Report

Regression and non-parametric estimators for two-phase forest inventories in the design-based Monte-Carlo approach

Author(s):
Mandallaz, Daniel; Massey, Alexander

Publication Date:
2015-04

Permanent Link:
https://doi.org/10.3929/ethz-a-010421779

Rights / License:
In Copyright - Non-Commercial Use Permitted
Regression and non-parametric estimators for two-phase forest inventories in the design-based Monte-Carlo approach

Daniel Mandallaz and Alexander Massey

Chair of Land Use Engineering

Department of Environmental Systems Science

ETH Zurich

April 2015
Abstract

This technical report adapts rank based nearest neighbor methods to classical kernel estimators using neighborhoods defined by a metric in the feature space resting upon the predictions obtained from a linear model. The approach is purely design-based. The main concepts of kernel-based density and regression estimators are summarized and reformulated in the design-based Monte-Carlo approach to forest inventory. New estimators are proposed which are based on the conditional expectation of the response variable given its prediction and are compared to the classical two-phase regression estimator and to two-phase k-nearest neighbor (k-NN) estimators, using either a multidimensional metric or a one-dimensional metric based on the Euclidian distance between the predictions. The results suggest that the commonly used analytical external variance formula may systematically underestimate the true variance for a variety of nonparametric kernel based estimators including kNN, but is still adequate for the classical regression estimator. Although using a bootstrap variance can help to correct this underestimation it was also found that the bootstrap variance estimates can be unstable if the bandwidth or the number of neighbors are recalculated in each bootstrap sample. These findings suggest that if the model captures the main features of the underlying process, which is clearly the case for timber volume estimates based on canopy characteristics obtained from LiDAR measurements, then it is advisable to use the classical regression estimator because it performs at least as well as the other techniques and is by far the simplest to implement.
1 Introduction

In this note we use the Monte-Carlo approach (infinite population model), as described in Mandallaz (2008) (Chapter 5). We consider the forested area $F$ of surface area $\lambda(F)$ for which the sum $Y_F = \sum_{i=1}^{N} Y_i$ of the response variable $Y_i$ for a well defined population of $N$ trees, e.g. the total volume of the $N$ trees, and the spatial mean $\bar{Y}_F = \frac{1}{\lambda(F)} \sum_{i=1}^{N} Y_i$ must be estimated.

2 Two-phase sampling

We have a sample $s_1$ of $n_1$ points independently and uniformly distributed in $F$ out of which $n_2$ are selected by simple random sampling without replacement. At all points $x \in s_1$ we have the vector of auxiliary information $Z(x) \in \mathbb{R}^p$ and at all terrestrial points $x \in s_2$ we have the local density $Y(x)$ which is essentially a Horwitz-Thompson estimator at point $x \in \mathbb{R}^2$. In the Monte-Carlo approach we have the fundamental relationship

$$\frac{1}{\lambda(F)} \int_F Y(x) dx = \frac{1}{\lambda(F)} \sum_{i=1}^{N} Y_i$$

We consider the linear model

$$Y(x) = Z^t(x) \beta + R(x)$$

In the model-dependent approach the point $x$ is fixed and $R(x)$ is a random variable with zero mean and a given covariance structure (in most instances very simple and not really adapted to the spatial nature of the problem in contrast to geostatistical techniques). In the design-based approach $Y(x)$, $Z(x)$ and $R(x)$ are random variables because $x$ is random. According to the terminology of Särndal et al. (2003) our approach is model-assisted, i.e. we use models to reduce the variance but we do not assume that they are correct. The inference is design-based and valid because it rests upon the randomization principle. Some authors call model-based the model-dependent approach, which is in our opinion a source of confusion because the inference is valid only if the model is true. Here, the true regression coefficient $\beta$ is by definition the theoretical least squares estimate minimizing

$$\int_F R^2(x) dx = \int_F (Y(x) - Z^t(x) \beta)^2 dx$$

It satisfies the normal equation

$$\left( \int_F Z(x) Z^t(x) dx \right) \beta = \int_F Y(x) Z(x) dx$$

and the orthogonality relationship

$$\int_F R(x) Z(x) dx = 0$$
We shall assume that $Z(x)$ contains the intercept term 1, or, more generally, that the intercept can be expressed as a linear combination of the components of $Z(x)$, which then ensures that the mean residual is zero, i.e. $\int_F R(x)dx = 0$. To simplify the notation we define the following

$$
U(x) := Y(x)Z(x)
$$

$$
A_{s_2} := \frac{1}{n_2} \sum_{x \in s_2} Z(x)Z'(x)
$$

$$
U_{s_2} := \frac{1}{n_2} \sum_{x \in s_2} U(x)
$$

The theoretical and empirical regression vector parameters can then be written as

$$
\beta := A^{-1}U
$$

$$
\hat{\beta}_{s_2} := A_{s_2}^{-1}U_{s_2}
$$

where we have set $A = \mathbb{E}_{x \in F}Z(x)Z'(x)$ and $U = \mathbb{E}_{x \in F}Z(x)Y(x)$. $\hat{\beta}_{s_2}$ is asymptotically design-unbiased for $\beta$.

The empirical predictions and residuals are $\hat{Y}(x) = Z'(x)\hat{\beta}_{s_2}$ and $\hat{R}(x) = Y(x) - \hat{Y}(x)$. The asymptotic design-based variance-covariance matrix of $\hat{\beta}_{s_2}$ is

$$
\Sigma_{\hat{\beta}_{s_2}} := A_{s_2}^{-1}\left( \frac{1}{n_2^2} \sum_{x \in s_2} \hat{R}^2(x)Z(x)Z'(x) \right) A_{s_2}^{-1}
$$

and the two-phase classical regression estimator is

$$
\hat{Y}_{reg} = \frac{1}{n_1} \sum_{x \in s_1} \hat{Y}(x) + \frac{1}{n_2} \sum_{x \in s_2} \hat{R}(x)
$$

In the following we shall work under the external model assumption, i.e. we neglect the error in $\hat{\beta}_{s_2}$ and set formally $\hat{\beta}_{s_2} \equiv \beta$, which is asymptotically equivalent to the variance obtained via the design-based variance covariance matrix (Mandallaz (2008), p. 125). One has the theoretical external variance

$$
\mathbb{V}(\hat{Y}_{reg}) = \frac{1}{n_1} \mathbb{V}_{x \in F}(Y(x)) + \left( 1 - \frac{n_2}{n_1} \right) \frac{1}{n_2} \mathbb{V}_{x \in F}\hat{R}(x)
$$

$$
\mathbb{V}(\hat{Y}_{reg}) = \frac{1}{n_1} \mathbb{V}_{x \in F}(\hat{Y}(x)) + \frac{1}{n_2} \mathbb{V}_{x \in F}\hat{R}(x)
$$

Sample copies of the second version yields the external variance estimate asymptotically equivalent to the g-weight variance based on $\Sigma_{\hat{\beta}_{s_2}}$ discussed in (Mandallaz (2008) and Mandallaz (2013),), which has slightly better statistical properties from a theoretical point of view. The working model $Z(x)$ yielding $\hat{Y}(x)$ does not have to be identical with the true model model generating $Y(x)$. On the other hand, the closer the working model
comes to the true model the smaller the residual variance and therefore the error of the regression estimator.

