Empirical comparison of nonparametric regression estimates on real data *

Daniel Jones[1], Michael Kohler[1], Adam Krzyżak[2,†]
and Alexander Richter[3]

[1] *Fachbereich Mathematik, Technische Universität Darmstadt, Schlossgartenstr. 7, 64289 Darmstadt, Germany, email: jones@mathematik.tu-darmstadt.de, kohler@mathematik.tu-darmstadt.de*

[2] *Department of Computer Science and Software Engineering, Concordia University, 1455 De Maisonneuve Blvd. West, Montreal, Quebec, Canada H3G 1M8, email: krzyzak@cs.concordia.ca*

[3] *Hessisches Statistisches Landesamt, Rheinstraße 35/37, 65175 Wiesbaden, Germany email: arichter@statistik-hessen.de*

August 13, 2012

Abstract

The performance of nine different nonparametric regression estimates is empirically compared on ten different real data sets. The number of data points in the real data sets varies between 7900 and 18000, where each real data set contains between 5 and 20 variables. The nonparametric regression estimates include kernel, partitioning, nearest neighbor, additive spline, neural network, penalized smoothing splines, local linear kernel, regression trees and random forests estimates. The main result is a table containing the empirical L_2 risks of all nine nonparametric regression estimates on the evaluation part of the different data sets. The neural networks and random forests are the two estimates performing best. The data sets are publicly available, so that any new regression estimate can be easily compared with all nine estimates considered in this paper by just applying it to the publicly available data and by computing its empirical L_2 risks on the evaluation part of the data sets.

AMS classification: Primary 62G08, secondary 62P99.

Key words and phrases: Nonparametric regression, L_2 error, real data performance.

1 Introduction

The evaluation of the finite sample size performance of a new nonparametric regression estimate is always difficult. Not only does it involve implementation and application to

*Running title: *Nonparametric regression on real data*

†Corresponding author: Tel. +1 514 848 2424, ext. 3007, Fax. +1 514 848 2830

simulated or real data of the newly proposed estimate, but it also requires that the same is done correctly for other standard estimates. Furthermore in case of simulated data it is never obvious how it should be generated in order to be similar to data occurring in applications. Using real data for nonparametric regression estimates is usually not an alternative since most real data sets with a reasonably large sample size (which is necessary for any application of multivariate nonparametric estimates) are not publicly available: Datasets of real data can be obtained for example from the Statistical Software Information - University of Massachusetts Amherst (2004) or DASL (1996). Here most of the times the ratio of sample size to number of covariates is not sufficiant for nonparametric regression estimates (e.g., 418/20, 203/16, 102/12 etc.). More appropriate data sets in this respect can be found at the UCI Machine Learning Repository (Frank and Asuncion (2010)). Unfortunately most of them are for classification purposes.

The purpose of this article is to present ten newly generated and publicly available data sets with sample size varying between 7900 and 18000. Each data set consists of a dependent variable which has to be predicted using between 4 and 19 covariates. All data sets are based on real data collected by the Hessian Statistical Office and the Federal Statistical Office of Germany which has been anonymized by the Research Data Center of the Hessian Statistical Office. So all data sets are real data sets occurring in practice but modified slightly such that they can be published and used without any restrictions.

The data sets are used to empirically compare nine different nonparametric regression estimates. For this purpose each estimation algorithm is implemented in MATLAB®. We consider standard local averaging estimates such as kernel, partitioning and nearest neighbor estimates, local linear kernel estimates, smoothing splines, least squares estimates using neural networks and additive B-splines, regression trees and random forests. Each time the smoothing parameter of the regression estimate is determined by cross-validation involving splitting of the sample. Each of ten data sets is divided into a learning and testing set containing two thirds of the data points and an evaluation set containing the remaining data points. The estimates are applied to the learning and testing data and the empirical L_2 risks are computed on the evaluation sets. The performance of the estimates is judged by the resulting empirical L_2 risks on the evaluation sets.

The ten data sets are described in detail in Section 2, the nonparametric regression estimates are described in Section 3 and the main result consisting of a table of all occurring empirical L_2 risks on the evaluation sets is presented and discussed in Section 4.

