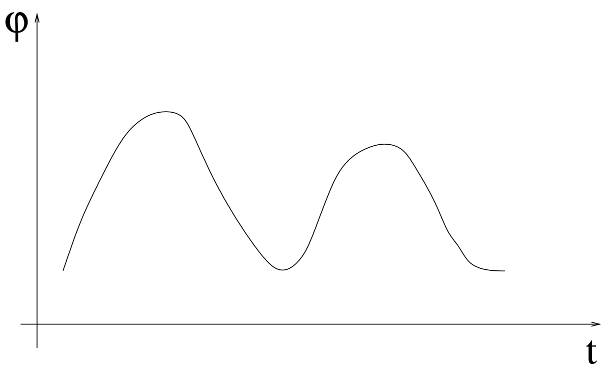
1. Lazy Learning inverse/forward control.
2. Lazy Learning self-tuning control.
3. Lazy Learning optimal control.

Embedding in time series

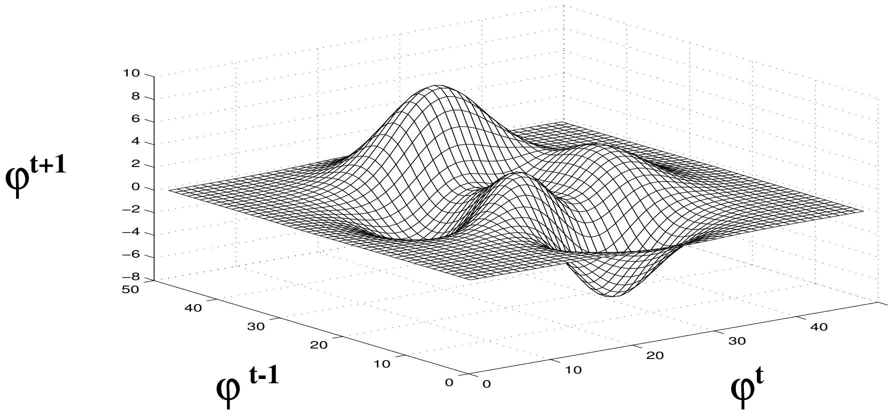
Consider a sequence $\mathcal{S}$ of measurements $\varphi^t \in \Re$ of a observable at equal time intervals.
We express the present value as a function of the previous n values of the time series itself

$$\varphi^t = f(\varphi^{t-1}, \varphi^{t-2}, \dots, \varphi^{t-n}) + \varepsilon$$

where f is an unknown nonlinear function and the vector $[\varphi^{t-1}, \varphi^{t-2}, \dots, \varphi^{t-n}]$ lies in the n dimensional time delay space or *lag space*.
This standard approach is called “state-space reconstruction” in the physics community, “tapped delay line” in the engineering community and Nonlinear Autoregressive (NAR) in the forecasting community.

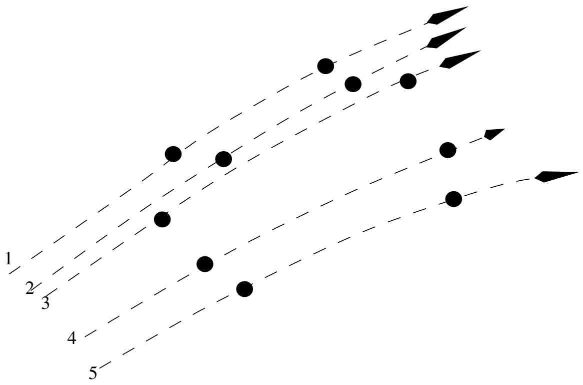


temporal representation



input/output representation

TIME SERIES $\varphi^{t+1} = f(\varphi^{t}, \varphi^{t-1}, \ldots, \varphi^{t-n+1})$



embedding representation

One-step and multi-step-ahead prediction

One-step ahead prediction: the n previous values of the series are assumed to be available for the prediction of the next value. This is equivalent to a problem of supervised learning. LL was used in this way in several prediction tasks: finance, economic variables, environmental modeling [23].

Multi-step ahead prediction: we predict the value of the series for the next h steps. We can classify the methods for multiple step prediction according to two features, the *horizon* of the predictor and the *training criterion*.

Multi-step-ahead-prediction

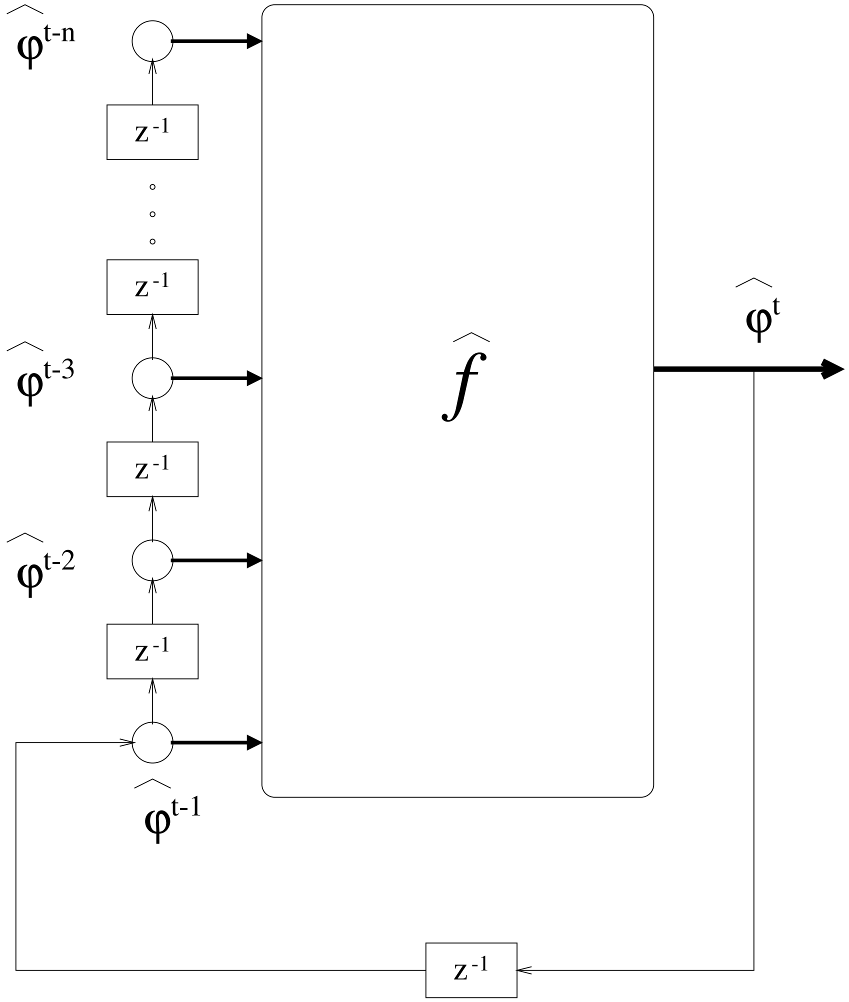
One-step-ahead predictor and one-step-ahead training criterion. The model predicts h steps ahead by iterating a one-step-ahead predictor whose parameters are optimized to minimize the training error on one-step-ahead forecast.

One-step-ahead predictor and h-step-ahead training criterion. The model predicts h steps ahead by iterating a one-step-ahead predictor whose parameters are optimized to minimize the training error on the iterated h-step-ahead forecast.