In this report the predictions are based on linear models, external or internal, the difference being asymptotically negligible. It is intuitively clear that the predictions \( \hat{Y}(x) \) could also be based on non-linear models, which leads to so-called calibration model discussed by Wu and Sitter (2001). The proof of this result is sketched in Appendix C.

3 Kernel-based estimators

In this section we consider non-parametric kernel based regression estimator. The idea is very simple: if the working model departs from the true unknown model, e.g. because relevant explanatory variables have not been incorporated into the working model due to either inadequate or unavailable hidden variables, then one can attempt to combine locally \( \hat{Y}(x) \) and \( Y(x) \) to reduce the residual size \( |Y(x) - \hat{Y}(x)| \).

3.1 Preliminaries

We first consider one dimensional kernels

\[ K : u \in \mathbb{R} \rightarrow K(u) > 0 \]

The kernel \( K() \) is a probability density function with zero mean and finite variance, i.e \( \int_{\mathbb{R}} K(u)du = 1, \int_{\mathbb{R}} uk(u)du = 0 \) and \( \int_{\mathbb{R}} u^2 K(u)du < \infty \). Popular choices are

\[ K(u) = 0.5I_{[-1,1]}(u) \quad \text{uniform kernel} \]
\[ K(u) = \frac{1}{\sqrt{2\pi}} \exp(-u^2/2) \quad \text{normal kernel} \]

\[ K(u) = \frac{3}{4\sqrt{5}}(1 - \frac{u^2}{5})I_{|u|\leq\sqrt{5}}(u) \quad \text{Epanechnikov kernel} \]

The first and third kernels have a finite support whereas the normal kernel does not. The concept can be easily generalized to multidimensional kernel \( K(u) \) with \( u = (u_1, u_2, \ldots u_p) \in \mathbb{R}^p \), by setting \( k(u) = \prod_{l=1}^{p} K_l(u_l) \) where the \( K_l(\cdot) \) are 1-d kernels.

For a sample size \( n_2 \) (or respectively \( n_1 \)) we can use a global bandwidth \( \epsilon(n_2) \rightarrow 0 \) or a local bandwidth sequence \( \epsilon(n_2, x_0) \) \( (x_0 \text{ being a given point } x_0 \in F \text{ for which one wants to get estimates}) \) where the dependence on \( x_0 \) is usually via \( Y(x_0), \hat{Y}(x_0) \) or \( R(x_0) \) depending on the context. Many results follow intuitively from the fact that \( K(\frac{u}{\epsilon(n_2, x_0)}) \) tends to the Dirac \( \delta \) function as \( \epsilon(n_2, x_0) \) tends to 0.

In the nearest neighbor literature a wide choice of metric in the feature space (i.e. distances between points \( Z(u) \) and \( Z(v) \) for any two \( u, v \in F \)) have been used. From a geometric point of view one can always orthogonalize the \( Z(u) \) so that the Euclidian and Mahalanobis distances are essentially equivalent. Also, it is clear that

\[ ||Z(u) - Z(v)|| = ||\beta'(Z(u) - Z(v))|| = |\hat{Y}(u) - \hat{Y}(v)| \]
also defines a metric in the feature space, which is for any given \( \beta \) and in particular for the theoretical least squares estimate, a natural choice for the problem at hand. Furthermore, the multidimensional feature space is reduced to the one-dimensional space of the predictions, thus avoiding the use of multidimensional kernels, the curse of dimensionality, as well as the difficulties arising from the different scales of the explanatory variables. One should also recall that in finite dimensional spaces, all norms are equivalent from a topological point of view (\( || Z(u) || := | \beta^T Z(u) | \) is a norm). Of course, in practice one will usually replace \( \beta \) by its estimate and do as if it were a fixed value, as under the external model assumption.

We shall consider the kernel-based prediction \( \hat{Y}^{(1)}(x_0) \) defined point-wise for a point \( x_0 \in s_1 \) as

\[
\hat{Y}^{(1)}(x_0) := \frac{1}{n_2(n_2, \hat{y}_0)} \sum_{x \in s_2} Y(x) K\left( \frac{\hat{Y}(x) - \hat{Y}(x_0)}{c(n_2, \hat{y}_0)} \right)
\]

where \( \hat{y}_0 = \hat{Y}(x_0) \). If the numerator in [11] is zero then one sets \( \hat{Y}^{(1)}(x_0) = \hat{Y}(x_0) \).

\( \hat{Y}^{(1)}(x_0) \) is the Nadaraya-Watson estimator based on the euclidian distance between linear predictions and can be viewed as an estimate of the conditional expectation of \( Y(x_0) \) given \( \hat{Y}(x_0) \) (see section 3.2 and Appendix A).

It is interesting to consider the special case in which the \( \hat{Y}(x) \) can only take a finite number of different values, such as in post-stratification, still an important and widely used technique in forest inventory. Recall that in post-stratification \( \hat{Y}(x) \) is simply the observed stratum mean. For \( \epsilon(n_2, \hat{y}_0) \) small enough (very moderate smoothing) it is easy to see that \( \hat{Y}^{(1)}(\hat{y}_0) \) will tend to the mean of the \( Y(x) \) for the \( x \) in the same stratum \( F_k \subset F \) as \( x_0 \). If the \( \hat{Y}(x) \) (for \( x \in s_1 \)) are all distinct then the limit for \( \epsilon(n_2, \hat{y}_0) \to 0 \) is \( \hat{Y}(x_0) \) for \( x_0 \in s_2 \), when the bandwidth \( \epsilon(n_2, \hat{y}_0) \) tends to infinity (strong smoothing) then \( \hat{Y}^{(1)}(x_0) \) will tend to the overall sample mean \( \frac{1}{n_2} \sum_{x \in s_2} Y(x) = Y_{s_2} \), so that the model is completely ignored.

Note that if all explanatory variables are discrete then \( Z(u) \) can be viewed as a set of 0 – 1 indicator variables and the distance between two points will also take a finite number of values (e.g. say 0, 1 in the case of one-way analysis of variance and 0, 1, . . . q in general ANOVA models). Hence, kernel estimators are likely to be potentially useful only if some of the explanatory variables are continuous so that the set of local densities and predictions \( (Y(x), \hat{Y}(x)) \in \mathbb{R}^2 \) can be viewed as a continuum from a mathematical point of view (in practice everything is discrete!).