2 Ten data sets

Our data sets come from different application areas. The applications are health insurance, agriculture, birth weight, value added tax, building of houses, housing benefit, old age beneficiaries, car accidents, income and university exams. In the next ten subsections we describe each data set in detail.

2.1 Health insurance costs (health). This data set consists of 9413 data points of dimension 5. The dependent variable is the amount of money in Euro that health insur-

ance spent in a year for the medical treatment of a member. The member is described by the covariates age (in years), sex (which is coded as 0 and 1), western part of Germany or not (0/1) and the number of non-stationary treatments in the year.

2.2 Agriculture data (agric.). This data set consists of 9022 data points of dimension 9. The dependent variable is the income of a farm in a year. The farm is described by the number of cows (in animal units, nonnegative real number), number of pigs (in animal units, nonnegative real number), number of horses, sheep and fowl (in animal units, nonnegative real number), agriculture area used for producing winter wheat and oat (nonnegative real number), agriculture area used for producing barley (nonnegative real number), agriculture area used for producing other plants (nonnegative real number), fallow ground area (nonnegative real number) and the area used for agriculture (nonnegative real number).

2.3 Birth weight data (birth). This data set consists of 14645 data points of dimension 7. The dependent variable is the birth weight of the baby. The baby is described by its sex (0/1), whether the mother gave simultaneously birth to several children or not (0/1), whether it is the first child of the mother (0/1), age of the mother (in years), age of the father (in years), duration of the marriage (in months, 0 if the parents are not married).

2.4 Value added tax data (tax). This data set consists of 8100 data points of dimension 7. The dependent variable is the value added tax which had to be paid in a year by a company. The independent variables correspond to how much was produced in the year before (nonnegative real number), how much value added tax had to be paid in advance in the year before (nonnegative real number), how much value added tax has to be paid in advance in the current year (nonnegative real number), how much goods where produced where the tax rate was 19% (nonnegative real number), volume of sales where tax has to be paid (nonnegative real number) and tax reduction on value added tax in the current year (nonnegative real number).

2.5 Building project data (build). This data set consists of 11276 data points of dimension 10. The dependent variable is the estimated cost of the building project (in 1000 Euro). The independent variables are whether the building project is organized by the public (0/1), a private person (0/1) or neither (0/1), whether it is a building for living or not (0/1), number of floors (natural number), effective area after finishing the building project (nonnegative real number), effective area before starting the building project (nonnegative real number), living area after finishing the building project (nonnegative real number) and living area before starting the building project (nonnegative real number).

2.6 Housing benefit data (liv.). This data set consists of 12395 data points of dimension 7. The dependent variable is the amount of housing benefit for a person per month in Euro (nonnegative real number). Independent variables indicate whether the person has an own income (0/1), the living area (nonnegative real number), the rent (nonnegative real number), the number of family members in the household (natural number), sex (0/1), additional money from the public for being a single parent (nonnegative real number) and the monthly income of the household (nonnegative real number).

2.7 Old age beneficiaries data (ret.). This data set consists of 18345 data points of dimension 10. The dependent variable is the monthly pension of a person. The person is

described by age (natural number), sex (0/1), marriage status (0/1), the level of career: simple (0/1), average (0/1), advanced (0/1) or very advanced (0/1), whether the pension has started because the beneficiary has reached the age of 65 (0/1), whether the pension is a retirement pay (0/1) and percentage of the retirement pay (in percent).

2.8 Car accident data (acc.). This data set consists of 11739 data points of dimension 8, each one corresponding to a car accident. The dependent variable is the amount of damage estimated by the police (in Euro, nonnegative real number). The car accident is described by the level of alcohol in the blood (per mile), the age of the driver (natural number), the sex of the driver (0/1), the number of years the driver has had a driving license (in years, natural number), the engine power in kilowatts (nonnegative number), the empty weight of the car (in kg) and number of years the car has been registered (in years, natural number).

2.9 Income data (inc.). This data set consists of 7947 data points of dimension 14, each one describing a person. The dependent variable is the monthly net income of the person. The person is described by age in years (natural number), sex (0/1), marriage status (married or not) (0/1) and working status (working or not) (0/1), number of years since the last graduation (natural number) and the level of the highest graduation of the person coded by 8 $\{0,1\}$-valued covariates.

