Residual Variance Estimation in Machine Learning

Elia Liitiäinen (corresponding author) [a], Michel Verleysen [b], Francesco Corona [a], Amaury Lendasse [a]

[a] *Department of Information and Computer Science, Helsinki University of Technology, P.O. Box 5400 Espoo, Finland, elia.liitiainen@hut.fi, +358-503279537 (mobile), +358-9-451 3277 (fax)*

[b] *Machine Learning Group, Université Catholique de Louvain, 3 Place du Levant, B-1348 Louvain-la-Neuve, Belgium*

Abstract

The problem of residual variance estimation consists of estimating the best possible generalization error obtainable by any model based on a finite sample of data. Even though it is a natural generalization of linear correlation, residual variance estimation in its general form has attracted relatively little attention in machine learning.

In this paper, we examine four different residual variance estimators and analyze their properties both theoretically and experimentally to understand better their applicability in machine learning problems. The theoretical treatment differs from previous work by being based on a general formulation of the problem covering also heteroscedastic noise in contrary to previous work, which concentrates on homoscedastic and additive noise.

In the second part of the paper, we demonstrate practical applications in input and model structure selection. The experimental results show that using residual variance estimators in these tasks gives good results often with a reduced computational complexity, while the nearest neighbor estimators are simple and easy to implement.

Key words: noise variance estimation, residual variance, model structure selection, input selection, nonparametric estimator, nearest neighbour

1 Introduction

Residual variance estimation and the related noise variance estimation problems are well-known in the field of statistics [3,24,13,20]. The problem consists of estimating the best possible generalization error obtainable by any model based on a finite sample of data. Thus it is a natural generalization of the Pearson correlation and as such an attractive measure of relevance due to its intuitive nature.

Despite the importance of the topic, it seems to be relatively unknown in machine learning. Moreover, many of the estimators derived in statistics fit poorly to high dimensional sparse data. Some references in machine learning include [5,14]; however, these works make the restrictive homoscedasticity assumption on the noise. This shortcoming has been addressed in [12], where a practical estimator with good convergence properties is derived.

The goal of this paper is two-fold. Firstly, we show how the residual variance can be estimated using simple and robust methods. We also analyze the convergence properties of the methods to understand better their weaknesses in real-world problems. The asymptotic consistency results are more general than previous theoretical results as the general heteroscedastic case is examined. Moreover, we analyze a locally linear estimator introduced in the statistics community [20], but not known in the field of machine learning.

The second goal is to show, how residual variance estimators can be used in applications. Here we demonstrate applications in input selection and model structure selection, both of which are important topics. The application in model structure selection has been investigated in [9,14] using two different estimators. There has also been research on input selection [18]. In this work the applications are combined with the theoretical results giving a wider prespective compared to the earlier works.

The outline of the paper is as follows. In Section 2 the residual variance estimation problem is introduced. In Section 3 some theoretical results for nearest neighbors are derived and in Section 4 estimators of the residual variance are derived and analyzed using the results in Section 3. To complement the analysis, experimental results are given in Section 5.

The application to model structure selection is introduced in Section 6 and the experimental analysis is in Section 7. Next the input selection problem is introduced in Section 8 and the corresponding experiments are in Section 9.

2 Residual Variance Estimation

Residual variance estimation means estimating the lowest possible expected mean squared error (MSE) in a given regression problem based on data. An abstract formulation of the problem is the goal of this section. Our approach corresponds to that in [5] with the distinction that the covariates do not have to be identically distributed.

2.1 Statement of the Problem

Consider the set of random variables $(Z_i)_{i=1}^{\infty} = (X_i, Y_i)_{i=1}^{\infty}$ taking values in $[0,1]^n \times [0,1]$. The assumption of boundeness is not essential; here it is made to simplify some proofs. In the regression problem the goal is to build a model that relates the variables $(X_i)_{i=1}^{M}$ to $(Y_i)_{i=1}^{M}$ with minimum possible error. In theory one would like to find the function g that minimizes the generalization error

$$L(g) = \frac{1}{M}\sum_{i=1}^{M} E[(Y_i - g(X_i))^2].$$

Here g is often parametrized for example as a linear or neural network model.

Let us assume that the variables $(X_i)_{i=1}^{M}$ possess the density functions (p_i). In this case the theoretically optimal solution is given by the following well-known theorem.

Theorem 1 *The functional L achieves its minimum for the function*

$$m(x) = \sum_{i=1}^{M} \frac{E[Y_i|X_i = x]p(X_i)}{\sum_{i=1}^{M} p(X_i)}.$$

Moreover, if

$$E[Y_i|X_i] = E[Y_j|X_j] \tag{1}$$

for all $i, j > 0$, then we may simplify

$$m(x) = E[Y_1|X_1 = x].$$

The residual variance estimation problem is the inverse of the regression problem: instead of trying to find the optimal model, we try to estimate the smallest achievable generalization error. In mathematical terms, the goal is to estimate V_M defined by

$$V_M = \inf_g L(g),$$

where the infimum is over square integrable functions. It turns out that estimating V_M is actually much easier than trying to reconstruct the function g based on a finite sample.

2.2 The Difference Between Homogenous and Heterogenous Noise

Even though residual variance estimation can be viewed as estimating the minimum of the cost function L, it can be also viewed in a slightly different but trivially equivalent way. By setting

$$r_i = Y_i - m(X_i)$$

we may always write

$$Y_i = m(X_i) + r_i,$$

that is, the output is generated by a model with additive noise. By definition, the residual variance is the (mean) variance of the noise:

$$V_M = \frac{1}{M} \sum_{i=1}^{M} E[r_i^2].$$

However, even though

$$E[r_i f(X_i)] = 0$$

for any bounded function f (see [17]), the noise variable r_i is in general not independent of X_i. Hence the variance function

$$\sigma(x) = \frac{1}{M} \sum_{i=1}^{M} \mathrm{Var}[r_i^2 | X_i = x]$$

is not necessarily constant. When σ does not depend on x, the noise is called homogenous, whereas the general case is referred to as heterogenous noise.

In practice, many tasks like variable selection require estimators that cope with heterogenous noise. In this paper we prove consistency in a general heteroscedastic case for the methods we discuss; however, strongly heterogenous noise may still affect the performance of some of the methods. It seems that this (at least theoretically) important point has been mostly neglected in previous work in machine learning including [5,14].

To avoid the use of technically complicated arguments, we assume from now on that the stationarity condition (1) holds with m and σ continuous functions on $[0, 1]^n$. Moreover, we assume that the residual variance σ is independent of M which is true when

$$\mathrm{Var}[Y_i | X_i] = \mathrm{Var}[Y_j | X_j]$$

for all $i, j > 0$. Thus we may drop the subscript M from V_M and simply write V for the residual variance.

3 On Nearest Neighbor Distances

The concept of nearest neighbor is well known in machine learning [5]. As the estimators of residual variance to be introduced in the next section are based on using k nearest neighbors, we discuss some relevant definitions and results.