For the uniform kernel \( \hat{Y}^{(1)}(x_0) \) is simply the mean of all observations in the \( \epsilon(n_2, x_0) \) neighborhood, whereas the k-nearest neighbors methods take the mean of the observations whose ranked distances are less than the integer \( k \). With discrete metrics one has ties in the ranked distances and one could raise the question whether it is meaningful to take the mean of only \( k \) observations if there are more with the same ranks.

For the above reasons and for mathematical convenience we shall assume in the following that \( \hat{Y}(x) \) and \( Y(x) \) have sufficiently regular probability density functions (e.g. twice differentiable), even if the proposed kernel estimators can be used in practice when this
is not the case. In other words we have

\[ P(\hat{Y}(x) \in A) = \int_A f_Y(y)dy, \ \forall A \subset \mathbb{R} \]

\[ P(Y(x) \in A) = \int_A f_Y(y)dy, \ \forall A \subset \mathbb{R} \]

Note that this implies that \( P(Y(x_1) = Y(x_2)) = 0 \) for all \( x_1 \neq x_2 \) and likewise for \( \hat{Y}(\cdot) \).

Under some regularity assumptions one can show for \( \hat{Y}(x) \) with a continuous distribution function on \( \mathbb{R} \) that one has \( \hat{Y}(x_0) \to Y(x_0) \) as \( \epsilon(n_2, x_0) \to 0 \), convergence being in the mean, probability, almost surely, point-wise or uniform, depending on the regularity assumptions (see Tsybakov (2009) for a modern introduction).

We define the two-phase kernel-based regression estimator as

\[ \hat{Y}_{\epsilon, \text{reg}}^{(1)} := \frac{1}{n_1} \sum_{x_0 \in s_1} \hat{Y}(x_0) + \frac{1}{n_2} \sum_{x \in s_2} \hat{R}_{\epsilon}^{(1)}(x) \]

where \( \hat{R}_{\epsilon}^{(1)}(x) = Y(x) - \hat{Y}_{\epsilon}^{(1)}(x) \). Note that the sum of the residuals is no longer zero in general.

By analogy with the classical theory one can hope that \( \hat{Y}_{\epsilon, \text{reg}}^{(1)} \) is asymptotically design unbiased with the following design-based variance estimate

\[ \hat{V}(Y_\epsilon^{(1)}_{\text{reg}}) = \frac{1}{n_1} \hat{V}(Y(x)) + \left( 1 - \frac{n_2}{n_1} \right) \frac{1}{n_2(n_2 - 1)} \sum_{x \in s_2} (\hat{R}_{\epsilon}^{(1)}(x) - \bar{R}_{\epsilon, 2}^{(1)})^2 \]

where

\[ \hat{V}(Y(x)) = \frac{1}{n_2 - 1} \sum_{x \in s_2} (Y(x) - \bar{Y}_2)^2, \ \bar{Y}_2 = \frac{1}{n_2} \sum_{x \in s_2} Y(x), \ \bar{R}_{\epsilon, 2}^{(1)} = \frac{1}{n_2} \sum_{x \in s_2} \hat{R}_{\epsilon}^{(1)}(x) \]

### 3.2 Design-based kernel regression estimators

There is an immense literature on kernel-based estimators but the overwhelming majority of the papers are in the model-dependent one-dimensional framework, i.e. \( Y(x_i) = m(x_i) + \epsilon(x_i), \ x_i \in \mathbb{R}^1, i = 1, 2 \ldots n, \) with \( m(x_i) = \mathbb{E}(Y(x_i) \mid x_i) \) and \( \mathbb{V}(Y(x_i) \mid x_i) = \sigma^2(x_i) \), the \( x_i \) may be fixed or randomly independently distributed on \( \mathbb{R}^1 \). Informal proofs of the most important results in the standard set-up are given in Appendix A and Appendix B.

Recall that in general the best predictor is given by the conditional expectation. Ideally one would therefore consider the conditional expectation of \( Y(x) \) given the auxiliary information, i.e. \( \mathbb{E}(Y(x) \mid Z(x) \in \mathbb{R}^p) \) which requires the use of multivariate kernels to estimate multivariate densities and, as afore mentioned, is subject to the curse of dimensionality problem when \( p \) is large (in practice one rarely goes beyond \( p = 2 \)). In some
sense one can view \( \hat{Y}(x) \in \mathbb{R}^1 \) as a concise summary of \( \mathbf{Z}(x) \in \mathbb{R}^p \) which leads us to consider instead

\[
E(Y(x) \mid \hat{Y}(x)) = E(\hat{Y}(x) + R(x) \mid \hat{Y}(x)) = \hat{Y}(x) + E(R(x) \mid \hat{Y}(x))
\]

To have an approximation of this conditional expectation we need estimates of the joint bivariate density of \((R(x), \hat{Y}(x))\) and of the marginal density of \(\hat{Y}(x)\). This is done with bivariate and univariate Nadaraya-Watson estimators to obtain first an estimate of the conditional density and then, by integration, the estimate of the conditional expectation (see Appendix A for details). Asymptotically we replace the true predictions \(\hat{Y}(x) = \beta'\mathbf{Z}(x)\) and residuals \(R(x) = Y(x) - \beta'\mathbf{Z}(x)\) by their empirical versions, e.g. after replacing \(\beta\) by \(\hat{\beta}_{s_2}\). Using the first term in (16) this leads to the estimator \(\hat{Y}_{(1)}(x_0)\) given in (11) and using the third we have the following estimate for the conditional expectation of the residual

\[
\hat{E}(\hat{R}(x_0) \mid \hat{Y}(x_0)) = \frac{1}{n_2 \epsilon_{y}(n_2, y_0)} \sum_{x \in s_2} \hat{R}(x) K\left(\frac{\hat{Y}(x_0) - \hat{Y}(x)}{\epsilon_{y}(n_2, y_0)}\right)
\]

\[
= \frac{1}{n_2 \epsilon_{y}(n_2, y_0)} \sum_{x \in s_2} \hat{R}(x) K\left(\frac{Y(x_0) - \hat{Y}(x)}{\epsilon_{y}(n_2, y_0)}\right)
\]

\[
= \sum_{x \in s_2} \hat{R}(x) K\left(\frac{Y(x_0) - \hat{Y}(x)}{\epsilon_{y}(n_2, y_0)}\right)
\]

\[
[17]
\]

and consequently the following kernel-based estimators at \(x_0\) at any point \(x_0 \in s_1\) in the large sample

\[
[18]
\hat{Y}_{(2)}(x_0) = \hat{Y}(x_0) + \hat{R}_{(2)}(x_0)
\]

From a formal point of view \(\hat{Y}_{(2)}(x_0)\) has a slight advantage over \(\hat{Y}_{(1)}(x_0)\) when the bandwidth goes to \(\infty\) as it tends to \(\hat{Y}(x_0)\) whereas \(\hat{Y}_{(1)}(x_0)\) tends to the overall sample mean \(\bar{Y}_{s_2}\). This also means that \(\hat{Y}_{(2)}(x_0)\) can escape the range of observations \(Y(x)\) where \(x \in s_2\) whereas \(\hat{Y}_{(1)}(x_0)\) can not.