2.10 University exam data (exam). This data set consists of 9388 data points of dimension 20, each one describing one exam of a student at a university. The dependent variable is the mark in the exam (with values 1, 2, ..., 5). The student is described by sex (0/1), age in years (natural number), whether he/she has the German citizenship or not (0/1), whether he/she studies full time (0/1) or part time (0/1) or whether both are unknown (0/1), for how many months he has studied in a foreign country (natural number), whether he/she has completed professional education (0/1), whether he/she has made an internship in connection with his/her current studies (0/1), number of terms he/she studied the subject related to the exam (natural number) and 9 additional $\{0,1\}$-valued covariates describing the subject of the study.

3 Nine nonparametric regression estimates

3.1 General procedure

Given an i.i.d. sample (x_1, y_1), ..., (x_n, y_n) from a distribution (X, Y), where X is a $\mathbb{R}^d$-valued random variable and Y is a $\mathbb{R}$-valued random variable, our goal is to estimate the regression function $m : \mathbb{R}^d \to \mathbb{R}$ defined by $m(x) = \mathbf{E}\{Y|X = x\}$. It is well known that this function minimizes the so-called L_2 risk

$$\mathbf{E}\{|m(X) - Y|^2\} = \min_{f:\mathbb{R}^d \to \mathbb{R}} \mathbf{E}\{|f(X) - Y|^2\}$$

(cf., e.g., Sections 1.1 in Györfi et al. (2002)). We compare our estimates by estimating and comparing their L_2-risk. We accomplish this by first splitting our sample into three parts: a learning sample of size $n_l \approx n/3$, a testing sample of size $n_t \approx n/3$ and an evaluation sample of size $n_v = n - n_l - n_t \approx n/3$. We then use the learning and the

testing sample to generate estimates m_{n_l,p^*} of m (for details see below) and estimate the L_2 risk of the estimates by the empirical L_2 risk on the evaluation data, i.e., by

$$\frac{1}{n_{ev}} \sum_{i=n_l+n_t+1}^{n} |m_{n_l,p^*}(x_i) - y_i|^2.$$

Each estimate described below will depend on some parameter p (possibly a vector) which we choose from some finite set $\mathcal{P}$ of parameters via splitting of the sample. To do this, we use for each parameter the learning data to define estimates $m_{n_l,p}$ (as described below), and choose in a second step that parameter value for which the empirical L_2 risk is minimal on the testing data, i.e., we produce an estimate m_{n_l,p^*}, where

$$p^* = \arg\min_{p \in \mathcal{P}} \frac{1}{n_t} \sum_{i=n_l+1}^{n_l+n_t} |m_{n_l,p}(x_i) - y_i|^2.$$

In order to simplify the computation we preprocess the data. This step will be described in the next subsection. In the subsequent subsections we describe how the individual estimates are defined. For all estimates the smoothing parameters are estimated via the above method of splitting of the sample, in the sequel we only specify the parameter set $\mathcal{P}$.

3.2 Preprocessing

For some of the parameters of the estimates it is not clear which finite subset of the whole parameter space should be chosen in order to compute p^*. The main problem here is that this depends on the range of the independent variables. To overcome this problem we renormalize all datasets such that every independent variable in the union of learning and testing samples takes values in the interval $[0, 1]$. As for all datasets the covariates are nonnegative this can be simply achieved by dividing each independent variable by the maximal observation in the union of the learning and the testing sample. The corresponding independent variables of the evaluation set are divided by the same values.

3.3 Local averaging estimates (kern., part., nn)

The simplest way to define a regression estimate is to use local averaging where the estimate is defined by

$$m_n(x) = \sum_{i=1}^{n} W_{n,i}(x) \cdot y_i.$$

Here the weight $W_{n,i}(x)$ depends only on the x-value of the sample and on x and is chosen such that it is large if x_i is in some sense close to x and small otherwise.