The formal definition of the nearest neighbor index of the point X_i is

$$N[i,1] = \text{argmin}_{1\leq j\leq M}\|X_i - X_j\|$$

and the k-th nearest neighbor index is defined recursively as

$$N[i,k] = \text{argmin}_{1\leq j\leq M, j\notin\{i,N[i,1],\dots,N[i,k-1]\}}\|X_i - X_j\|,$$

that is, the closest point after removal of the preceeding neighbors. The corresponding distances are defined as

$$d_{i,k} = \|X_i - X_{N[i,k]}\|.$$

The next theorem derived in [11] bounds the average k-th nearest neighbor distance. Notice that the bound is suboptimal in the sense that rather rough approximations are used in the proof; however, it is sufficient for our purpose as it shows that the distances are on average at most of order $O(M^{-\alpha/n})$.

Theorem 2 *For* $0 < \alpha \leq n$*,*

$$\frac{1}{M}\sum_{i=1}^{M} d_{i,k}^{\alpha} \leq 3^{2\alpha}k^{2\alpha}n^{\alpha/2}M^{-\alpha/n}.$$

Proof The proof can be found in [11]. However, because there is a slight mistake in the constants, we repeat the proof here.

The proof starts by fixing a realization of the sample $(X_i)_{i=1}^M$ and a point $x \in [0,1]^n$. Suppose that x belongs to the open ball $B(X_j, d_{j,k})$ for some $0 < j \leq M$. Then, if we define the new sample $(\tilde{X}_i)_{i=1}^{M+1}$ as the union of x and $(X_i)_{i=1}^M$ with $\tilde{X}_{M+1} = x$, we know that in this new sample $x = \tilde{X}_{\tilde{N}[j,l]}$ for some $0 < l \leq k$, where the l-th nearest neighbor is taken in the augmented sample. However, for any choice of r, the number of elements in the set

$$I_{x,r} = \{0 < i \leq M : \tilde{X}_{\tilde{N}[i,r]} = x\} \tag{2}$$

is bounded by $3^n r$ ([5]). This, on the other hand, implies that the number of elements in the set

$$I_x = \{0 < i \leq M : \tilde{X}_{\tilde{N}[i,r]} = x, \text{ for some } 0 < r \leq k\} = \cup_{r=1}^{k} I_{x,r} \tag{3}$$

is bounded by (with the notation $|\cdot|$ for cardinality)

$$|I_x| \leq \sum_{r=1}^{k} |I_{x,r}| \leq \frac{1}{2}k(k+1)3^n \leq k^2 3^n. \tag{4}$$

Thus, if we pick a point x, it can belong to at most $k^2 3^n$ different k-th nearest neighbor balls $B(X_j, d_{j,k})$.

Let us define S_n as the volume of the unit ball. Denoting by $I_{B(x,r)}$ the indicator function of the ball $B(x,r)$ and observing that

$$\delta_{\alpha,k} = \frac{1}{M}\sum_{i=1}^{M} d_{i,k}$$

can be written as an integral, we have (using $d_{i,k} \leq \sqrt{n}$)

$$\begin{aligned} \delta_{n,k} &= \frac{S_n^{-1}}{M}\sum_{i=1}^{M}\int_{\Re^n} I_{B(X_i,d_{i,k})}(x)dx \\ &= \frac{S_n^{-1}}{M}\int_{B(0,2\sqrt{n})}\sum_{i=1}^{M} I_{B(X_i,d_{i,k})}(x)dx \leq \frac{3^{2n}n^{n/2}k^2}{M}. \end{aligned} \tag{5}$$

By Jensen's inequality [17] it can be shown that

$$\delta_{\alpha,k} \leq \delta_{n,k}^{\alpha/n}, \tag{6}$$

which implies that $\delta_{\alpha,k} \leq 3^{2\alpha} n^{\alpha/2} k^{2\alpha/n} M^{-\alpha/n}$ finishing the proof. □

It is of interest to ask, if the exponent α/n is optimal. As shown in [5], this is indeed the case when the intrinsic dimensionality of the data is n. However, for data lying in a low dimensional manifold, the nearest neighbor distances approach zero faster than the theorem would imply as shown in [8].

4 Estimators of Residual Variance

There exists certainly a wide variety of choices to estimate the residual variance. It is not our purpose to review all of these; instead we have chosen four different methods with different properties. These methods are simple and have relatively well understood properties; moreover, the Gamma test is a rather well established method. The goal is to provide a practical solution for applications in input and model structure selection.

4.1 The 1-NN Estimator

The first nearest neighbor estimator (also refered to as the Delta test) is based on the idea that similar inputs should produce outputs close to each other. The estimator can be written as [7]

$$\hat{V}_M^1 = \frac{1}{2M}\sum_{i=1}^{M}(Y_i - Y_{N[i,1]})^2.$$

The idea is that the approximation

$$Y_i - Y_{N[i,1]} \approx r_i - r_{N[i,1]} \tag{7}$$

is valid for M large enough. On the other hand, independence yields for any $M > 0$,

$$E[r_i r_{N[i,1]}] = 0; \tag{8}$$

and combining equations (7) and (8) leads to

$$E[\hat{V}_M^1] \approx \frac{1}{2}V + \frac{1}{2M}\sum_{i=1}^{M} E[r_{N[i,1]}^2].$$

Because $X_{N[i,1]}$ is close to X_i, it is seems likely that for M large enough

$$E[r_{N[i,1]}^2] \approx V.$$

Moreover, this certainly holds for homoscedastic noise as well as for a large class of distributions for the covariates. The following theorem formalizes the discussion.

Theorem 3 *The estimator $\hat{V}_M^1$ is asymptotically unbiased in the sense that*

$$|E[\hat{V}_M^1] - V| \to 0$$

as $M \to \infty$.

Proof By a straightforward algebraic manipulation, $\hat{V}_M^1$ may be represented as

$$E[\hat{V}_M^1] = I_1 + I_2 + I_3$$

with

$$
\begin{aligned}
I_1 &= \frac{1}{2M}\sum_{i=1}^{M} E[r_i^2] + \frac{1}{2M}\sum_{i=1}^{M} E[r_{N[i,1]}^2] = \frac{1}{2}V + \frac{1}{2M}\sum_{i=1}^{M} E[r_{N[i,1]}^2] \\
I_2 &= \frac{1}{M}\sum_{i=1}^{M} E[(r_i - r_{N[i,1]})(m(X_i) - m(X_{N[i,1]}))] \\
I_3 &= E[\frac{1}{2M}\sum_{i=1}^{M}(m(X_i) - m(X_{N[i,1]}))^2].
\end{aligned}
$$

By the basic properties of conditional expectations, we may conditionalize with respect to the sample $(X_i)_{i=1}^M$ to obtain (see [17])

$$
\begin{aligned}
&E[(r_i - r_{N[i,1]})(m(X_i) - m(X_{N[i,1]}))] \\
&= E[(r_i - r_{N[i,1]})E[(m(X_i) - m(X_{N[i,1]}))|X_1^M]] \\
&= E[(m(X_i) - m(X_{N[i,1]}))E[r_i - r_{N[i,1]}|X_1^M]] = 0
\end{aligned}
$$

and it follows that $I_2 = 0$. Let us show that I_3 converges to zero. By continuity, for any $\delta > 0$ there exists $\epsilon > 0$ such that

$$|m(x) - m(y)| < \delta$$

when $\|x - y\| < \epsilon$ and $x, y \in [0,1]^n$. Thus we have by Chebyshev's inequality

$$
\begin{aligned}
|I_3| &\leq \delta^2 + \frac{1}{M}\sum_{i=1}^{M} P(d_{i,1} > \epsilon) \\
&\leq \delta^2 + \frac{1}{\epsilon M}\sum_{i=1}^{M} d_{i,1}.
\end{aligned}
$$

Now an application of Theorem 2 shows that indeed $I_3 \to 0$ when $M \to \infty$.