Note that the probability density function of the predictions could also be estimated in the large sample \(s_1\), a possibility we shall not pursue further here, primarily because it has not been investigated yet in the literature and because it is not available in standard software (such as \(R\)), at least not to our knowledge.

**Mathematical difficulties**

The true predictions \(\hat{Y}_0(x) = \beta'\mathbf{Z}(x)\) and observations \(Y(x)\) are i.i.d. but the points \((\hat{Y}_0(x), Y(x))\) are on a 1-d curve in \(\mathbb{R}^2\) (possibly with discontinuities) so that one has a degenerate bivariate distribution. This is not the case with the empirical predictions \(\hat{Y}(x) = \hat{\beta}_{s_2}'\mathbf{Z}(x)\), which are only asymptotically uncorrelated so that the formulae for the variances of the various kernel-based estimators (given in Appendix A and Appendix B)
are probably at best only asymptotically valid, possibly with a lower convergence rate. On the other hand, the kernel-estimators have a strong intuitive background, and they remain meaningful even if the set of predictions is discrete.

Neglecting the correlation between the predictions (asymptotically of order $O(\frac{\epsilon(n_2,\hat{y}_0)}{n_2})$), the results given in Appendix B show that the convergence rate at $x_0$ is of order $O(\frac{\epsilon(n_2,\hat{y}_0)}{n_2})$. It can be conjectured that the convergence rate will be slower if one takes the correlations of the predictions into account, but this seems to be beyond simple analytical treatment. In any case, one often relies in practice on the external model approach and use as usual \[14\] and \[15\] with $\hat{Y}^{(2)}(x_0)$ and $\hat{R}^{(2)}(x_0)$. It turns out that the external variance can severely underestimates the true variance, except for the classical regression estimator $\hat{Y}_{reg}$, and that one must use bootstrap to obtain correct variance estimates and confidence intervals.

Remarks:

1. In the case of post-stratification $\hat{R}^{(2)}_t(x_0)$ will tend to the sum of the residuals in the stratum of $x_0$ as $\epsilon_\theta(n_2,\hat{y}_0) \rightarrow 0$, and therefore to zero. Hence, in this case, $\hat{Y}^{(2)}_t(x_0) \rightarrow \hat{Y}(x_0)$ and nothing is gained. If the predictions are all distinct it will tend to $\hat{R}(x_0)$ for $x_0 \in s_2$ and $\hat{Y}^{(2)}_t(x_0) \rightarrow \hat{Y}(x_0)$.

2. If the predictions are all different and $\epsilon_\theta(n_2,\hat{y}_0) \rightarrow 0$ then $\hat{R}^{(2)}_t(x_0)$ will tend to $\hat{R}(x_0)$ for $x_0 \in s_2$ and to 0 if $x_0 \in s_1 \setminus s_2$. Hence, $\hat{Y}^{(2)}_t(x_0) \rightarrow \hat{Y}(x_0)$ $\forall x_0 \in s_2$ and $\hat{Y}^{(2)}_t(x_0) \rightarrow \hat{Y}(x_0)$ $\forall x_0 \in s_1 \setminus s_2$. The resulting regression estimator is then \[ \frac{1}{n_2} \sum_{x_0 \in s_1} \hat{Y}(x_0) \] with estimated external variance \[ \frac{1}{n_2} \hat{V}(\hat{Y}(x)) \] because all the residuals in $s_2$ are zero! This is of course to good to be true! The reason is that to ensure consistency of the estimated densities we cannot simply set $\epsilon_\theta(n_2,\hat{y}_0) = 0$, one needs also \( \lim_{n_2 \rightarrow \infty} n_2 \epsilon_\theta(n_2,\hat{y}_0) = \infty \). Hence, we cannot determine the bandwidth by minimizing the resulting variance of the regression estimator (the minimum being achieved with zero residual variance) but we have to rely instead on cross-validation procedures. Note also that for $\epsilon_\theta(n_2,\hat{y}_0) \rightarrow \infty$ one has $\hat{Y}^{(2)}_t(x_0) \rightarrow \hat{Y}(x_0)$ and $\hat{Y}^{(1)}_t(x_0) \rightarrow \hat{Y}_{s_2}$.

3. If some important explanatory variables are not accounted for it may well be that locally we do not have zero mean residual (i.e. a pattern in the structure of the residuals), in which case the new kernel-based regression estimators might be better. Note, that correct exploratory data analysis should identify such problems. If the choice of the explanatory variables is adequate one can hope that the kernel-based point estimates will be close to the classical regression estimate. The simulations and the case study discussed in section 5 and others in the literature, provide some evidence for that, as well as for the k-NN methods. For the variance estimate, the situation is more intricate and analytical derivations of the design-based variance are, so far, only available for $\hat{Y}_{reg}$.
Note that the kernel-based regression estimator \( \hat{Y}_e^{(1)}(x_0) \) is a convex linear combinations of the observations (i.e. a weighted average of the observations, where the weights are positive and sum up to 1). Hence, the range of the predictions is always contained in the range of the observations, which can be a disadvantage if the prediction model can be trusted. The estimator \( \hat{Y}_e^{(2)}(x_0) \) does not have this disadvantage. Other estimators using kernels but not in the same way as \( \hat{Y}_e^{(1)}(x_0) \) lead to non-convex linear combinations, such as the widely used R-functions \texttt{glkern} (with global optimal bandwidth) and \texttt{lokern} (with local optimal bandwidth) where the option ”derivative=0” is used. The resulting estimators are known to have good performances (see Jennen-Steinmetz and Gasser (1988) and Gasser and Mueller (1984) for details). In this work we have used the R-function \texttt{npreg} to calculate \( \hat{Y}_e^{(1)}(x_0) \) and the residual component of \( \hat{Y}_e^{(2)}(x_0) \). The global optimal bandwidth is obtained by the standard leave one out cross-validation technique, which can be calculated very efficiently for \( \hat{Y}_e^{(1)}(x_0) \) but not for \( \hat{Y}_e^{(2)}(x_0) \), an unfortunate development for variance estimation via bootstrap.

The overwhelming majority of kernel based non-parametric curve-fitting procedures are primarily concerned with interpolation, whereas in the forest inventory context with two-phase sampling the extrapolation aspects are also very important. In the case study below we will see that the mean-canopy height obtained by remote sensing (LiDAR) in the large sample \( s_1 \) is the most important explanatory variable, leading single-handedly to \( R^2 \) above 50%, and it clear that the highest values will most likely be in \( s_1 \) and not in \( s_2 \). Thus, forcing the prediction range to be contained in the observation range is a disadvantage. As afore mentioned, this is not the case for \( \hat{Y}_e^{(2)}(x_0) \).