Direct forecasting. The model makes a direct forecast at time $t+h$:

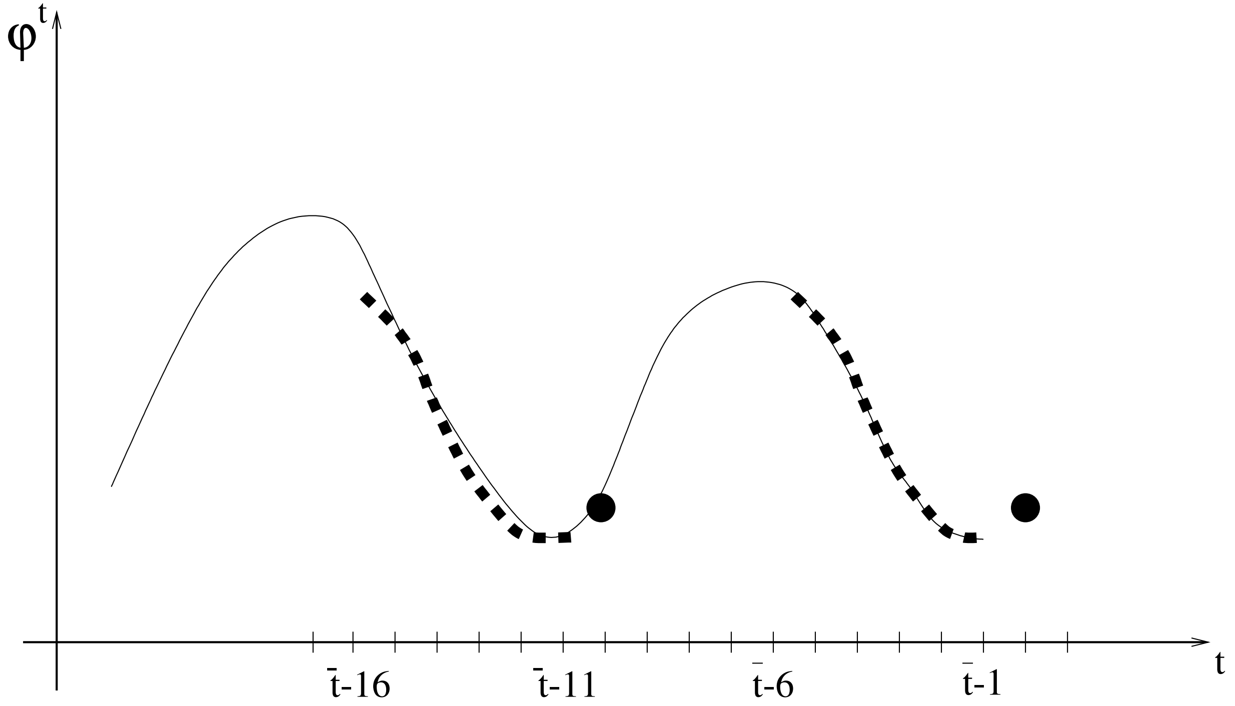
$$\varphi^{t+h} = f^h(\varphi^t, \varphi^{t-1}, \dots, \varphi^{t-n+1})$$

Iteration of a one-step-ahead predictor


$\widehat{\varphi}^{t-n}$
z^{-1}
z^{-1}
$\widehat{\varphi}^{t-3}$
z^{-1}
$\widehat{\varphi}^{t-2}$
z^{-1}
$\widehat{\varphi}^{t-1}$
$\widehat{f}$
$\widehat{\varphi}^{t}$
z^{-1}

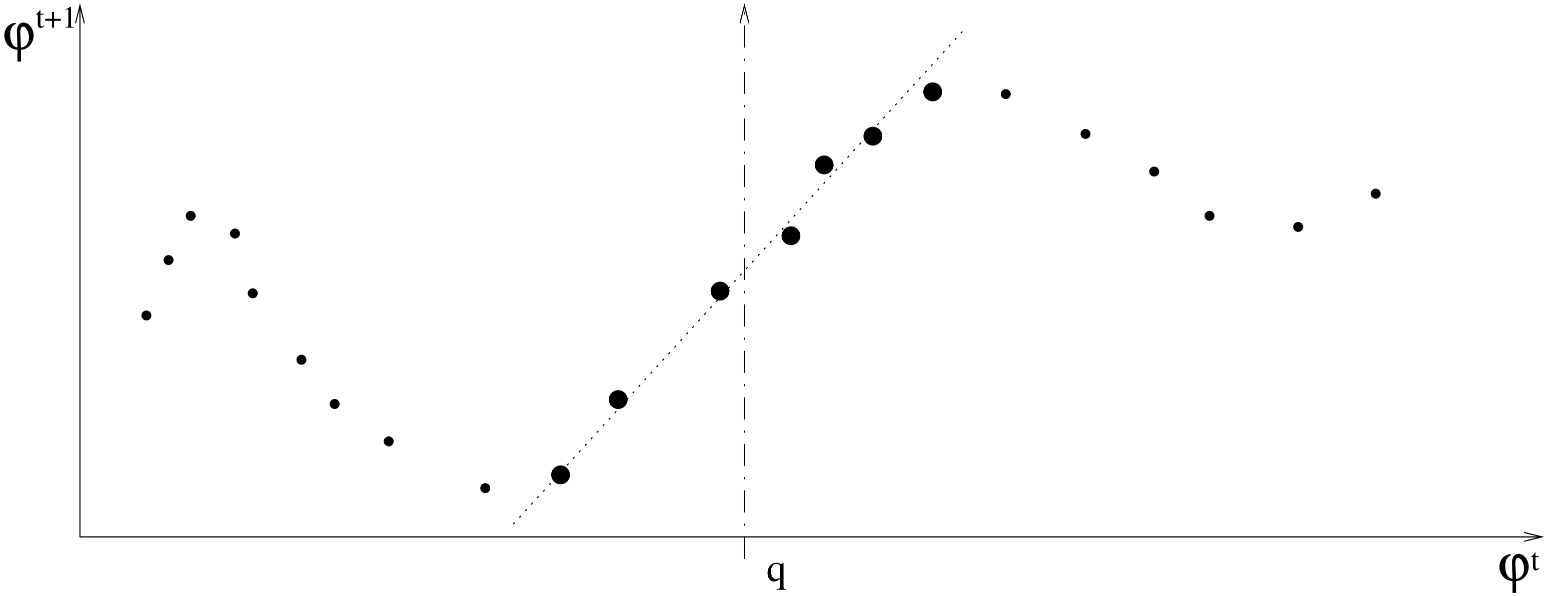
Local Modeling in the time domain

Consider the embedding $\varphi^{t+1} = f(\varphi^t, \varphi^{t-1}, \ldots, \varphi^{t-5})$ of order $n = 6$.