To finish the proof, we must show that I_1 approaches V in the limit $M \to \infty$. Using again the basic properties of conditional expectations we get the result

$$E[r_{N[i,1]}^2] = \sum_{j=1}^{M} E[r_j^2 I(N[i,1] = j)]$$

with I denoting the indicator function. Conditionalization with respect to the sample $(X_i)_{i=1}^M$ yields

$$
\begin{aligned}
E[r_j^2 I(N[i,1] = j)] &= E[E[r_j^2|X_1^M]I(N[i,1] = j)] \\
&= E[I(N[i,1] = j)\sigma(X_j)].
\end{aligned}
$$

Thus to finish the proof we need to show that

$$\frac{1}{M}\sum_{i=1}^{M} E[\sigma(X_{N[i,1]})] \to \frac{1}{M}\sum_{i=1}^{M} E[\sigma(X_i)].$$

However, this follows exactly in the same way as the previous step of the proof because σ is bounded and continuous. □

In Theorem 3 we analyzed the bias of the algorithm; what about the variance? To understand the variance better, it should be observed that perturbing one element in the sample $(Z_i)_{i=1}^{M}$ has only a limited effect on $\hat{V}_M$. Without going into details, it follows straightforwardly from the classical Efron-Stein concentration inequality [22] that

$$\mathrm{Var}[\hat{V}_M^1] = O(M^{-1/2}).$$

See also [5] (Section 5) on L-dependent random variables, where another proof technique is used. In practice, the variance does not pose as much trouble as the bias as will be demonstrated by experiments. Notice that mean square convergence implies convergence in probability, but not almost sure convergence. It is possible to prove almost sure convergence, but it is outside the scope of this paper.

Another important question is the rate of convergence of the algorithm. For homoscedastic noise an answer is given by the following theorem. For the proof and concrete examples see [10].

Theorem 4 *Assume that m has bounded partial derivatives and that σ is a constant. Then*

$$\limsup_{M\to\infty} M^{\min\{2/n,1\}}|E[\hat{V}_M^1] - V| > 0.$$

The rate $O(M^{-2/n})$ is actually optimal as demonstrated in [10]. Thus it seems that convergence is fast for small n, say $n \leq 2$, whereas it slows down quickly when n grows. Also, strongly heteroscedastic noise may in theory weaken the performance. From the theoretical point of view the 1-NN estimator is not entirely satisfying in this sense as proving rates of convergence for heteroscedastic noise is difficult.

4.2 The Gamma Test

In Section 4.1 we observed that while the 1-NN estimator is a simple method, it suffers badly from the curse of dimensionality. One attempt to improve its properties is the Gamma test [5,7]. To derive the method, let us examine the quantities

$$\begin{aligned}\delta_{2,k} &= \frac{1}{M}\sum_{i=1}^{M} d_{i,k}^2\\ \gamma_k &= \frac{1}{2M}\sum_{i=1}^{M}(Y_i - Y_{N[i,k]})^2.\end{aligned}$$

Let us assume homoscedastic noise and the existence of a gradient ∇m. For M large enough to ensure the validity of the linear approximation to m and that γ_k and $E[\gamma_k]$ are close to each other, we have by similar logic as in Section 4.1,

$$\begin{aligned}\gamma_k &\approx V + \frac{1}{2M}\sum_{i=1}^{M}(m(X_i) - m(X_{N[i,k]}))^2\\ &\approx V + \frac{1}{2M}\sum_{i=1}^{M}((X_i - X_{N[i,k]})^T \nabla_{X_i} m)^2.\end{aligned}$$

Let us define

$$A(M,k) = \frac{\frac{1}{M}\sum_{i=1}^{M}((X_i - X_{N[i,k]})^T \nabla_{X_i} m)^2}{\delta_k}.$$

The Gamma test is based on the assumption that $A(M,k)$ approaches to a constant A independent of k when $M \to \infty$. While a rigorous theoretical analysis of this assumption has not been done, some convincing arguments can be found in [5]. Thus it seems reasonable to assume a linear relation between γ_k and δ_k when k varies between 1 and some small number l ($l = 10$ is used in this paper). This results in a regression problem

$$\text{argmin}_{a,b} \sum_{k=1}^{l}(\gamma_k - a - b\delta_{2,k})^2.$$

The Gamma test approximation to the noise variance is simply a. The following theorem gives an asymptotic consistency result for the Gamma test. Notice that the required conditions are more restrictive than those of the 1-NN estimator; this reflects the fact that as a more sophisticated method, it is not expected to be as robust. The proof is given in [11].

Theorem 5 *Choose $l > 0$. Then if*

$$\sup_{M>0} P\left(\sup_{0<k\leq l-1} \frac{\delta_{2,k+1}}{\delta_{2,k}} > L\right) \to 0 \qquad (9)$$

as $L \to \infty$, the Gamma test estimate $\hat{V}_M^2$ converges in probability to V.

Certainly it would be possible to prove almost sure convergence, but this would require some additional arguments which we do not consider relevant for this paper.

When is Condition (9) valid? The following proposition shows that for continuous data it is likely to hold in practice.

Theorem 6 *Assume that the points $(X_i)_{i=1}^M$ are i.i.d. on a convex compact set with a continuous and positive density possessing bounded partial derivatives. Then (9) holds.*

Proof The proof can be found in [5]. □

We may conclude that at least in the i.i.d. case the Gamma test is consistent. The rate of convergence is a more difficult topic; in [10] we conjectured that the bias is of order $O(M^{-3/n})$ without a theoretical proof. It seems likely that in mildly linear problems convergence is actually much faster.

4.3 A Locally Linear Estimator

Many practical problems are more or less linear; hence estimators performing well in linear problems seem attractice. In this section we analyze such a method introduced in [21]. It is shown that the estimator is very attractive especially for nearly homoscedastic noise.