4 Design-based k-NN

As for kernel-based estimators the literature on k-Nearest Neighbors (k-NN) is immense and largely model-dependent. In the forest inventory context Bafetta et al. (2009, 2011) are key references in the design-based context whereas McRoberts et al. (2007), McRoberts et al. (2011), Magnussen and Tomppo (2014), McRoberts (2012), Breidenbach and Nothdurft (2010) are more model-dependent. Also, most authors work in the finite population framework of pixels in which the first phase is exhaustive.

We shall use here the procedure described by Hechenbichler and Schliep (2004), which is implemented in the R program \texttt{kknn}. For an arbitrary point \( x_0 \in s_1 \) the nearest neighbor in \( s_2 \), with respect to a distance \( d(\cdot, \cdot) \) in \( \mathbb{R}^p \) is the point \( x_{(1)} \in s_2 \) such that \( d(Z(x_0), Z(x_{(1)})) = \min_{x \in s_2} d(Z(x_0), Z(x)) \). The second nearest neighbor \( x_{(2)} \) is defined by \( d(Z(x_0), Z(x_{(2)})) = \min_{x \in s_2 \setminus x_{(1)}} d(Z(x_0), Z(x)) \) and so on until we have obtained the \( k+1 \) nearest neighbors. We emphasize the fact that if \( x_0 \) is in \( s_2 \) then we set \( x_{(1)} = x_0 \). If the components of \( Z(x) \) have very different scales it is usually recommended to standardize the component so that they all have standard deviation equal to 1.

The simulations and case study presented in section 5 contain only continuous variables so that ties do not occur. The k-nn methods with the uniform kernel simply take the mean \( \frac{1}{k} \sum_{i=1}^{k} Y(x_{(i)}) = \hat{Y}_{knn}(x_0) \). In this case it is possible, in principle, to calculate the
theoretical variance of the corresponding regression estimator, albeit in the finite population case with exhaustive first phase (see Bafetta et al. (2009)). However, it is usually more efficient to give larger weights to the closest neighbors. To achieve that we have to standardize the distances according to

\[ D(Z(x_0), Z(x_i)) = \frac{d(Z(x_0), Z(x_i))}{d(Z(x_0), Z(x_{(k+1)}) \right) \]

and the weighted k-NN estimator is given by

\[ \hat{Y}_{knn}(x_0) = \frac{\sum_{i=1}^{k} K(D(Z(x_0), Z(x_i))) Y(x_i)}{\sum_{i=1}^{k} K(D(Z(x_0), Z(x_i)))} \]

for an arbitrary one-dimensional kernel \( K(\cdot) \) (Note that the kernel is one-dimensional whereas the distance is calculated in the multidimensional feature space). This leads to the k-NN regression estimator

\[ \hat{Y}_{knn} = \frac{1}{n_1} \sum_{x_0 \in s_1} \hat{Y}_{knn}(x_0) + \frac{1}{n_2} \sum_{x_0 \in s_2} (Y(x_0) - \hat{Y}_{knn}(x_0)) \]

Note that the predictions, \( \hat{Y}_{knn}(x_0) \), are a weighted mean of the observations and therefore constrained to be in the observed range of the observations. Thus k-NN is a purely interpolation technique and never extrapolates, which, as already mentioned, can be a disadvantage. As for kernel-based regression estimator one can use the one dimensional metric based on the predictions alone, i.e. \( d(Z(x_0), Z(x_i)) = |\hat{\beta}(Z(x_0) - Z(x_i))| \). The resulting estimator is denoted by \( \hat{Y}_{pred,knn} \).

The external variance estimates are obtained as usual via sample copies of [8]. In practice one has to be careful about the tuning options available in software packages, in particular with respect to the definition of nearest neighbors when \( x_0 \in s_2 \). If the point itself is viewed as its nearest neighbor the optimal choices \( k = 1 \) can result from cross-validation and should of course not be retained as it its leads to constant 0 residuals. We recommend at least \( k \geq 3 \) and to plot \( \hat{Y}_{knn} \) and \( \hat{V}(\hat{Y}_{knn}) \) as a function of \( k \). In our experience, except for \( \hat{Y}_{reg} \), the external variance estimate is to small and the bootstrap variance estimate should be preferred.
5 Examples

To illustrate the various techniques we consider the real case study discussed in Mandallaz et al. (2013) and the artificial simulation example used in Mandallaz (2013).

We have compared several kernel-based estimators calculated with the widely used R-functions `glkern`, `lokern` (both from the R-package `lokern`), `npreg` (from the R-package `np`) and the k-NN estimators calculated with the R-package `kknn`. In the main text we do not discuss the function `lokern` because it was found that the local optimal bandwidth led to very spurious results near the boundaries, particularly at the lower end (see e.g. Fig. 1 of Appendix E obtained from the case study).

The two-phase bootstrap procedure is very simple: first a bootstrap sample $s_{1\text{boot}}$ is generated from $s_1$ and the points $x \in s_{1\text{boot}}$ that are in $s_2$ define the the second-phase bootstrap sample $s_{2\text{boot}}$.

The bootstrap variances with optimal bandwidth recalculated in each bootstrap sample were very unstable and frequently led to extreme outliers (above 1000 times the mean), particularly for `lokern` as can be seen from Fig. 4 and 5 in the Appendix E based on the simulation example. This was also the case, but not as extreme, for the case study as can be seen from Tables 7 and 8 given in the Appendix D.

For $\hat{Y}_{\text{gkern}}^{(2)}$ the R-function `glkern` was applied to the residual part only and the optimal bandwidth was obtained via the imbedded cross-validation procedure. For $\hat{Y}_{\text{npreg}}^{(2)}$ the optimal bandwidth obtained via the imbedded cross-validation procedure and applied to only the residual part led essentially to a flat horizontal line close to 0 (i.e. making $\hat{Y}_{\text{npreg}}^{(2)} \approx \hat{Y}_{\text{reg}}$). For this reason we wrote our own very simple cross-validation procedure based on the mean squared sequentially deleting each $x_0 \in s_2$ and refitting both $\hat{Y}(x_0)$ using the linear model as well as the smoothed residual part (using `npreg` on the second term) to get the optimal bandwidth out of a small set of feasible values.

The resulting $\hat{Y}_{\text{npreg}}^{(2)}$ then differs slightly from $\hat{Y}_{\text{reg}}$. The same procedure was used for the simulation presented in section 5.2.

5.1 Case study

The auxiliary vector $Z(x) \in \mathbb{R}^p$ has $p = 7$ components: $Z_0(x) \equiv 1$, mean canopy height $Z_1(x)$, maximal canopy height $Z_2(x)$, 75% quantile of the canopy height, $Z_3(x)$, standard deviation of canopy height $Z_4(x)$, the LiDAR estimated volume density $Z_5(x)$ and the LiDAR estimated density of stems $Z_6(x)$. The sample sizes were $n_1 = 306, n_2 = 67$. The fitted model is

$$\hat{Y}(x) = 322.57 + 52.55Z_1(x) - 19.24Z_2(x) - 33.04Z_3(x) + 71.06Z_4(x) + 0.19Z_5(x) - 0.09Z_6$$

The coefficient of determination for this model is $R^2 = 0.64$.