Local Modeling in the I/O space

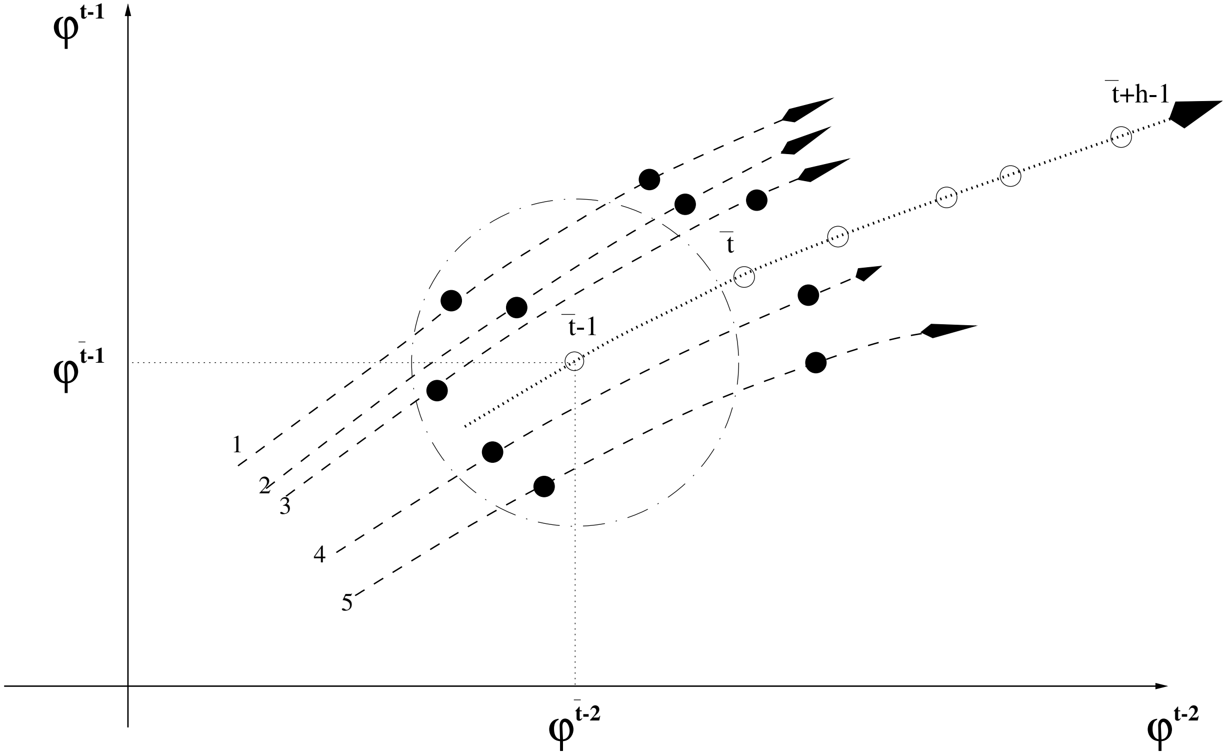
Consider the embedding $\varphi^{t+1} = f(\varphi^t)$ of order $n = 1$.



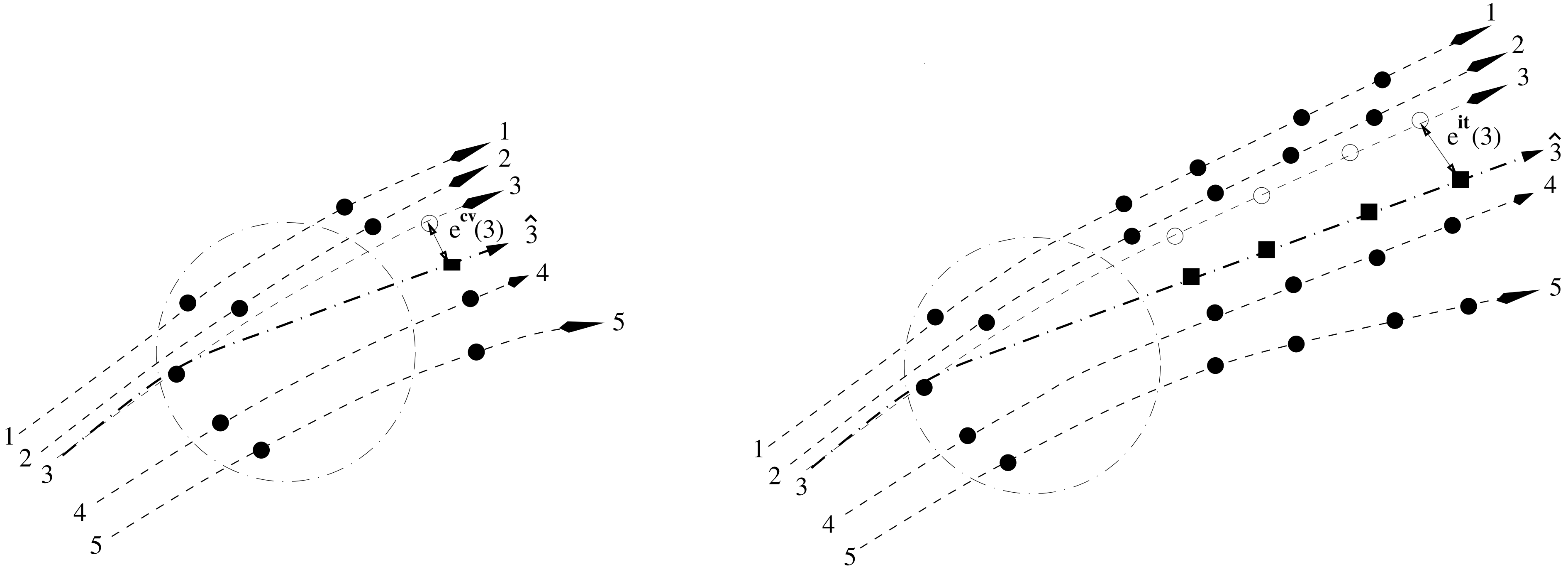
Note the labels of the axis !!!

Local modeling in the embedding space

Consider the embedding $\varphi^{t+1} = f(\varphi^{t}, \varphi^{t-1})$ of order $n = 2$.