Let us assume a homoscedastic model

$$Y_i = m(X_i) + r_i$$

and for each $i > 0$, introduce the weights $\omega_{i,1}, \ldots, \omega_{i,l}$, which are positive functions of the sample $(X_i)_{i=1}^M$. Here l should be chosen larger than n; the choice $l = n + 1$ is fine. For M large enough, we approximate

$$m(X_{N[i,k]}) - m(X_i) \approx (X_{N[i,k]} - X_i)^T \nabla_{X_i} m.$$

Now if

$$\sum_{k=1}^{l} \omega_{i,k} = 1, \tag{10}$$

then

$$
\begin{aligned}
E[(Y_i - \sum_{k=1}^{l} \omega_{i,k} Y_{N[i,k]})^2 | X_1^M] \approx\, & E[(\sum_{k=1}^{l} \omega_{i,k} (X_{N[i,k]} - X_i)^T \nabla_{X_i} m)^2 | X_1^M] \\
& + E[(r_i - \sum_{k=1}^{l} \omega_{i,k} r_{N[i,k]})^2 | X_1^M] \\
=\, & E[(\sum_{k=1}^{l} \omega_{i,k} (X_{N[i,k]} - X_i)^T \nabla_{X_i} m)^2 | X_1^M] \\
& + V(1 + \sum_{k=1}^{l} \omega_{i,k}^2).
\end{aligned}
$$

Notice the important role of homoscedasticity in the derivation of the last equality. Here we used basic properties of conditional expectations, which implies for example the following two equalities:

$$
E[\omega_{i,k} r_i r_{N[i,k]} | X_1^M] = 0 \tag{11}
$$
$$
E[\omega_{i,k}^2 r_{N[i,k]}^2] = V E[\omega_{i,k}^2 | X_1^M]. \tag{12}
$$

Now a natural restriction would be to require

$$
\sum_{k=1}^{l} \omega_{i,k} (X_{N[i,k]} - X_i) = 0, \tag{13}
$$

because then

$$
\hat{V}_M^3 = \frac{1}{M} \sum_{i=1}^{M} \frac{(Y_i - \sum_{k=1}^{l} \omega_{i,k} Y_{N[i,k]})^2}{1 + \sum_{k=1}^{l} \omega_{i,k}^2} \tag{14}
$$

would be unbiased except for the contribution of higher order terms.

For each i, (13) and (10) yield $n+1$ equations for k weights. As discussed in [21], under the condition that the covariates possess a density with respect to the Lebesgue measure, there exists a solution to the set of equations almost surely when $k \geq n+1$. Actually, if k is strictly larger than $n+1$, the solution is not unique. In this case one may choose the weights with the smallest Euclidean norm.

Clearly the estimate $\hat{V}_M^3$ shares similar asymptotic consistency properties as the 1-NN estimate. Moreover, it is unbiased in a class of linear problems; this is a very nice property when working with real world data. The following theorem summarizes these properties:

Theorem 7 *$\hat{V}_M^3$ is a consistent estimator in the sense that*

$$
|\hat{V}_M^3 - V| \to 0
$$

in probability as $M \to \infty$. Moreover, if the sample $(X_i, Y_i)_{i=1}^M$ is generated *from a linear model*

$$Y_i = w^T X_i + r_i$$

with homoscedastic noise, then

$$E[\hat{V}_M^3] = V.$$

Proof The proof is similar to that in section 4.1 and we give here only a short sketch. As before, it is possible to show that for any $\epsilon, k > 0$,

$$\frac{1}{M}\sum_{i=1}^{M} P(|Y_{N[i,k]} - Y_i - r_{N[i,k]} + r_i| > \epsilon) \to 0$$

as $M \to \infty$. Moreover, each term in equation (14) is bounded; consequently

$$E[\hat{V}_M^3] \to \frac{1}{M}\sum_{i=1}^{M} E[(\tilde{\omega}_{0,k} r_i - \sum_{k=1}^{l} \tilde{\omega}_{i,k} r_{N[i,k]})^2]. \tag{15}$$

as $M \to \infty$. Here, we defined

$$\tilde{\omega}_{i,k} = \frac{\omega_{i,k}}{\sqrt{1 + \sum_{i=1}^{k} \omega_{i,k}^2}}$$

and $\tilde{\omega}_{i,0} = (1 + \sum_{i=1}^{k} \omega_{i,k}^2)^{-1/2}$. Using formulas such as (11) and (12) we can represent the terms in the right side of equation (15) as

$$\begin{aligned}
&E[(\tilde{\omega}_{0,k} r_i - \sum_{k=1}^{l} \tilde{\omega}_{i,k} r_{N[i,k]})^2] \\
&= E[\tilde{\omega}_{0,k}^2 \sigma(X_i) + \sum_{k=1}^{l} \tilde{\omega}_{i,k}^2 \sigma(X_{N[i,k]})] \\
&= E[\sigma(X_i) + \sum_{k=1}^{l} \tilde{\omega}_{i,k}^2 (\sigma(X_{N[i,k]}) - \sigma(X_i))].
\end{aligned}$$

The integrand being bounded, it remains to show that

$$\frac{1}{M}\sum_{i=1}^{M} P(|\sigma(X_{N[i,k]}) - \sigma(X_i)| > \epsilon) \to 0$$

as $M \to \infty$. However, this is true by continuity and Theorem 2 using the Chebyshev's inequality again in the same way as in Section 4.1. Thus we have proven asymptotic unbiasedness and it remains to show the convergence of the variance to zero. This analysis is omitted here, see [5] on the variance of functions of nearest neighbors.

The second claim of the theorem follows straightforwardly by the discussion earlier in this section. □

For an extensive analysis of the speed of convergence under homoscedastic noise, see [21]. Intuitively it seems that the local linear estimator is never significantly worse than 1-NN while in most cases it tends to be much more accurate. However, again a good performance for heteroscedastic noise is not guaranteed even in linear problems.

4.4 The Modified 1-NN Estimator

The three estimators discussed so far share the problem of a weak theoretical background concerning heteroscedastic noise. While in practice this might not always be a problem, it is of interest to have a method with better properties in this sense. Here we discuss the method in [12] defined by the formula

$$\hat{V}_M^3 = \frac{1}{M}\sum_{i=1}^{M}(Y_i - \frac{1}{k}\sum_{l=1}^{k} Y_{N[i,2l]})(Y_i - \frac{1}{k}\sum_{l=1}^{k} Y_{N[i,2l+1]}).$$

In [12], the method was stated for $k = 1$ and we will adopt the same convention here. The idea behind the estimator is rather simple. Recall that we may always write

$$Y_i = m(X_i) + r_i.$$

As before, we approximate $m(X_{N[i,k]}) - m(X_i) \approx 0$ for $k = 1, 2$ to get

$$\frac{1}{M}\sum_{i=1}^{M} E[(Y_{N[i,2]} - Y_i)(Y_{N[i,1]} - Y_i)] \approx \frac{1}{M}\sum_{i=1}^{M} E[(r_{N[i,2]} - r_i)(r_{N[i,1]} - r_i)] = V.$$

The difference to the 1-NN estimator is that we did not require $E[r^2_{N[i,k]}] = E[r_i^2]$. This ensures that the variance function σ does not affect the rate of convergence as demonstrated by the following theorem. Again we do not discuss the variance of the estimator rigorously due to additional technical complications; however, it is intuitively clear that it is not much higher than that of the 1-NN estimator.