For the **kernel estimate (kern.)** we choose a kernel function $K : \mathbb{R}^d \to \mathbb{R}$ and a bandwidth $h > 0$ and set

$$W_{n,i}(x) = \frac{K((x - x_i)/h)}{\sum_{j=1}^{n} K((x - x_j)/h)}.$$

Hence

$$m_n(x) = \sum_{i=1}^{n} \frac{K((x - x_i)/h) \cdot y_i}{\sum_{j=1}^{n} K((x - x_j)/h)}$$

(cf., e.g., Nadaraya (1964), Watson (1964)). We choose $K(u) = \exp(-\|u\|^2/2)$ (Gaussian kernel) and h from the set $\mathcal{P} = \{1/2^l : l = 0, 1, 2, \ldots, 10\}$.

For the **partitioning estimate (part.)** we choose a partition $\{A_{n,j} : j = 1, \ldots, K\}$ of $\mathbb{R}^d$ and set

$$W_{n,i}(x) = \frac{1_{\{X_i \in A_n(x)\}}}{\sum_{j=1}^{n} 1_{\{X_j \in A_n(x)\}}},$$

where $A_n(x)$ denotes the cell $A_{n,j}$ of the partition containing x. Consequently

$$m_n(x) = \sum_{i=1}^{n} \frac{1_{\{X_i \in A_n(x)\}} \cdot y_i}{\sum_{j=1}^{n} 1_{\{X_j \in A_n(x)\}}}$$

(cf., e.g., Györfi (1981)). We choose equidistant partitions, where for each component of x a compact interval is chosen and subdivided into K equidistant subintervals and the cross-product of all intervals of all components is used, and one remaining set is added to the partition which contains all remaining points in $\mathbb{R}^d$. Here for each component the compact interval is chosen as $[q_{0.01}, q_{0.99}]$, where q_α is the empirical α-quantile of the observations in the learning and testing set. This choice of the compact interval ensures that we have observations in the remaining set of the partition described above. The number K is chosen from the set $\mathcal{P} = \{2^l : l = 0, 1, 2, \ldots, 10\}$.

For the **nearest neighbor estimates (nn)** we use for each x a permutation

$$(x_{(1)}(x), y_{(1)}(x)), \ldots, (x_{(n)}(x), y_{(n)}(x))$$

of the data set (x_1, y_1), $\ldots$, (x_n, y_n) such that the distance of $x_i(x)$ to x is increasing, i.e.,

$$|x - x_{(1)}(x)| \leq |x - x_{(2)}(x)| \leq \cdots \leq |x - x_{(n)}(x)|.$$

(Here $|x|$ denotes the Euclidean norm of $x \in \mathbb{R}^d$.) Tie breaking is done by indices, i.e., in case of $|x - x_i| = |x - x_j|$ we choose the data point with the smaller index first. The nearest neighbor estimate depending on a parameter $k \in \{1, \ldots, n\}$ is defined by

$$m_n(x) = \frac{1}{k} \sum_{i=1}^{k} y_{(i)}(x)$$

(cf., e.g., Stone (1977) or Devroye et al. (1994)). The parameter k is chosen from the set $\mathcal{P} = \{2^l : l = 0, 1, 2, \ldots, 11\}$.

3.4 Least squares estimates (neur., add.)

For the least squares estimate the L_2 risk of a function f is estimated by its empirical L_2 risk

$$\frac{1}{n} \sum_{i=1}^{n} |f(x_i) - y_i|^2$$

and the latter one is minimized over some space of functions $\mathcal{F}$.

For **additive spline estimates (add.)** the function space is restricted to consist of additive functions of the form

$$f(x^{(1)}, \ldots, x^{(d)}) = f_1(x^{(1)}) + \cdots + f_d(x^{(d)}).$$

For f_i we consider univariate B-splines $B_{(M,\alpha)}$ with equidistant knot sequences. Here M is the degree of the B-spline and α is the distance between two consecutive knots. We consider knot sequences on the interval $[0, 1]$. Then $\mathcal{F}$ is defined as the linear span of all functions of the above form and the least squares estimate is defined by

$$m_n = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} |f(x_i) - y_i|^2$$

(cf., Stone (1985)). The parameter (M, α) of the estimate is chosen from the set

$$\mathcal{P} = \{(M, \alpha) \, : \, M \in \{0, 1, 2, 3\}, \, \alpha \in \{1/2^l \, : \, l = 0, 1, 2, \ldots, 10\}\}.$$