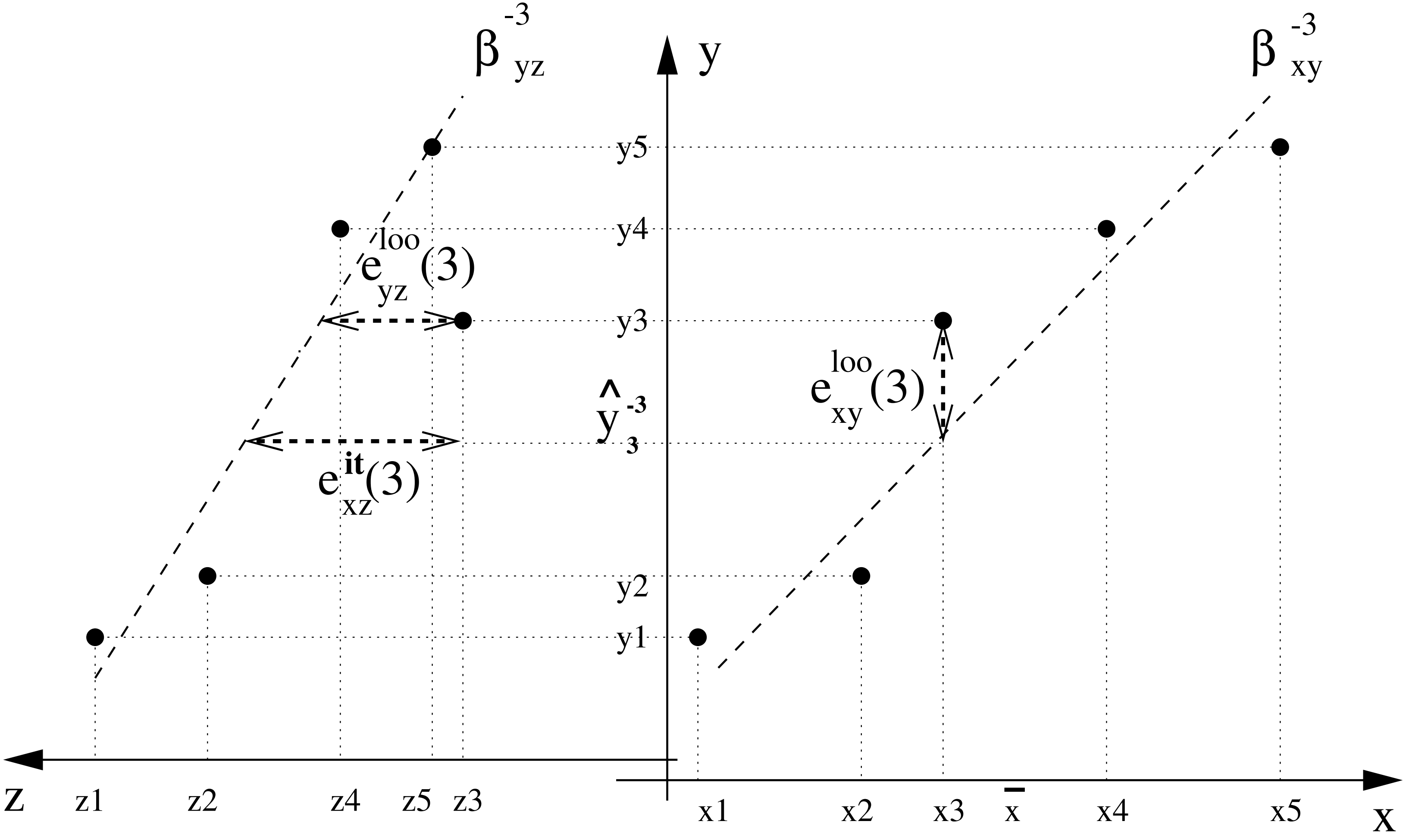


Conventional and iterated leave-one-out



a) b)

It Press in the space $X \to Y \to Z$



x represents the value of the time series with order $n = 1$ at time $t - 1$, y represents the value of the time series at time t, and z represents the value of the time series at time $t + 1$.

From conventional to iterated PRESS

- PRESS statistic returns leave-one-out as a by product of the local weighted regression.
- We derived in [12] an analytical iterated formulation of the PRESS statistic for long horizon assessment.
- Iterated assessment criterion improves stability and prediction accuracy.

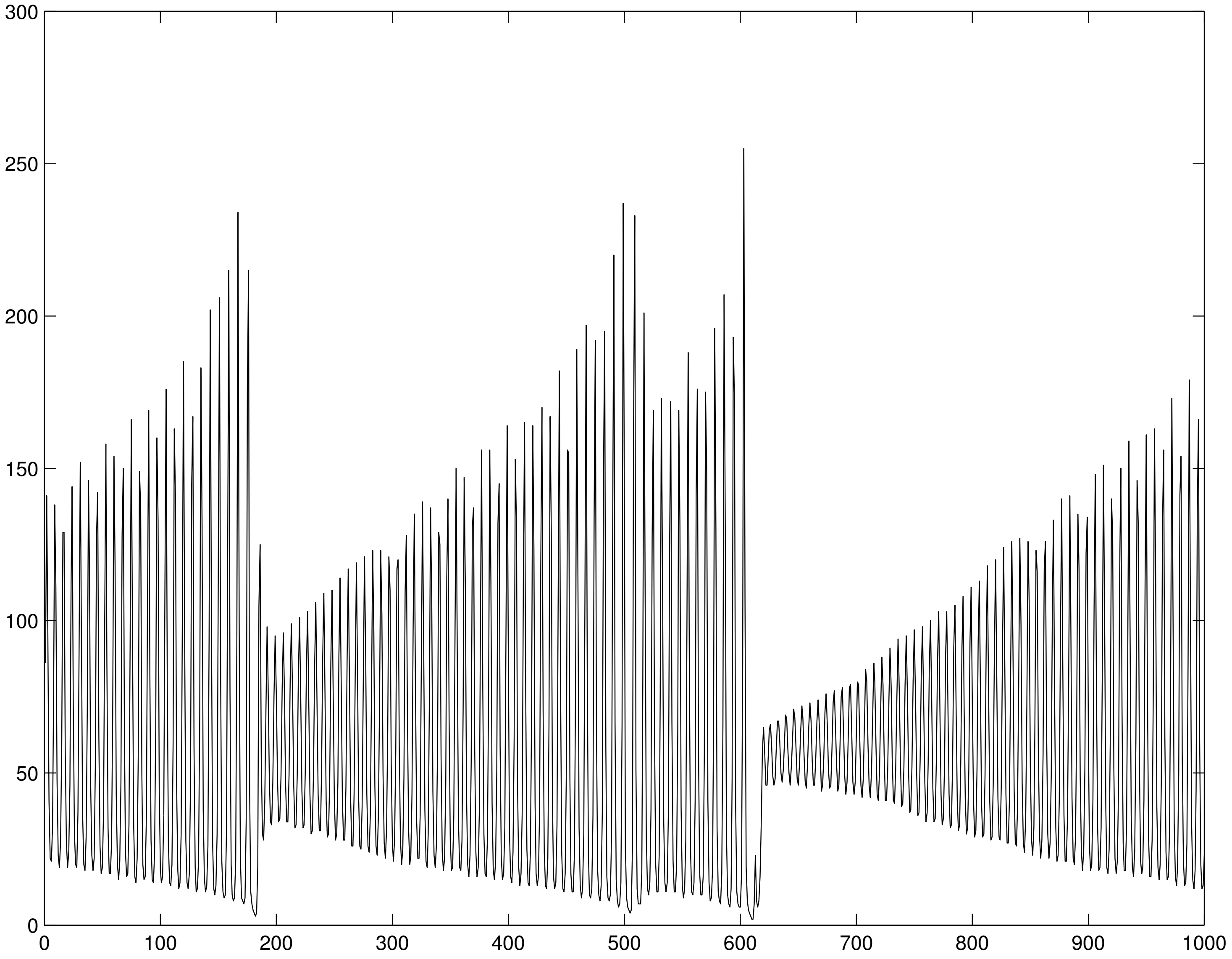
The Iterated multi-step-ahead algo

1. Time series embedded as an input/output mapping $f : \Re^n \to \Re$.
2. The one-step-ahead predictor is a local estimate of the mapping f.
3. The h-step-ahead prediction is performed by iterating a one-step-ahead estimator.
4. Local structure identification performed in a space of alternative model configurations, each characterized by a different bandwidth.
5. Prediction ability assessed by the iterated formulation of the cross-validation PRESS statistic (h-step-ahead criterion).