Theorem 8 *Assume that m is Lipschitz continuous with a Lipschitz constant $L > 0$. Then*

$$\limsup_{M\to\infty} M^{\min\{2/n,1\}}|E[\hat{V}_M^1] - V| > 0.$$

Proof By Lipschitz continuity,

$$|E[\hat{V}_M^1] - V| \leq \frac{1}{M}\sum_{i=1}^{M}|m(X_{N[i,1]}) - m(X_i)||m(X_{N[i,2]}) - m(X_i)| \leq \frac{L}{M}\sum_{i=1}^{M} d_{i,2}^2.$$

The proof is finished by an application of Theorem 2. □

Actually, it seems that convergence tends to be faster than that implied by the previous theorem. To see this, write

$$\begin{aligned} E[\hat{V}_M^1] - V &= \frac{1}{M}\sum_{i=1}^{M}(m(X_{N[i,1]}) - m(X_i))(m(X_{N[i,2]}) - m(X_i)) \\ &\approx \frac{1}{M}\sum_{i=1}^{M}\Delta_{i,1}\Delta_{i,2}d_{i,1}d_{i,2}, \end{aligned}$$

where we define

$$\Delta_{i,k} = \frac{(X_{N[i,k]} - X_i)^T \nabla_{X_i} f}{d_{i,k}}.$$

Thus $\Delta_{i,1}$ is simply proportional to the cosine of the angle between the gradient of m and the nearest neighbor vector. When M is large enough, this variable is expected to be approximately independent of the variables $\Delta_{i,2}$, $d_{i,1}$ and $d_{i,2}$ with an expectation value close to zero. While this claim is a rather strong one and requires analysis of nearest neighbor distributions, it will be verified experimentally. Remark that for the ordinary 1-NN estimator, the bias enters as $\Delta_{i,1}^2 d_{i,1}^2$ and thus the same discussion is not valid.

To summarize, it seems that the modified 1-NN estimator is equally simple and robust as the ordinary 1-NN residual variance estimator while it has some attractive theoretical properties. The practical value of the theoretical considerations will be verified by simulations.

4.5 Conclusion

As a conclusion, it can be stated that in terms of accuracy, the 1-NN estimator is inferior to the others. Compared to the modified 1-NN estimator, it does not seem to have additional advantages. The modified 1-NN estimator has rather nice theoretical properties especially for heteroscedastic noise, while it considerably simpler than the Gamma test. Thus of these three estimators, the use of the modified 1-NN estimator should be preferred in theoretical grounds.

The locally linear estimator has the advantage of being unbiased in linear problems. However, obtaining tight error bounds for this method seems tedious and thus a good performance is hard to guarantee theoretically. As such, the method is attractive in high dimensional problems, where other methods tend to fail.

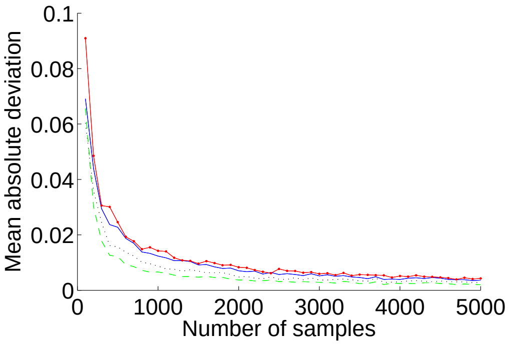

Fig. 1. Toy example 1: Residual variance estimation. The dashed line corresponds to the Gamma test, the dotted to the modified 1-NN, the solid to the 1-NN and the dashdotted to the local linear estimator.

5 Experiments on Residual Variance Estimators

The residual variance estimators are compared on two experiments. As a general preprocessing step, all the input data sets are set to unit variance and zero mean; the output variable was not preprocessed as it wouldnt affect the result. This preprocessing is used in all the experiments in this paper.

5.1 The Smoothed Parity Function

The first experiment is made with smoothed parity functions. The output Y is related to the covariate $(X^{(1)}, X^{(2)})$ by

$$Y = \sin(\pi X^{(1)}) \sin(\pi X^{(2)}) + r$$

with $r \sim N(0, 0.1)$ and $X \sim N(0, I)$. The results are illustrated Figure 1; as a measure of performance, the mean absolute deviation of the estimated noise variance from the true value is calculated. All the estimators are approximately equal in this example. It can be seen that the estimators have rather similar convergence properties, but the Gamma test gives the best results. The modified 1-NN estimator is slightly worse, but not significantly. Slightly suprisingly the local linear estimator is the worst in this example.

5.2 A Linear Combination of Smoothed Parity Functions

The second experiment is made with a linear combination of smoothed parity functions. The output Y is related to the covariate X by

$$Y = \frac{1}{2} \sin(\pi X^{(1)}) \sin(\pi X^{(2)}) + \frac{1}{2} \sin(\pi X^{(3)}) \sin(\pi X^{(4)}) + r,$$

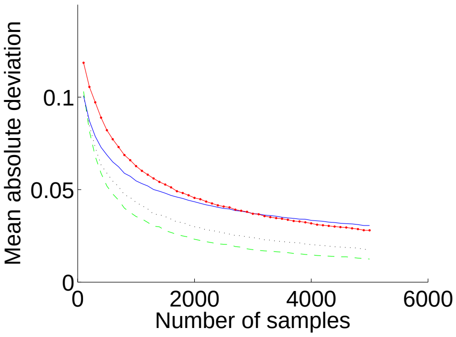

Fig. 2. Toy example 2: Residual variance estimation. The dashed line corresponds to the Gamma test, the dotted to the modified 1-NN, the solid to the 1-NN and the dashdotted to the local linear estimator.

with $r \sim N(0, 0.1)$ and $X \sim N(0, I)$. The results are illustrated in Figure 2. In this example we have some differences between the methods. The Gamma test is the best, but the modified 1-NN estimator has rather similar convergence properties. It is slightly suprising that the local linear estimator is worse than 1-NN even for quite large number of samples. Thus it seems that this estimator has some difficulties in nonlinear problems unless the number of samples is high.

6 Choosing Structural Parameters for a Learning Machine

In this section we demonstrate an important application of residual variance estimators: model structure selection. For example in multilayer perceptron networks (MLP) this means choosing the number of neurons in the hidden layers, whereas an alternative is to use a complex model with regularization. Another example is Least-Squares Support Vector Machines (LS-SVM), where two hyperparameters must be chosen. The idea is to set a target (the residual variance) that the training error should reach thus avoiding the need of leave-one-out (LOO) or alternative estimators of the generalization error.

6.1 MLP

The MLP is a widely used model, which has the uniform approximation capability. Here we address the problem of choosing the number of neurons to obtain the least possible generalization error.

By training a MLP for a number of neurons increasing from 1 to L for some $L > 0$, a set of models with increasing complexity is obtained. In principle, if local minima are avoided, the training error is a decreasing function of the number of neurons approaching to zero as the model complexity grows. However, of course in practice this is not always true as local minima may be difficult to avoid.

The problem of overfitting occurs when the number of neurons is too high. This means that the model becomes overly complex with a poor generalization performance. Recall now that the residual variance is the best possible generalization error obtainable; this means that a model with a training error higher than it tends to underfit, whereas in the opposite case, overfitting is likely. Thus we would like the training error be close to the residual variance motivating the following algorithm:

1. Choose $L > 0$ and train a MLP for the number of neurons $1 \dots L$.
2. Calculate an estimate of the residual variance.
3. Of the resulting models, choose the one that has the least number of neurons among those networks that have a training error below the estimated residual variance.