For the estimator $\hat{Y}_{\text{knn}}$ we use the gaussian kernel with $k = 7$, which is the optimal value obtained from the leave-one-out cross-validation procedure. Fig. 2 in the Appendix E shows the relation between the variance and the parameter $k$ and justifies the choice $k = 7$. 

for $\hat{Y}_{knn}$. The corresponding value for the $\hat{Y}_{pred,knn}$, also based on the gaussian kernel, was found to be $k = 9$.

To stabilize the bootstrap variances, the optimal global bandwidth obtained from the original full sample was used in all the bootstrap samples, and likewise for the optimal number $k$ of neighbors. The regression coefficients, on the other hand, were re-calculated in each bootstrap sample. There are very slight differences in the point estimates $\hat{Y}_{reg}$ between the results presented here and those discussed in Mandallaz et al. (2013) because the boundary adjustments at the forest edge had to be simplified in order to reduce the computing time for the bootstrap calculations, based on 1000 replicates.

Table 1 displays the results for the entire domain.
Table 1: Global estimation for the case study in Canton Grisons

<table>
<thead>
<tr>
<th>Estimator</th>
<th>point estimate</th>
<th>external s.e.</th>
<th>bootstrap s.e.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{Y}_F$</td>
<td>399.43</td>
<td>23.82</td>
<td>23.90</td>
</tr>
<tr>
<td>$\hat{Y}_{reg}$</td>
<td>384.95</td>
<td>16.52</td>
<td>18.34</td>
</tr>
<tr>
<td>$\hat{Y}_{(1)}^{glkern}$</td>
<td>389.89</td>
<td>16.08</td>
<td>19.36</td>
</tr>
<tr>
<td>$\hat{Y}_{(1)}^{npreg}$</td>
<td>388.58</td>
<td>16.08</td>
<td>17.79</td>
</tr>
<tr>
<td>$\hat{Y}_{(2)}^{glkern}$</td>
<td>389.79</td>
<td>16.08</td>
<td>19.41</td>
</tr>
<tr>
<td>$\hat{Y}_{(2)}^{npreg}$</td>
<td>387.51</td>
<td>16.28</td>
<td>18.55</td>
</tr>
<tr>
<td>$\hat{Y}_{knn}$</td>
<td>390.53</td>
<td>16.35</td>
<td>17.83</td>
</tr>
<tr>
<td>$\hat{Y}_{pred,knn}$</td>
<td>389.22</td>
<td>15.69</td>
<td>17.42</td>
</tr>
</tbody>
</table>

Discussion:

The differences between the estimators are not statistically significant, which is reassuring. The bootstrap standard errors are roughly 15% larger than the external standard errors. For $\hat{Y}_{reg}$ the g-weight variance is 16.96 and closer to the bootstrap variance (see Mandallaz et al. (2013)). To calculate confidence intervals with the external variance one can use the normal distribution or the Student’s distribution on $n_2 - p$ degrees of freedom.

We now consider the small-area estimation problem: we have a partition of the entire domain $F$, in 4 small areas $G_k$, i.e. $F = \bigcup_{k=1}^{4} G_k$, with approximately equal surface areas $\lambda(G_k)$. We use the simple external model approach after restricting the samples to the small areas. The regression coefficients, predictions and nearest neighbors are always based on the entire domain $F$. For the bootstrap calculations the re-sampling takes place in $F$ and the samples are then restricted to the small areas $G \in F$. Table 2 displays the results.
Table 2: Small area estimation for the case study in Canton Grisons

<table>
<thead>
<tr>
<th>small area</th>
<th>$G_1$</th>
<th>$G_2$</th>
<th>$G_3$</th>
<th>$G_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_1 : n_2$</td>
<td>94:19</td>
<td>81:17</td>
<td>66:15</td>
<td>65:16</td>
</tr>
<tr>
<td>$\hat{Y}_G$</td>
<td>410.40</td>
<td>461.44</td>
<td>318.00</td>
<td>396.85</td>
</tr>
<tr>
<td>(44.58)</td>
<td>(56.35)</td>
<td>(34.36)</td>
<td>(47.86)</td>
<td></td>
</tr>
<tr>
<td>[44.17]</td>
<td>[55.37]</td>
<td>[33.93]</td>
<td>[47.88]</td>
<td></td>
</tr>
<tr>
<td>$\hat{Y}_{reg}$</td>
<td>397.27</td>
<td>426.81</td>
<td>327.64</td>
<td>366.44</td>
</tr>
<tr>
<td>(28.80)</td>
<td>(35.01)</td>
<td>(31.64)</td>
<td>(36.01)</td>
<td></td>
</tr>
<tr>
<td>[31.61]</td>
<td>[34.93]</td>
<td>[32.06]</td>
<td>[35.53]</td>
<td></td>
</tr>
<tr>
<td>$\hat{Y}^{(1)}_{gilkern}$</td>
<td>405.60</td>
<td>427.16</td>
<td>340.50</td>
<td>364.63</td>
</tr>
<tr>
<td>(28.82)</td>
<td>(34.63)</td>
<td>(29.65)</td>
<td>(34.65)</td>
<td></td>
</tr>
<tr>
<td>[34.40]</td>
<td>[34.30]</td>
<td>[30.80]</td>
<td>[34.76]</td>
<td></td>
</tr>
<tr>
<td>$\hat{Y}^{(1)}_{npreg}$</td>
<td>397.36</td>
<td>428.37</td>
<td>341.08</td>
<td>369.36</td>
</tr>
<tr>
<td>(28.65)</td>
<td>(35.11)</td>
<td>(29.17)</td>
<td>(34.76)</td>
<td></td>
</tr>
<tr>
<td>[30.72]</td>
<td>[33.27]</td>
<td>[29.21]</td>
<td>[33.67]</td>
<td></td>
</tr>
<tr>
<td>$\hat{Y}^{(2)}_{gilkern}$</td>
<td>405.46</td>
<td>426.94</td>
<td>340.58</td>
<td>364.59</td>
</tr>
<tr>
<td>(28.81)</td>
<td>(34.62)</td>
<td>(29.67)</td>
<td>(34.64)</td>
<td></td>
</tr>
<tr>
<td>[34.42]</td>
<td>[34.25]</td>
<td>[30.86]</td>
<td>[34.70]</td>
<td></td>
</tr>
<tr>
<td>$\hat{Y}^{(2)}_{npreg}$</td>
<td>400.43</td>
<td>430.39</td>
<td>334.59</td>
<td>363.70</td>
</tr>
<tr>
<td>(29.61)</td>
<td>(34.73)</td>
<td>(30.09)</td>
<td>(35.01)</td>
<td></td>
</tr>
<tr>
<td>[32.72]</td>
<td>[34.23]</td>
<td>[30.91]</td>
<td>[34.98]</td>
<td></td>
</tr>
<tr>
<td>$\hat{Y}_{knn}$</td>
<td>382.46</td>
<td>464.14</td>
<td>319.42</td>
<td>376.96</td>
</tr>
<tr>
<td>(27.07)</td>
<td>(36.64)</td>
<td>(32.59)</td>
<td>(32.14)</td>
<td></td>
</tr>
<tr>
<td>[28.28]</td>
<td>[34.77]</td>
<td>[28.04]</td>
<td>[30.02]</td>
<td></td>
</tr>
<tr>
<td>$\hat{Y}_{pred,knn}$</td>
<td>398.60</td>
<td>424.02</td>
<td>353.21</td>
<td>364.48</td>
</tr>
<tr>
<td>(28.87)</td>
<td>(34.09)</td>
<td>(29.96)</td>
<td>(32.14)</td>
<td></td>
</tr>
<tr>
<td>[29.36]</td>
<td>[32.42]</td>
<td>[28.41]</td>
<td>[30.62]</td>
<td></td>
</tr>
</tbody>
</table>