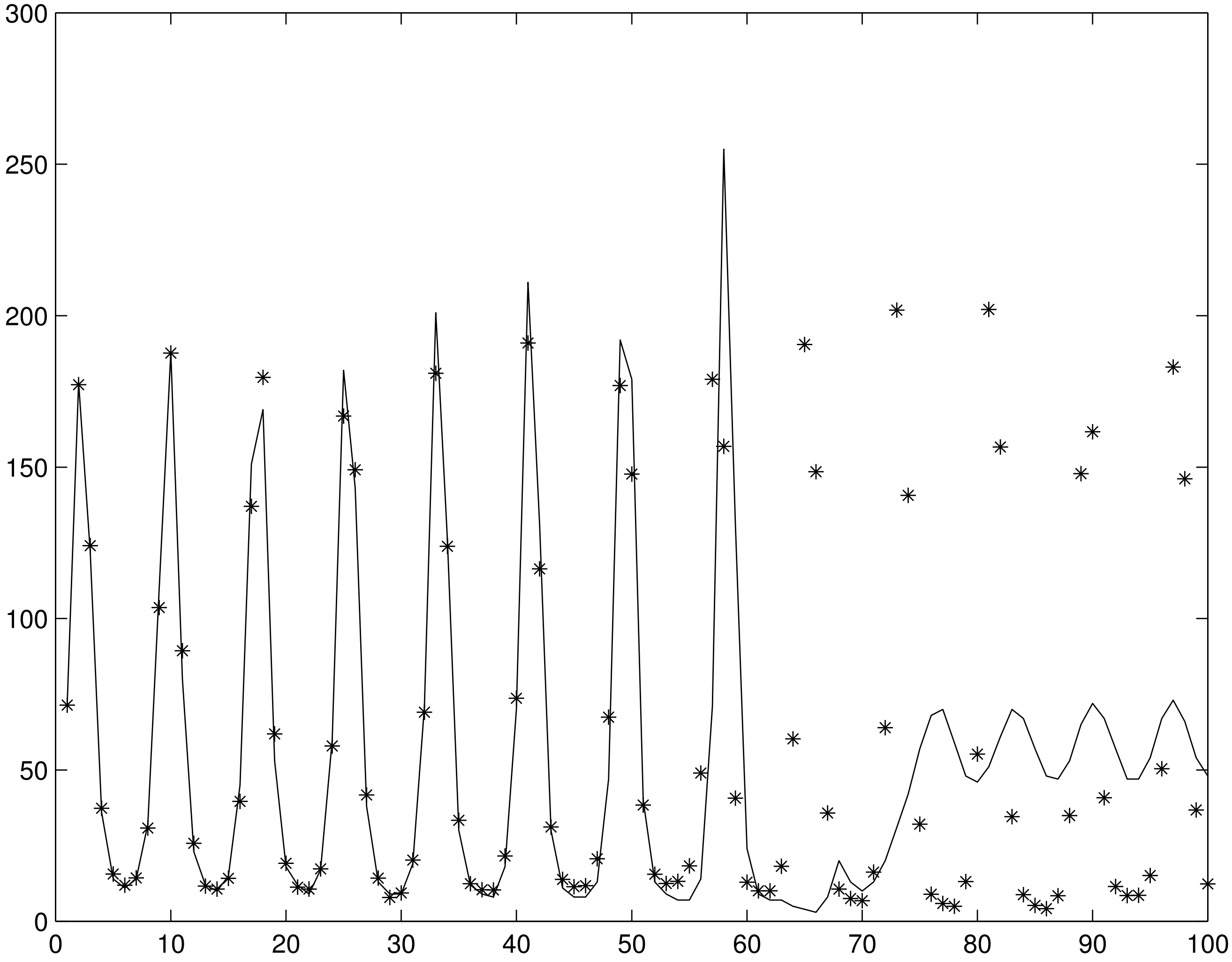
The Santa Fe time series

- The iterated PRESS approach has been applied both to the prediction of a *real-world* data set (*A*) and to a computer generated time series (*D*) from the *Santa Fe Time Series Prediction and Analysis Competition*.
- The *A* time series has a training set of 1000 values and a test set of 10000 samples: the task is to predict the continuation for 100 steps, starting from different points.
- The *D* time series has a training set of 100000 values and a test set of 500 samples: the task is to predict the continuation for 25 steps, starting from different points.