For **neural network estimates (neur.)** we set

$$\sigma(u) = \frac{1}{1 + \exp(-u)}$$

(logistic squasher) and define the function space $\mathcal{F}$ for the least squares estimate depending on a parameter $K \in \mathbb{N}$ as the set of all functions of the form

$$f(x) = c_0 + \sum_{k=1}^{K} c_k \cdot \sigma \left(\sum_{j=1}^{d} a_{j,k} \cdot x^{(j)} + b_k \right)$$

for $a_{j,k}, b_k, c_k \in \mathbb{R}$ (cf., e.g., Chapter 11 in Hastie, Tibshirani and Friedman (2009)). The corresponding least squares estimate cannot be computed exactly since in general the corresponding nonlinear minimization cannot be solved. Instead a gradient descent algorithm (so-called backpropagation) is used to compute the estimate approximately. For this we use the Neural Network Toolbox in MATLAB®. Finally we choose the parameter K of the estimate from the set $\mathcal{P} = \{2^l \, : \, l = 0, 1, 2, \ldots, 7\}$.

3.5 Penalized smoothing splines (spli.)

For penalized least squares estimates we also consider the empirical L_2 risk but we add a term which penalizes the roughness of the function, i.e.,

$$m_n = \arg\min_{f \in \mathcal{F}} \left(\frac{1}{n} \sum_{i=1}^{n} |f(x_i) - y_i|^2 + \lambda \cdot J_k^2(f) \right),$$

where

$$J_k^2(f) = \sum_{\alpha_1, \ldots, \alpha_d \in \mathbb{N}, \, \alpha_1 + \cdots + \alpha_d = k} \frac{k!}{\alpha_1! \cdot \cdots \cdot \alpha_d!} \int_{\mathbb{R}^d} \left| \frac{\partial^k f}{\partial x_1^{\alpha_1} \ldots \partial x_d^{\alpha_d}}(x) \right|^2 dx$$

(cf, e.g., Wahba (1990)). For the penalized smoothing spline we consider for $2k > d$ the class of functions called the Sobolev class

$$\begin{aligned}\mathcal{F} &= W^k(\mathbb{R}^d) \\ &= \left\{ f \,:\, \frac{\partial^k f}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}} \in L_2(\mathbb{R}^d) \text{ for all } \alpha_1, \dots, \alpha_d \in \mathbb{N} \text{ with } \alpha_1 + \dots + \alpha_d = k \right\}.\end{aligned}$$

It can be shown that in this case m_n can be found by solving a system of linear equations, for more details see for example Chapter 20.4 of Györfi et al. (2002). For our computation we fix $k = \lceil d/2 \rceil$ and choose λ from the set $\mathcal{P} = \{2^l \,:\, l = -20, -19, \dots, 19, 20\}$.

3.6 Local linear kernel regression estimate (loc.)

The kernel estimate fits locally a constant to the data (cf., e.g., Problem 2.2 in Györfi et al. (2002)). Local linear kernel estimates extend this by fitting locally a linear function to the data and by using the function value at point x as an estimate of $m(x)$, i.e.,

$$(c_0^*, \dots, c_d^*) = \arg \min_{(c_0, \dots c_d) \in \mathbb{R}^{d+1}} \frac{1}{n} \sum_{i=1}^{n} K((x - x_i)/h) \cdot (y_i - (c_0 + c_1 \cdot x_i^{(1)} + \dots + c_d \cdot x_i^{(d)}))^2$$

and

$$m_n(x) = c_0^* + c_1^* \cdot x^{(1)} + \dots + c_d^* \cdot x^{(d)}$$

(cf., e.g., Stone (1982)). Again we choose $K(u) = \exp(-\|u\|^2/2)$ and h from the set $\mathcal{P} = \{1/2^l \,:\, l = 0, 1, 2, \dots, 10\}$.