Legend:
The standard error based on the external model assumption are given in () and the standard errors based on the bootstrap variance estimates are given in [].
Discussion:

- All points estimators are close to each other and not significantly different.
- The external variance estimates and bootstrap variances of all estimators are comparable.
- The small-area point estimates in the extended model and their g-weight variance estimates (as discussed in Mandallaz et al. (2013)) are also very close to their counterparts in Table 2.
- One may eventually obtain better bootstrap results for the small areas by using a modified balanced replication as suggested in Magnussen et al. (2013). However, this was not the case here.
- To calculate confidence intervals with the external variance we recommend to use the Student’s distribution on \( n_{2,G} - 1 \) degrees of freedom rather than the normal distribution.

5.2 Simulations

To illustrate the theory and check empirically the validity of the various mathematical approximation used to derive the variance estimates we present simulations performed on a purely artificial example, already used in Mandallaz (2013). The local density \( Y(x) \) is defined according to the following procedure: at point \( x = (x_1, x_2)^t \in \mathbb{R}^2 \), the auxiliary vector is defined as \( Z(x) = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2)^t \in \mathbb{R}^p \) \( (p = 6 \) in this example). The true parameter is \( \beta_0 = (30, 13, -6, -4, 3, 2)^t \in \mathbb{R}^6 \) and the local density over the domain \( F = [0, 2] \times [0, 3] \) is given by the function

\[
Y(x) = Z^t(x)\beta_0 + 6 \cos(\pi x_1) \sin(2\pi x_2) := \hat{Y}_0(x) + R(x)
\]

Figure 3 in Appendix E displays the local density. \( \hat{Y}_{knn} \) is based on the Euclidian distance for \( Z(x) \). We fixed the number of neighbors to \( k = 3 \) because if the software is allowed to choose the optimal value of \( k \) (on average 2.7) we obtain \( k_{opt} = 1 \) in about 30% of the runs, which is totally absurd since this leads to constant zero residuals (essentially the best prediction at \( x_0 \in s_2 \) is then the true observation \( Y(x_0) \)). This corresponds roughly to setting the local bandwidth \( h(x_0) \) to zero for kernel-based estimators, which is not allowed because consistency requires \( n_2h(x_0) \to \infty \) (see the remarks at the end of Appendix B). Fig. 6 in Appendix E illustrates the fact that the external variance of \( \hat{Y}_{knn} \) underestimates the correct empirical variance, particularly in the range of interesting \( k \) values. For the estimator \( \hat{Y}_{pred,knn} \) the optimal values of \( k \) were obtained from restricted simulations with 500 runs for each scenario.

Tables 3 and 4 give the results for global estimation when the fitted model is indeed the true model and Tables 5 and 6 when the fitted model is a subset of the true model. All simulations are based on 10’000 runs. The 95% bootstrap confidence intervals are
obtained via the well-known formula \([2\hat{Y} - q_{0.975}^*, 2\hat{Y} - q_{0.025}^*]\), where \(q_{\alpha}^*\) is the \(\alpha\)-bootstrap quantile of the bootstrap estimate \(\hat{Y}^*\), defined as the mean of the point estimates of the 1000 bootstrap samples obtained in each of the 10'000 runs.
Table 3: $n_1 = 400, n_2 = 100$: fitted model is the true model

<table>
<thead>
<tr>
<th>Estimator</th>
<th>$E^*(\cdot)$</th>
<th>$V^*(\cdot)$</th>
<th>$E^*(V(\cdot))$</th>
<th>$E^*(\hat{V}_{\text{boot}}(\cdot))$</th>
<th>$E^*(P)$</th>
<th>$E^*(P_{\text{boot}})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{Y}_{\text{reg}}$</td>
<td>39.17</td>
<td>0.19</td>
<td>0.19</td>
<td>0.20</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>$\hat{Y}_{\text{glkern}}^{(1)}$</td>
<td>39.15</td>
<td>0.20</td>
<td>0.19</td>
<td>0.20</td>
<td>0.94</td>
<td>0.95</td>
</tr>
<tr>
<td>$\hat{Y}_{\text{npreg}}^{(1)}$</td>
<td>39.15</td>
<td>0.20</td>
<td>0.18</td>
<td>0.20</td>
<td>0.94</td>
<td>0.94</td>
</tr>
<tr>
<td>$\hat{Y}_{\text{glkern}}^{(2)}$</td>
<td>39.15</td>
<td>0.20</td>
<td>0.19</td>
<td>0.20</td>
<td>0.94</td>
<td>0.95</td>
</tr>
<tr>
<td>$\hat{Y}_{\text{npreg}}^{(2)}$</td>
<td>39.15</td>
<td>0.20</td>
<td>0.18</td>
<td>0.20</td>
<td>0.94</td>
<td>0.95</td>
</tr>
<tr>
<td>$\hat{Y}_{\text{pred,knn,k}=11}$</td>
<td>39.16</td>
<td>0.20</td>
<td>0.18</td>
<td>0.20</td>
<td>0.93</td>
<td>0.94</td>
</tr>
<tr>
<td>$\hat{Y}_{\text{knn,k}=3}$</td>
<td>39.15</td>
<td>0.18</td>
<td>0.14</td>
<td>0.19</td>
<td>0.92</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Table 4: $n_1 = 200, n_2 = 50$: fitted model is the true model