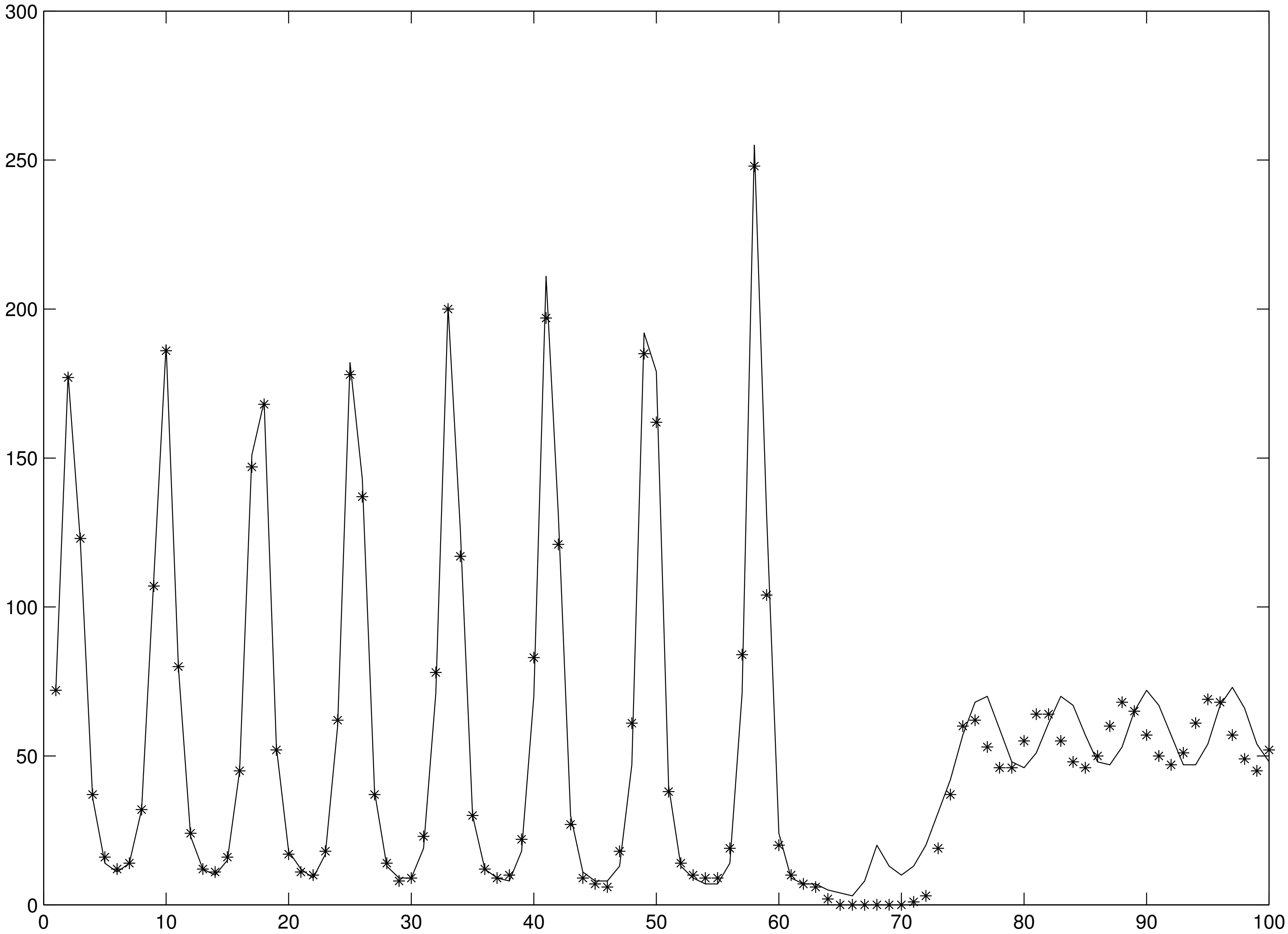
A series: training set



A series: one-step criterion



A series: multi-step criterion



Experiments: The Santa Fe Time Series A

order n=16 Training set: 1000 values Test set: 100 steps

Test data	Non iter. PRESS	Iter. PRESS	Sauer	Wan
1-100	0.350	0.029	0.077	0.055
1180-1280	0.379	0.131	0.174	0.065
2870-2970	0.793	0.055	0.183	0.487
3000-3100	0.003	0.003	0.006	0.023
4180-4280	1.134	0.051	0.111	0.160

Sauer: combination of iterated and direct local models.

Wan: recurrent network.

The Santa Fe Time Series D

order $n = 20$ Training set: $100,000$ values Test set: 25 steps

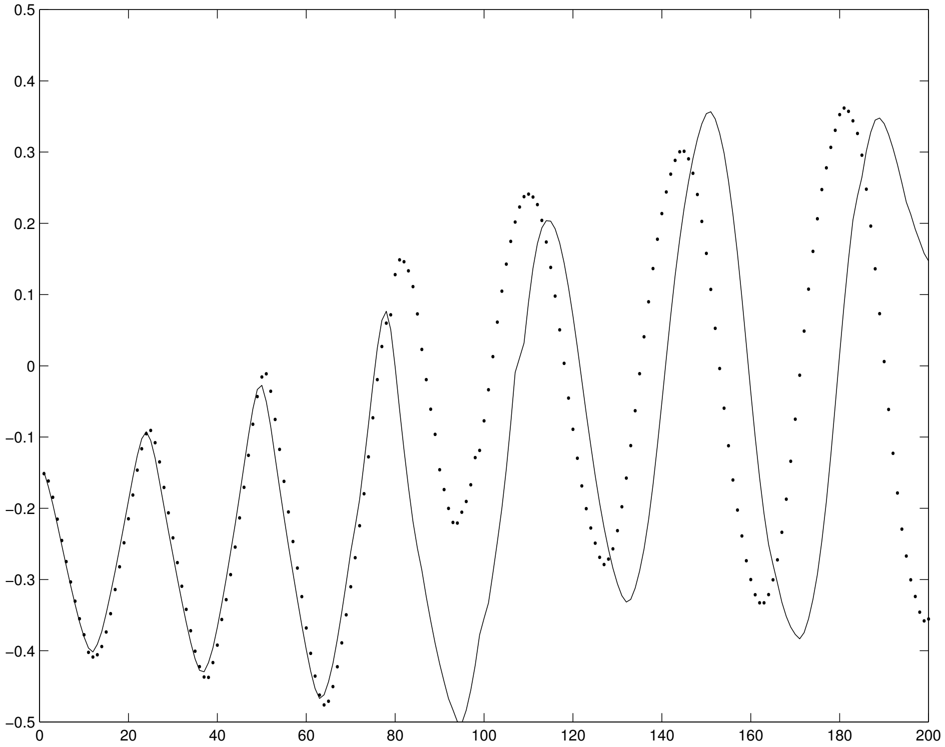
Test data	Non iter. PRESS	Iter. PRESS	Zhang Hutchinson
0-24	0.1255	0.0492	0.0665
100-124	0.0460	0.0363	0.0616
200-224	0.2635	0.1692	0.1475
300-324	0.0461	0.0405	0.0541
400-424	0.1610	0.0644	0.0720

Zhang: combination of iterated and direct multilayer perceptron.

Award in Leuven Competition

Training set made of 2000 points.

Task: predict the continuation for the next 200 points.



Iterated Lazy Learning ranked second and fourth [8].

Lazy Learning for iterated prediction

Multi-step ahead by iteration of a one-step predictor.

Lazy learning to implement the one-step predictor.

Selection of the local structure by an iterated PRESS.

Iterated criterion avoids the accumulation of prediction errors and improves the performance.

Complexity in global and local modeling

Consider N training samples, n features and Q query points.

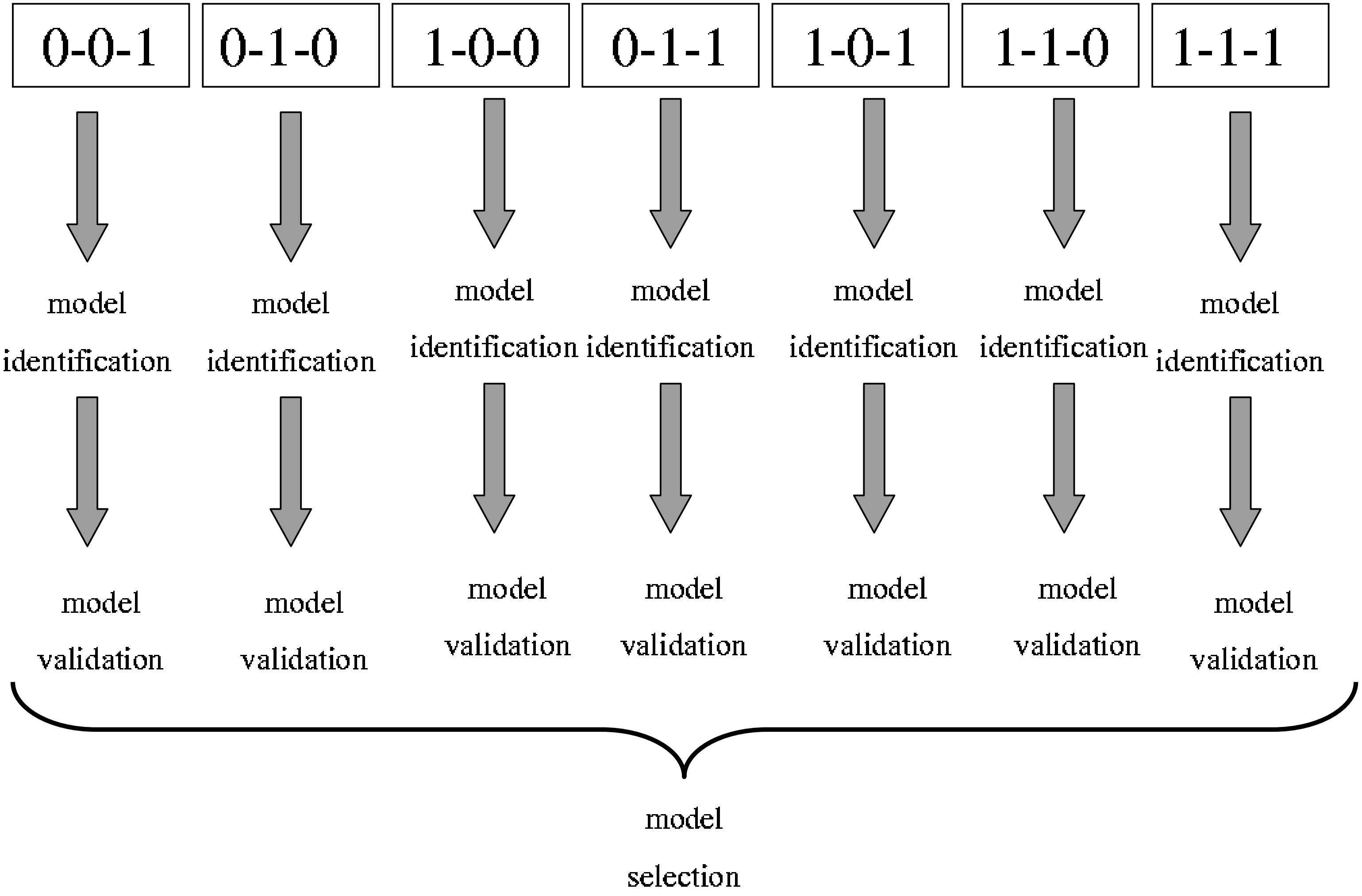
	GLOBAL	LAZY
Parametric ident.	$\mathcal{C}$(NLS)	$\mathcal{C}$(Nn)+$\mathcal{C}$(LS)
Structural ident. by K-fold cross-validation	K $\mathcal{C}$(NLS)	small
prediction for Q queries	negligible	Q ($\mathcal{C}$(Nn)+$\mathcal{C}$(LS))
TOTAL	K $\mathcal{C}$(NLS)	Q [$\mathcal{C}$(Nn)+$\mathcal{C}$(LS)]