3.7 Regression trees (tree)

In principle the regression trees are the least squares estimates where a piecewise constant function is fitted to the data. They are especially suited for the high-dimensional data. The underlying partition is generated recursively, where in each step one component of x is chosen and one of the sets there is subdivided at some split point into two parts. The component and the split point are chosen in some greedy way favoring the choice that leads to the highest decrease of the empirical L_2 risk of the estimate (cf., e.g., Breiman et al. (1984)). Splitting is only performed if a minimum number of observations per leaf *min_leaf* can be established. This is done until the restriction implied by *min_leaf* terminates the algorithm, leading to the sequence of partitions with increasing cardinality and corresponding piecewise constant estimates. One of these estimates is chosen as the final estimate by using a pruning step: The overall performance of a partition π leading to a piecewise constant estimate $m_{n,\pi}$ is described by

$$\frac{1}{n} \sum_{i=1}^{n} |m_{n,\pi}(x_i) - y_i|^2 + c \cdot |\pi|$$

where $|\pi|$ denotes the number of cells in π and $c > 0$ is a parameter of the estimate. Different values of c lead to at most n different partitions, so de facto the parameter is a

number between 1 and n. Moreover we choose $min\_leaf \in \{2^l : l = 0, 1, 2, \ldots, 7\}$. For the implementation of the estimate we use the function classregtree from the Statistics Toolbox in MATLAB®.

3.8 Random forests (forest)

Random forests, proposed by Breiman (2001), estimate the regression function by computing many different regression trees without pruning on subsamples of the data and by averaging the corresponding predictions of the individual estimates. To compute the k-th estimate $m_{n,k}$ a subsample of size n is chosen randomly from the original sample with replacement (so a data point might occur several times in the subsample) and a regression tree is computed on that subsample without pruning. Here instead of taking into account all variables for splitting, at each node only a small number F of input variables is chosen at random. The random forest estimate is defined as the average prediction of all trees, i.e.,

$$m_n(x) = \frac{1}{K} \sum_{k=1}^{K} m_{n,k}(x).$$

The parameters which occur are min_leaf as above, $K \in \{2^l : l = 0, 1, 2, \ldots, 11\}$ and $F \in \{1, 2, ..., \lceil d/2 \rceil\}$, where d is the number of covariates and $\lceil z \rceil$ denotes the smallest integer greater than or equal to z. For the implementation of the estimate we use the function treebagger from the Statistics Toolbox in MATLAB®.

4 Empirical comparison of the estimates

In order to represent risks in a reasonable range we present in the sequel relative risks where we consider the fractions of the empirical L_2 risks of our estimates and the empirical L_2 risk of a constant estimate given by the arithmetic mean of the dependent variable computed using only the training data. As a consequence the relative errors are always greater than or equal to zero and most of the time also less than one. As baseline we use a simple linear estimate (lin.), which just computes a linear regression using all covariates on the training data and uses this to predict the values of the dependent variable on the evaluation data.

In Table 1 we compare the empirical L_2 risk of the constant estimate with the empirical L_2 risk of the (nonparametric) regression estimate performing best. From Table 1 we see that our data sets are rather mixed: for one data set (tax) the data can be predicted very well, but for four other (health, birth, acc. and exam) currently none of our estimates can explain more than 25% of the noise.

In Table 2 we present the relative empirical L_2 risks of all 9 nonparametric regression estimates and the simple linear estimate on the evaluation part of all ten data sets. In each row the best L_2 risk is marked in boldface. All L_2 risks not larger than the best value plus 5% of the improvement with respect to the constant estimate are underlined. For two estimates in high dimensions where the memory was insufficient to compute predictions, missing results are denoted by $-$. The last two lines describe how often the

Estimate	health	agric.	birth	tax	build
constant	$4.0755 \cdot 10^5$	$3.3329 \cdot 10^9$	$3.2196 \cdot 10^5$	$7.9260 \cdot 10^{10}$	$3.7671 \cdot 10^5$
best est.	$3.1810 \cdot 10^5$	$1.2603 \cdot 10^9$	$2.7168 \cdot 10^5$	$3.2082 \cdot 10^7$	$1.2188 \cdot 10^5$
percentage	78.05%	37.81%	84.38%	0.040%	32.35%

Estimate	liv.	ret.	acc.	inc.	exam
constant	$8.8377 \cdot 10^3$	$1.0562 \cdot 10^6$	$3.9879 \cdot 10^7$	$2.5412 \cdot 10^6$	0.7179
best est.	$1.6986 \cdot 10^3$	$2.3304 \cdot 10^5$	$3.2534 \cdot 10^7$	$1.8696 \cdot 10^6$	0.6236
percentage	19.22%	22.06%	81.58%	73.57%	86.86%

Table 1. Empirical L_2 risk on the evaluation data of the constant estimate and the best estimate for all ten data sets.