The advantage of the proposed method is that it is faster than the traditional k-fold cross-validation method by the factor k. Moreover, it is less prone to local minima as the performance goal is always met.

6.2 LS-SVM

The least-squares support vector machine (LS-SVM) is a well-known modification of the common support vector machine. By using the least-squares cost function, an analytic global solution in the training phase is obtained for fixed hyperparameter values. The LS-SVM model is defined in the primal space by

$$Y = \omega^T \phi(X) + b,$$

where the fixed function ϕ maps the input vector X into a high dimensional space. The idea is to find the free parameters ω and b as the solution of

$$\text{argmin}_{\omega,b,e} \frac{1}{2}\omega^T\omega + \frac{1}{2}\gamma \sum_{i=1}^{M} e_i^2 \text{ subject to } Y_i = \omega^T \phi(X_i) + b + e_i\text{,i=1,}\dots\text{,M.}$$

Because ϕ cannot be computed explicitly (as it is a mapping to a high or infinite dimensional space), this optimization problem is solved in the dual space leading to a solution of the form

$$\hat{m}(x) = \sum_{i=1}^{M} \alpha_i K(x, X_i) + b.$$

The kernel function K is defined by ϕ; any kernel satisfying the Mercer's condition can be chosen. In the experiments we will use the Gaussian kernel given by

$$K(x,y) = e^{-\frac{\|x-y\|^2}{\sigma^2}}.$$

While the weights $(\alpha_i)_{i=1}^{M}$ can be solved analytically, the hyperparameters (γ, σ) are rather difficult to find. A commonly used method is grid search (implemented in the LS-SVM toolbox [2]), which, however, is time consuming. An alternative, faster, method based on the use of noise variance estimators, is discussed here (see also [15]).

The selection of the pair (γ, σ) is a two-dimensional optimization problem: for each pair, there is the corresponding cross-validation error (we will use 10-fold CV). The idea is to convert it into a one-dimensional problem by selecting $\gamma(\sigma)$ in such a way that the training error is the same as the estimate of the residual variance. To see that this is possible, observe that for each fixed kernel bandwith σ, the training error is a continuous decreasing function of γ approaching to zero as $\gamma \to 0$ and the variance of the output as $\gamma \to \infty$. Thus, for some $\gamma > 0$ the training error must be the same as the residual variance. The resulting one-dimensional problem is solved by grid search, which is now easier than the search in a two-dimensional space.

7 Experiments on Model Structure Selection

The experimental section on model structure selection consists of simulations on three different data sets.

7.1 Toy Example

The first experiment is made using data generated by the model

$$Y = \sin(2\pi X^{(1)}) \sin(2\pi X^{(2)}) + r$$

with the residual r following the normal distribution and X uniform on $[0,1]^2$. The number of points in the training set is chosen as 1000 and for the test set, 10000 realizations are generated. To compare different models, the mean squared error is calculated. The variance of the residual is 0.25.

As a first step, results using 10-fold crossvalidation (CV) to choose the model structure for the MLP and LS-SVM were calculated. For the MLP this means choosing the number of neurons to minimize the CV error and correspondingly for the LS-SVM the hyperparameters (regularization, kernel bandwith)

MLP (10-fold CV)	0.278 (5)
LS-SVM (10-fold CV)	0.270

Table 1
Toy example: test errors after model structure selection using cross-validation. The number of neurons is in parentheses.

	1-NN	Gam.	LL	Mod. 1-NN
MLP	0.278 (5)	0.278 (5)	0.278 (5)	0.278 (5)
LS-SVM	0.276	0.282	0.282	0.279

Table 2
Toy example: test errors of the model structure selection using residual variance estimators. The number of neurons is in parentheses.

	1-NN	Gam.	LL	Mod. 1-NN
estimate	0.26	0.27	0.27	0.26

Table 3
Toy example: the estimated noise variances

are chosen. The test errors are reported in Table 1. In this problem they are very close to the residual variance. To optimize the MLP, 50 different initial conditions are used for each training together with Levenberg-Marquardt optimization to avoid local minima as much as possible. Of the resulting MLPs, the one giving the lowest training error is chosen. The number of neurons varies from 1 to 20. To optimize the LS-SVM, the gridsearch method implemented in [2] is used.

Next results for the model structure selection method using residual variance estimators are calculated. The training of the MLP is implemented in a similar way as described before with the number of neurons varying between 1 and 20. Each estimator is used to pick up one model from the 20 different models; thus the test errors are the same when the same number of neurons is picked. As explained in Section 6.2, the optimization problem of finding the hyperparameters of the LS-SVM is reduced to a one dimensional problem, which can be solved rather easily. Here, gridsearch is used to solve the resulting nonlinear optimization problem. The results of this experiment are in Table 2.

We can see that the normalized mean squared test errors are essentially the same in Tables 1 and 2; thus in this example the gain in computational speed does not come at the cost of decreased performance. Table 3 shows that the estimated noise variances are rather close to the true value.

MLP (10-fold CV)	0.014 (18)
LS-SVM (10-fold CV)	0.00013

Table 4
Stereopsis: model structure selection using cross-validation. The reported values are the MSEs on the test set. The selected number of neurons is in parentheses.

	1-NN	Gam.	LL	Mod. 1-NN
MLP	0.0013 (1)	0.0013 (1)	0.0013 (1)	0.0013 (1)
LS-SVM	0.088	0.014	0.0020	0.0064

Table 5
Stereopsis: model structure selection using residual variance estimators (NMSE on the test set). The selected number of neurons is in parentheses.

	1-NN	Gam.	LL	Mod. 1-NN
estimate	0.10	0.015	0.0017	0.0063

Table 6
Toy example: the estimated value of the noise-to-signal ratio Var$[r]$/Var$[Y]$.

7.2 Stereopsis

The stereopsis data set is a well-known benchmark originating from the Evaluating Predictive Uncertainty Challenge organized by the PASCAL network of excellence [1]. It consists of 192 samples for training and 300 for testing. The number of variables is four.

The results in Table 4 correspond to the training using 10-fold cross-validation and the corresponding results for the residual variance based training methods are in Table 5. The training is performed in a similar fashion as in Section 7.1.

In this example we see some decrease in performance depending on the chosen estimator of residual variance. Clearly the local linear estimator gives the best results, whereas the simple 1-NN estimator is not accurate enough. Moreover, the training based on the use of residual variance estimators resulted in less complicated models for the MLP. Table 6 gives more interesting details about the reason behind the obtained values: the local linear estimator gives much smaller estimates than the other ones. This is because it tends to be less unbiased in problems with a small amount of samples.

MLP (10-fold CV)	0.33 (11)
LS-SVM (10-fold CV)	0.24

Table 7
Boston: model structure selection using cross-validation (NMSE on the test set). The selected number of neurons is in parentheses.