where $\mathcal{C}$(NLS) stands for the cost of *Non-Linear least-Squares* and $\mathcal{C}$(LS) stands for the cost of *Linear least-Squares*.

Feature selection and LL

- Local modeling techniques are known to be weak in large dimensional spaces.
- A way to defy the curse of dimensionality is dimensionality reduction (aka feature selection).
- It requires the assessment of an exponential number of alternatives (2^n subsets of input variables) and the choice of the best one.
- Several techniques exist: we focus here on wrappers.
- Wrappers rely on expensive cross-validation (e.g. leave-one-out assessment)
- Our idea: combine racing [34] and sub-sampling [29] to accelerate the wrapper feature selection procedure in LL.

Wrapper feature selection: n=3



0-0-1 | 0-1-0 | 1-0-0 | 0-1-1 | 1-0-1 | 1-1-0 | 1-1-1

model identification

model validation

model selection

Racing for feature selection

- Suppose we have several sets of different input variables.
- The computational cost of making a selection results from the cost of identification and the cost of validation.
- The validation cost required by a global model is independent of Q, while this is not the case for LL.
- The idea of racing techniques consists in using **blocking** and **paired multiple test** to compare different models in similar conditions and discard as soon as possible the worst ones.
- Racing reduces the number of tests Q to be made.
- This makes more competitive the wrapper LL approach.

Model selection via leave-one-out (N=10)

	M1	M2	M3	M4	M5
	0.1	0.3	0.2	0	0.05
	0.4	0.6	0.5	0.1	0.2
	0.3	1.7	0.4	0.1	0.4
	0.7	2.5	1.2	0.9	0.8
	0.5	2	1	0.4	0.5
	2	3.1	2.7	1.9	2.4
	0.1	4	3.5	0	3
	4	5.2	5.3	3.5	8.4
	3.2	4	3.9	3.4	4.2
	4	4	4	0.2	3.9
estimated MSE=	1.5	2.7	2.2	1.0	2.4

50 predictions

WINNER

Model selection by racing (N=10)

M1	M2	M3	M4	M5
0.1	0.3	0.2	0	0.05
0.4	0.6	0.5	0.1	0.2
0.3	**OUT**	0.4	0.1	0.4
0.7		1.2	0.9	0.8
0.5		1	0.4	0.5
2		**OUT**	1.9	2.4
0.1			0	3
4			3.5	**OUT**
3.2			3.4	
4			0.2	
OUT			**WINNER**	

34 predictions

Sub-sampling and LL

- The goal of model selection is to find the best hypothesis in a set of alternatives.
- What is relevant is ordering the different alternatives: M2 > M3 > M5 > M1> M2.
- Reducing the training set size N, we hope to reduce the accuracy of each single model but not necessarily their ordering.
- In LL reducing the training set size N reduces the cost.
- The idea of sub-sampling is to reduce the size of the training set without altering the ranking of the different models.
- This makes more competitive the LL approach

RACSAM for feature selection

We proposed the following algorithm [14]

1. Define an initial group of promising feature subsets.
2. Start with small training and test sets.
3. Discard by racing all the feature subsets that appear as significantly worse than the others.
4. Increase the training and test size until at most W winners models remain.
5. Update the group with new candidates to be assessed and go back to 3.

Experimental session

- We compare the performance accuracy of the LL algorithm enhanced by the RACSAM procedure to the the accuracy of two state-of-art algorithms, a SVM for regression and a regression tree (RTREE).
- Two version of the RACSAM algorithm were tested: the first (LL-RAC1) takes as feature set the best one (in terms of estimate Mean absolute Error (MAE)) among the W winning candidates : the second (LL-RAC1) averages the predictions of W LL predictors.
- $W = 5$, and p-value is 0.01.

Experimental results

Five-fold cross-validation on six real datasets of high dimensionality: `Ailerons` $(N = 14308, n = 40)$, `Pole` $(N = 15000, n = 48)$, `Elevators` $(N = 16599, n = 18)$, `Triazines` $(N = 186, n = 60)$, `Wisconsin` $(N = 194, n = 32)$ and `Census` $(N = 22784, n = 137)$.

Dataset	AIL	POL	ELE	TRI	WIS	CEN
LL-RAC1	9.7e-5	3.12	1.6e-3	0.21	27.39	0.17
LL-RAC2	9.0e-5	3.13	1.5e-3	0.12	27.41	0.16
SVM	1.3e-4	26.5	1.9e-3	0.11	29.91	0.21
RTREE	1.8e-4	8.80	3.1e-3	0.11	33.02	0.17

Applications

- Financial prediction of stock markets: in collaboration with **Masterfood**, Belgium.
- Prediction of yearly sales: in collaboration with **Dieteren**, Belgium, the first Belgian car dealer.
- Non linear control and identification task in power systems: in collaboration with **Universitá del Sannio** (I) [44, 18].
- Modeling of industrial processes: in collaboration with **FaFer Usinor** steel company (B), and **Honeywell** Technology Center, (US).
- Performance modelling of embedded systems: during my stay at **Philips Research** [16], Eindhoven (NL).
- Quality of service: during my stay at **IMEC**, Leuven (B) [17].
- Black-box simulators: in collaboration with **CENEARO**, Gosselies (B) [15].
- Environmental predictions: in collaboration with **Politecnico di Milano** (I) [23].

Software

- MATLAB toolbox on Lazy Learning [5].
- R contributed package `lazy`.
- Joint work with Dr. Mauro Birattari (IRIDIA).
- Web page: `http://iridia.ulb.ac.be/~lazy`.
- About 5000 accesses since October 2002.