Data	lin.	part.	kern.	nn	neur.	tree	forest	add.	loc.	spli.
health	79.41	80.32	83.54	81.43	78.08	80.27	78.61	79.43	78.13	**78.05**
agric.	**37.81**	41.39	80.04	39.59	38.63	44.1	38.29	38.23	>160	38.47
birth	84.44	84.6	86.41	86.49	84.71	84.42	84.66	84.55	85.66	**84.38**
tax	0.084	10.63	49.14	7.792	0.100	21.61	23.04	0.403	0.539	**0.040**
build	86.62	52.15	56.28	39.59	**32.35**	40.42	35.00	>160	>160	44.89
liv.	41.89	29.57	52.06	28.04	19.78	25.13	**19.22**	22.39	39.6	28.27
ret.	27.15	22.78	26.82	23.57	**22.06**	23.28	22.22	24.51	22.82	22.1
acc.	84.25	81.63	105.8	**81.58**	83.01	85.24	81.97	84.36	94.49	81.7
inc.	76.04	82.17	92.43	75.91	74.47	78.9	**73.57**	76.39	103.9	-
exam	91.37	92.92	95.48	90.61	91.3	88.97	**86.86**	91.52	-	-
best	1	0	0	1	2	0	3	0	0	3
b.+5%	3	3	0	3	8	2	9	5	3	6

Table 2. Relative empirical L_2 risk on the evaluation data (in percent).

relative L_2 risk of an estimate is the best value (first line) and how often it is not larger than the best value plus 5% of the improvement with respect to the constant estimate.

From Table 2 we see that for one of the data sets (agric.) the linear regression is currently the best one, and for two more (birth and tax) linear regression performs not much worse than the best one. However, for all other data sets the best nonparametric regression estimate clearly outperforms the linear regression. Furthermore it can be seen in Table 2 that two estimates stand out. They are neural networks and random forests. For almost all datasets both are able to produce results that are at most 5% away from the best improvement achieved with respect to the constant estimate.

References

[1] Breiman, L. (2001). Random Forests. *Machine Learning*, **45**, pp. 5–32.

[2] Breiman, L., Friedman, J., Olshen, R., Stone, C. (1984). *Classification and Regression Trees*. Wadsworth, Belmont, CA.

[3] DASL (1996). *The Data and Story Library*, [http://lib.stat.cmu.edu/DASL/DataArchive.html].

[4] Devroye, L., Györfi, L., Krzyżak, A., and Lugosi, G. (1994). On the strong universal consistency of nearest neighbor regression function estimates. *Annals of Statistics*, **22**, pp. 1371–1385.

[5] Frank, A. and Asuncion, A. (2010). *UCI Machine Learning Repository*. Irvine, CA: University of California, School of Information and Computer Science, [http://archive.ics.uci.edu/ml].

[6] Györfi, L. (1981). Recent results on nonparametric regression estimate and multiple classification. *Problems of Control and Information Theory*, **10**, pp. 43–52.

[7] Györfi, L., Kohler, M., Krzyżak, A. and Walk, H. (2002). *A Distribution-Free Theory of Nonparametric Regression*. Springer-Verlag, New York.

[8] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. 2nd Edition. Springer-Verlag, New York.

[9] MATLAB® version 7.13.0.564. Natick, Massachusetts: The MathWorks Inc., 2011.

[10] Nadaraya, E. A. (1964) On estimating regression. *Theory of Probability and its Applications*, **9**, pp. 141–142.

[11] Statistical Software Information - University of Massachussets Amherst (2004). *Statistical Datasets*, [http://www.umass.edu/statdata/statdata].

[12] Stone, C. J. (1977). Consistent nonparametric regression. *Annals of Statististics*, **5**, pp. 595–645.

[13] Stone, C. J. (1982). Optimal global rates of convergence for nonparametric regression. *Annals of Statistics*, **10**, pp. 1040–1053.

[14] Watson, G. S. (1964). Smooth regression analysis. *Sankhya Series A*, **26**, pp. 359–372.

[15] Wahba, G. (1990). *Spline Models for Observational Data*. SIAM, Philadelphia.