	1-NN	Gam.	LL	Mod. 1-NN
MLP	0.088 (2)	0.088 (2)	0.076 (3)	0.088 (2)
LS-SVM	0.17	0.18	0.13	0.17

Table 8
Boston: model structure selection using residual variance estimators. The selected number of neurons is in parentheses.

	1-NN	Gam.	LL	Mod. 1-NN
estimate	0.16	0.17	0.088	0.15

Table 9
Toy example: the estimated value of the noise-to-signal ratio Var$[r]$/Var$[Y]$.

7.3 Boston Housing Data

The Boston housing data set is another well-known benchmark. The data consists of 506 points, the output describing the median value of homes in American towns in USD and the input consisting of a set of attributes [6]; here the original data set is permuted randomly. As a first step we trained the MLP and LS-SVM using 10-fold CV; the results are shown in Table 7. It can be seen that the obtained accuracy is not high. The results with residual variance estimators are shown in Table 8. Again, a similar procedure as in the previous section was made to try to avoid local minima for the MLP. Surprisingly the NMSE values are much lower than in Table 7. Again the local linear estimator seems to be a good choice. Thus in this example not only we have a speed-up in computation, but also the obtained models are much more accurate.

The results are comparable to other results like those in [23]. One must note that here we did not perform input selection; in general input selection helps to avoid the curse of dimensionality.

8 Input Selection

To define the problem of input selection, let us assume that the data is generated by the model

$$Y = m(X) + r = m(X^{(1)}, \dots, X^{(l)}) + r.$$

In practical modelling tasks, the number of variables in $X = (X^{(1)}, \dots, X^{(l)})$ may be high bringing problems with the curse of dimensionality and computational complexity. For example, especially distance-based methods suffer from a fast decrease in performance when l grows as demonstrated in [10]. Thus it is of importance to find a small subset of inputs, $(X^{(j_1)}, \dots, X^{(j_k)})$ that would still allow us to model Y.

To perform input selection, one needs a measure of relevance to evaluate different combinations of inputs. Here we propose the use of the residual variance as such a measure. At first sight a good idea would seem to be to evaluate a large amount of subsets of inputs and then take the one that minimizes the residual variance; however, this approach contains one important flaw: the combination that minimizes the residual variance is taking all the variables.

However, in practice all the estimators are biased. So one way to avoid the monotonicity is to use an estimator that has an exploitable bias. In [4] it has been proven that the 1-NN estimator is such an estimator: adding too many variables tends to increase the estimated residual variance. In a simple case this can be stated as follows:

Theorem 9 *Assume that the sample $(X_i)_{i=1}^M$ is generated by the linear model*

$$Y = w^T X + b + \epsilon$$

with ϵ independent noise and X a uniform random variable. Let I be the set of indices

$$I = \{i : w^{(i)} \neq 0\}$$

and let $\hat{I}_M$ be the subset that minimizes the 1-NN estimator. Then

$$P(\hat{I}_M = I) \to 1$$

as $M \to \infty$.

Moreover, it seems that this theorem can be generalized to the nonlinear case, even though this has not yet been proven theoretically. The idea is that, while the estimator converges to the true value, choosing too many variables tends to give a too large value. Thus we may conclude that the 1-NN estimator can be used to select inputs.

Var[ϵ]	1/600	1/200	3/200
1-NN	100 (100)	100 (100)	86 (100)
mod 1-NN	40 (90)	17 (88)	8 (92)
Gamma	0 (100)	0 (100)	0 (100)
LL	21 (100)	8 (100)	3 (100)

Table 10
Toy example: the percentage of the correct input combination. In parentheses, the number of runs, where the correct inputs were a subset of the selected variables.

Is it possible to use some of the other estimators in a similar way? The answer to this question is negative as will be seen in the experimental section.

9 Experiments on Input Selection

In this section, input selection using residual variance estimators is demonstrated through three experiments on different datasets.

9.1 Toy Example

We generated data from the model

$$Y = X^{(1)}X^{(2)} + \sin X^{(3)} + \epsilon$$

with Gaussian noise ϵ and the covariate X an 8 dimensional random vector on the unit cube. To perform the input selection, all combinations of inputs are tried and the one minimizing the estimator is selected. A good input selection method should be able to pick up the first three ones with a high probability.

The results of the experiment are shown in Table 10. The number of samples is 1000 and the variance of the noise is varied (the values 1/600,1/200 and 3/200 were tried). Because the number of variables is low, we were able to test all possible input combinations with each residual variance estimator.

The superiority of the 1-NN based method is clear as expected; the rest of the methods are not able to solve this simple toy example. The conclusion is that there is only one method that can be used for input selection. The performance of input selection seems dependent on the level of noise as expected, but the 1-NN method is rather robust. However, it is interesting to notice that the Gamma test and the local linear estimator are always able to choose the right combination, but tend to pick up too many variables.

LS-SVM (10-fold CV)	0.069

Table 11
Tecator: results without input selection (NMSE on the test set).

MLP (10-fold CV)	0.013
LS-SVM (10-fold CV)	0.040

Table 12
Tecator: results with input selection (NMSE on the test set). The selected inputs are 951 nm, 952 nm, 984 nm, 990 nm and 991 nm.

9.2 Tecator

The tecator data set is well-known in the field of chemometry [16]. The input consists of a set of continuous spectra discretized at the frequency interval 950..1050 nm, the dimension of the discretized space being 100. The output is the fat content of meat samples.

As a first step, the LS-SVM is trained on the whole data set without input selection. The MLP is not trained, because without regularization it overfits. The obtained result is in Table 11. Next the input selection procedure was performed using the 1-NN estimator. To find the minimizing subset of variables, 10 different initial conditions were tried one being the empty set and the rest random subsets. As the search algorithm, we used forward-backward selection (see for example [19]), which proceeds at each iteration by removing or adding a variable in such a way that the cost function decreases as much as possible. Such a stepwise method is prone to local minima; thus trying many initial conditions is recommended.

The selected frequencies are 951 nm, 952 nm, 984 nm, 990 nm and 991 nm. In Table 13 we have trained models with these inputs using crossvalidation. Compared to the results in [16], the results of the LS-SVM are worse. This may be due to the grid search method used in the optimization phase which may find suboptimal solutions. Also, in [16], a spline compression method is used to reduce the dimensionality of the input space prior to input selection. However, especially the result of the MLP reveals that the select input variables indeed contain relevant information of the fat content as the prediction accuracy is rather good.

9.3 Computer Activity Data

The computer activity data consists of a collection of computer systems activity measures obtained from a Sun Sparcstation 20/712 with 128 Megabytes of memory. The data consists of 22 attributes explained in Table 13. The task is to predict the portion of time that cpus run in user mode. As many real world data sets, the computer activity measurements are temporally correlated making the regression task more difficult.

As a first step, we trained the LS-SVM and MLP without input selection with the results in Table 14. Again model selection was made using 10-fold cross-validation. Secondly, we added variable selection using the 1-NN estimator; the results are shown in Table 13. We can see that 12 inputs were chosen. The test results using these variables together with 10-fold CV are found in Table 15.