The importance of being Lazy

- Fast data-driven design.
- No global assumption on the noise.
- Linear methods still effective in a multivariate non-linear setting (LWR, PRESS).
- An estimate of the variance is returned with each prediction.
- Intrinsically adaptive.

Future work

- Extension of the LL method to other local selection criteria (VC dimension, GCV).
- Classification applications.
- Integration with powerful software and hardware devices.
- From large to huge databases.
- New applications: bioinformatics, text mining, medical data, sensor networks, power systems.

References

[1] D. W. Aha. Editorial of special issue on lazy learning. *Artificial Intelligence Review*, 11(1–5):1–6, 1997.

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

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

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

[5] M. Birattari and G. Bontempi. The lazy learning toolbox, for use with matlab. Technical Report TR/IRIDIA/99-7, IRIDIA-ULB, Brussels, Belgium, 1999.

[6] M. Birattari, G. Bontempi, and H. Bersini. Lazy learning meets the recursive least-squares algorithm. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *NIPS 11*, pages 375–381, Cambridge, 1999. MIT Press.

[7] G. Bontempi. *Local Learning Techniques for Modeling, Prediction and Control*. PhD thesis, IRIDIA- Université Libre de Bruxelles, 1999.

[8] G. Bontempi, M. Birattari, and H. Bersini. Lazy learning for iterated time series prediction. In J. A. K. Suykens and J. Vandewalle, editors, *Proceedings of the International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling*, pages 62–68. Katholieke Universiteit Leuven, Belgium, 1998.

[9] G. Bontempi, M. Birattari, and H. Bersini. Recursive lazy learning for modeling and control. In *Machine Learning: ECML-98 (10th European Conference on Machine Learning)*, pages 292–303. Springer, 1998.

[10] G. Bontempi, M. Birattari, and H. Bersini. Lazy learners at work: the lazy learning toolbox. In *Proceeding of the 7th European Congress on Inteligent Techniques and Soft Computing EUFIT '99*, 1999.

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

[12] G. Bontempi, M. Birattari, and H. Bersini. Local learning for iterated time-series prediction. In I. Bratko and S. Dzeroski, editors, *Machine Learning: Proceedings of the Sixteenth International Conference*, pages 32–38, San Francisco, CA, 1999. Morgan Kaufmann Publishers.

[13] G. Bontempi, M. Birattari, and H. Bersini. A model selection approach for local learning. *Artificial Intelligence Communications*, 121(1), 2000.

[14] G. Bontempi, M. Birattari, and P.E. Meyer. Combining lazy learning, racing and subsampling for effective feature selection. In *Proceedings of the International Conference on Adaptive and Natural Computing Algorithms*. Springer Verlag, 2005. To appear.

[15] G. Bontempi, O. Caelen, S. Pierret, and C. Goffaux. On the use of supervised learning techniques to speed up the design of aeronautics components. *WSEAS Transactions on Systems*, 10(3):3098–3103, 2005.

[16] G. Bontempi and W. Kruijtzer. The use of intelligent data analysis techniques for system-level design: a software estimation

example. *Soft Computing*, 8(7):477–490, 2004.

[17] G. Bontempi and G. Lafruit. Enabling multimedia qos control with black-box modeling. In D. Bustard, W. Liu, and R. Sterritt, editors, *Soft-Ware 2002: Computing in an Imperfect World, Lecture Notes in Computer Science*, pages 46–59, 2002.

[18] G. Bontempi, A. Vaccaro, and D. Villacci. A semi-physical modelling architecture for dynamic assessment of power components loading capability. *IEE Proceedings of Generation Transmission and Distribution*, 151(4):533–542, 2004.

[19] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA, 1984.

[20] W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74:829–836, 1979.

[21] W. S. Cleveland and S. J. Devlin. Locally weighted regression: an approach to regression analysis by local fitting. *Journal of American Statistical Association*, 83:596–610, 1988.

[22] W. S. Cleveland and C. Loader. Smoothing by local regression: Principles and methods. *Computational Statistics*, 11, 1995.

[23] G. Corani. Air quality prediction in milan: feed-forward neural networks, pruned neural networks and lazy learning. *Ecological Modelling*, 2005. In press.

[24] T. Cover and P. Hart. Nearest neighbor pattern classification. *Proc. IEEE Trans. Inform. Theory*, pages 21–27, 1967.

[25] J. Fan and I. Gijbels. Adaptive order polynomial fitting: bandwidth robustification and bias reduction. *J. Comp. Graph. Statist.*, 4:213–227, 1995.

[26] J. Fan and I. Gijbels. *Local Polynomial Modelling and Its Applications*. Chapman and Hall, 1996.

[27] W. Hardle and J. S. Marron. Fast and simple scatterplot smoothing. *Comp. Statist. Data Anal.*, 20:1–17, 1995.

[28] R. Henderson. Note on graduation by adjusted average. *Transactions of the Actuarial Society of America*, 17:43–48, 1916.

[29] G. H. John and P. Langley. Static versus dynamic sampling for data mining. In *Proceedings of the Second International Con-*

ference on Knowledge Discovery in Databases and Data Mining. AAAI/MIT Press, 1996.

[30] M. C. Jones, J. S. Marron, and S. J. Sheather. A brief survey of bandwidth selection for density estimation. *Journal of American Statistical Association*, 90, 1995.

[31] V. Y. Katkovnik. Linear and nonlinear methods of nonparametric regression analysis. *Soviet Automatic Control*, 5:25–34, 1979.

[32] C. Loader. *Local Regression and Likelihood*. Springer, New York, 1999.

[33] C. R. Loader. Old faithful erupts: Bandwidth selection reviewed. Technical report, Bell-Labs, 1987.

[34] O. Maron and A. Moore. The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 11(1–5):193–225, 1997.

[35] T. M. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[36] R. Murray-Smith and T. A. Johansen. Local learning in local model networks. In R. Murray-Smith and T. A. Johansen, editors,

Multiple Model Approaches to Modeling and Control, chapter 7, pages 185–210. Taylor and Francis, 1997.

[37] R. H. Myers. *Classical and Modern Regression with Applications*. PWS-KENT Publishing Company, Boston, MA, second edition, 1994.

[38] B. U. Park and J. S. Marron. Comparison of data-driven bandwidth selectors. *Journal of American Statistical Association*, 85:66–72, 1990.

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

[40] J. Rice. Bandwidth choice for nonparametric regression. *The Annals of Statistics*, 12:1215–1230, 1984.

[41] D. Ruppert, S. J. Sheather, and M. P. Wand. An effective bandwidth selector for local least squares regression. *Journal of American Statistical Association*, 90:1257–1270, 1995.

[42] G. V. Schiaparelli. Sul modo di ricavare la vera espressione delle leggi della natura dalle curve empiricae. *Effemeridi Astronomiche di Milano per l'Arno*, 857:3–56, 1886.

[43] C. Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5:595–645, 1977.

[44] D. Villacci, G. Bontempi, A. Vaccaro, and M. Birattari. The role of learning methods in the dynamic assessment of power components loading capability. *IEEE Transactions on Industrial Electronics*, 52(1), 2005.

[45] G. Wahba and S. Wold. A completely automatic french curve: Fitting spline functions by cross-validation. *Communications in Statistics*, 4(1), 1975.

[46] D. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[47] M. Woodrofe. On choosing a delta-sequence. *Ann. Math. Statist.*, 41:1665–1671, 1970.