It can be seen that the performance of the LS-SVM was improved significantly, whereas the MLP is equally good in both cases. Thus we may conclude that in this experiment, we were able to reduce model complexity without compromising the prediction accuracy.

10 Conclusion

In this paper, we discussed the concept of residual variance. Efficient methods for estimating it were surveyed and theoretical results were given. The theoretical analysis led to conclusions that were supported by experimental analysis; the 1-NN estimator is the least accurate, whereas the modified 1-NN and Gamma test estimators are the best in terms of accuracy.

Two important applications of residual variance estimators in machine learning were presented, model struture selection and input selection. For model structure selection, it is essential to use an accurate estimator such as the modified 1-NN estimator or the Gamma test. The experimental results show, that using residual variance estimation to choose the structural parameters of a model lead to good models while at the same time the computational complexity is reduced compared to minimizing a cross-validation error.

When doing input selection with residual variance estimators, we recommend using the 1-NN estimator. Experimental results reveal that the monotonicity of residual variance leads to bad solutions even in simple problems when other estimators are used. A recent theoretical result [4] reveals that the 1-NN estimator does not suffer from the same problem as the other estimators and this conclusion is supported by our experiments, where good input combinations were consistenly selected. We obtained significant reduction in model complexity while at the same time, prediction accuracy was not compromised.

As a topic of future research, we find the case of sparse high-dimensional data important. However, based on the experiments we may conclude that

attribute	short description
1*	Reads (transfers per second) between system memory and user memory
2	writes (transfers per second) between system memory and user memory
3*	Number of system calls of all types per second
4	Number of system read calls per second
5*	Number of system write calls per second
6*	Number of system fork calls per second
7*	Number of system exec calls per second
8*	Number of characters transferred per second by system read calls
9*	Number of characters transferred per second by system write calls
10	Number of page out requests per second
11	Number of pages, paged out per second
12	Number of pages per second placed on the free list
13	Number of pages checked if they can be freed per second
14	Number of page attaches per second
15*	Number of page-in requests per second
16	Number of pages paged in per second
17*	Number of page faults caused by protection errors (copy-on-writes)
18*	Number of page faults caused by address translation
19	Process run queue size
20*	Number of memory pages available to user processes
21*	Number of disk blocks available for page swapping
22	Portion of time that cpus run in user mode

Table 13
Computer activity data: the features. The selected inputs by the 1-NN estimator are marked by (*). Notice that feature 22 corresponds to the output to be predicted.

LS-SVM (10-fold CV)	0.18
MLP (10-fold CV)	0.028

Table 14
Computer activity data: results without input selection (NMSE on the test set).

LS-SVM (10-fold CV)	0.064
MLP (10-fold CV)	0.025

Table 15
Computer activity data: results with input selection (NMSE on the test set).

the modified 1-NN and Gamma test estimators perform rather well even in this case.

References

[1] predict.kyb.tuebingen.mpg.de/pages/home.php.

[2] www.esat.kuleuven.ac.be/sista/lssvmlab.

[3] L. Devroye, L. Györfi, D. Schäfer, The estimation problem of minimum mean squared error, Statistics and Decisions 21 (2003) 15–28.

[4] E. Eirola, E. Liitiäinen, A. Lendasse, F. Corona, M. Verleysen, Using the delta test for variable selection, in: ESANN 2008, European Symposium on Artificial Neural Networks, Bruges (Belgium), 2008.

[5] D. Evans, Data-derived estimates of noise for unknown smooth models using near-neighbour asymptotics, Ph.D. thesis, Cardiff University (2002).

[6] D. Harrison, D. L. Rubinfeld, Hedonic prices and the demand for clean air, Journal of Environmental Economics & Management 5 (1978) 81–102.

[7] A. J. Jones, New tools in non-linear modelling and prediction, Computational Management Science 1 (2004) 109–149.

[8] S. R. Kulkarni, S. E. Posner, Rates of convergence of nearest neighbor estimation under arbitrary sampling, IEEE Transactions on Information Theory 41 (4) (1995) 1029–1039.

[9] A. Lendasse, Y. Ji, N. Reyhani, M. Verleysen, LS-SVM hyperparameter selection with a nonparametric noise estimator, in: E. O. S. Z. W. Duch, J. Kacprzyk (ed.), ICANN05, International Conference on Artificial Neural Networks, Artificial Neural Networks: Formal Models and Their Applications, vol. 3697 of Lecture Notes in Computer Science, 2005, pp. 625–630.

[10] E. Liitiäinen, F. Corona, A. Lendasse, Nearest neighbor distributions and noise variance estimation, in: ESANN 2007, European Symposium on Artificial Neural Networks, Bruges (Belgium), 2007, pp. 67–72.

[11] E. Liitiäinen, F. Corona, A. Lendasse, Non-parametric residual variance estimation in supervised learning, in: IWANN 2007, International Work-Conference on Artificial Neural Networks, San Sebastian (Spain), Lecture Notes in Computer Science, Springer-Verlag, 2007, pp. 63–71.

[12] E. Liitiäinen, F. Corona, A. Lendasse, On nonparametric residual variance estimation, Neural Processing Letters (preprint).

[13] U. Müller, A. Schik, W. Wefelmeyer, Estimating the error variance in nonparametric regression by a covariate-matched U-statistic, Statistics 37 (3) (2003) 179–188.

[14] K. Pelckmans, J. D. Brabanter, J. Suykens, B. D. Moor, Variogram based noise variance estimation and its use in kernel based regression, in: Proc. of the IEEE Workshop on Neural Networks for Signal Processing, 2003, pp. 199–208.

[15] K. Pelckmans, J. D. Brabanter, J. Suykens, B. D. Moor, The differogram: Nonparametric noise variance estimation and its use for model selection, Neurocomputing 69 (1-3) (2004) 100–122.

[16] F. Rossi, D. Francois, V. Wertz, M. Meurens, M. Verleysen, Fast selection of spectral variables with b-spline compression, Chemometrics and Intelligent Laboratory Systems 86 (2) (2007) 208–218.

[17] A. N. Shiryaev, Probability, Springer, 1995.

[18] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, A. Lendasse, Methodology for long-term prediction of time series, Neurocomputing 70 (16-18) (2007) 2861–2869.

[19] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, A. Lendasse, Methodology for long-term prediction of time series, Neurocomputing 70 (2007) 2861–2869.

[20] V. Spokoiny, Variance estimation for high-dimensional regression models, Journal of Multivariate Analysis 10 (4) (2002) 465–497.

[21] V. Spokoiny, Variance estimation for high-dimensional regression models, Journal of Multivariate Analysis 10 (4) (2002) 465–497.

[22] J. M. Steele, An Efron-Stein inequality for nonsymmetric statistics, Annals of Statistics 14 (59) (1986) 753–758.

[23] J. Tikka, Input selection for radial basis function networks by constrained optimization, in: Proceedings of the 17th International Conference on Artificial Neural Networks (ICANN 2007), Lecture Notes in Computer Science, Springer-Verlag, 2007, pp. 239–248.

[24] T. Tong, W. Yuedong, Estimating residual variance in nonparametric regression using least squares, Biometrika 92 (4) (2005) 821–830.