<table>
<thead>
<tr>
<th>Estimator</th>
<th>$E^*(\cdot)$</th>
<th>$V^*(\cdot)$</th>
<th>$E^*(V(\cdot))$</th>
<th>$E^*(\hat{V}_{\text{boot}}(\cdot))$</th>
<th>$E^*(P)$</th>
<th>$E^*(P_{\text{boot}})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{Y}_{\text{reg}}$</td>
<td>39.18</td>
<td>0.41</td>
<td>0.38</td>
<td>0.42</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>$\hat{Y}_{\text{glkern}}^{(1)}$</td>
<td>39.17</td>
<td>0.42</td>
<td>0.36</td>
<td>0.45</td>
<td>0.92</td>
<td>0.95</td>
</tr>
<tr>
<td>$\hat{Y}_{\text{npreg}}^{(1)}$</td>
<td>39.15</td>
<td>0.42</td>
<td>0.35</td>
<td>0.43</td>
<td>0.92</td>
<td>0.94</td>
</tr>
<tr>
<td>$\hat{Y}_{\text{glkern}}^{(2)}$</td>
<td>39.17</td>
<td>0.42</td>
<td>0.36</td>
<td>0.45</td>
<td>0.92</td>
<td>0.95</td>
</tr>
<tr>
<td>$\hat{Y}_{\text{npreg}}^{(2)}$</td>
<td>39.17</td>
<td>0.43</td>
<td>0.36</td>
<td>0.42</td>
<td>0.93</td>
<td>0.94</td>
</tr>
<tr>
<td>$\hat{Y}_{\text{pred,knn,k}=6}$</td>
<td>39.15</td>
<td>0.43</td>
<td>0.35</td>
<td>0.43</td>
<td>0.92</td>
<td>0.94</td>
</tr>
<tr>
<td>$\hat{Y}_{\text{knn,k}=3}$</td>
<td>39.15</td>
<td>0.43</td>
<td>0.31</td>
<td>0.43</td>
<td>0.91</td>
<td>0.94</td>
</tr>
</tbody>
</table>

Legend:
The true mean value is $\bar{Y} = 39.17$ and the true coefficient of determination is $R^2 = 0.83$.
$E^*(\hat{P})$ is the empirical coverage probability of the 95% confidence interval based on the estimated variance and the normal distribution.
$E^*(\hat{P}_{\text{boot}})$ is the empirical coverage probabilities based on the bootstrap confidence intervals. $E^*(\cdot)$ and $V^*(\cdot)$ denote the empirical mean and variances over 10’000 runs.
$\hat{V}(\hat{Y}_{\text{estimator}})$ is the estimated variance under the external model assumption.
$\hat{V}_{\text{boot}}(\cdot)$ is the variance obtained from 1000 bootstrap replicates.
Table 5: \( n_1 = 400, n_2 = 100 \): fitted model is not the true model

<table>
<thead>
<tr>
<th>Estimator</th>
<th>( E^*(\cdot) )</th>
<th>( V^*(\cdot) )</th>
<th>( E^*(V(\cdot)) )</th>
<th>( E^*(\hat{V}_{boot}(\cdot)) )</th>
<th>( E^*(P(\cdot)) )</th>
<th>( E^*(P_{boot}(\cdot)) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{Y}_{reg} )</td>
<td>39.18</td>
<td>0.28</td>
<td>0.26</td>
<td>0.27</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>( \hat{Y}_{(1)}^{glkern} )</td>
<td>39.19</td>
<td>0.27</td>
<td>0.25</td>
<td>0.28</td>
<td>0.93</td>
<td>0.94</td>
</tr>
<tr>
<td>( \hat{Y}_{(1)}^{npreg} )</td>
<td>39.18</td>
<td>0.28</td>
<td>0.24</td>
<td>0.27</td>
<td>0.92</td>
<td>0.94</td>
</tr>
<tr>
<td>( \hat{Y}_{(2)}^{glkern} )</td>
<td>39.19</td>
<td>0.27</td>
<td>0.25</td>
<td>0.28</td>
<td>0.93</td>
<td>0.94</td>
</tr>
<tr>
<td>( \hat{Y}_{(2)}^{npreg} )</td>
<td>39.18</td>
<td>0.27</td>
<td>0.25</td>
<td>0.27</td>
<td>0.93</td>
<td>0.94</td>
</tr>
<tr>
<td>( \hat{Y}_{knn,pred,k=15} )</td>
<td>39.17</td>
<td>0.28</td>
<td>0.24</td>
<td>0.27</td>
<td>0.93</td>
<td>0.94</td>
</tr>
<tr>
<td>( \hat{Y}_{knn,k=3} )</td>
<td>39.12</td>
<td>0.21</td>
<td>0.12</td>
<td>0.22</td>
<td>0.91</td>
<td>0.96</td>
</tr>
</tbody>
</table>

Table 6: \( n_1 = 200, n_2 = 50 \): fitted model is not the true model

<table>
<thead>
<tr>
<th>Estimator</th>
<th>( E^*(\cdot) )</th>
<th>( V^*(\cdot) )</th>
<th>( E^*(V(\cdot)) )</th>
<th>( E^*(\hat{V}_{boot}(\cdot)) )</th>
<th>( E^*(P(\cdot)) )</th>
<th>( E^*(P_{boot}(\cdot)) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{Y}_{reg} )</td>
<td>39.21</td>
<td>0.56</td>
<td>0.50</td>
<td>0.56</td>
<td>0.94</td>
<td>0.95</td>
</tr>
<tr>
<td>( \hat{Y}_{(1)}^{glkern} )</td>
<td>39.21</td>
<td>0.57</td>
<td>0.48</td>
<td>0.62</td>
<td>0.92</td>
<td>0.94</td>
</tr>
<tr>
<td>( \hat{Y}_{(1)}^{npreg} )</td>
<td>39.18</td>
<td>0.57</td>
<td>0.46</td>
<td>0.56</td>
<td>0.92</td>
<td>0.93</td>
</tr>
<tr>
<td>( \hat{Y}_{(2)}^{glkern} )</td>
<td>39.21</td>
<td>0.57</td>
<td>0.48</td>
<td>0.62</td>
<td>0.92</td>
<td>0.94</td>
</tr>
<tr>
<td>( \hat{Y}_{(2)}^{npreg} )</td>
<td>39.19</td>
<td>0.57</td>
<td>0.48</td>
<td>0.56</td>
<td>0.92</td>
<td>0.94</td>
</tr>
<tr>
<td>( \hat{Y}_{knn,pred,k=9} )</td>
<td>39.17</td>
<td>0.56</td>
<td>0.46</td>
<td>0.56</td>
<td>0.92</td>
<td>0.94</td>
</tr>
<tr>
<td>( \hat{Y}_{knn,k=3} )</td>
<td>39.12</td>
<td>0.50</td>
<td>0.32</td>
<td>0.51</td>
<td>0.89</td>
<td>0.94</td>
</tr>
</tbody>
</table>

Legend:
The columns headers are the same as in Tables 3 and 4. The true mean value is the same, i.e. \( \bar{Y} = 39.17 \). The fitted model did not include the explanatory variables \( x_1 \) and \( x_1 x_2 \). The true coefficient of determination decreases from \( R^2 = 0.83 \) to \( R^2 = 0.66 \).
Discussion:

1. All estimators are practically design-unbiased, whether the fitted model is true or not, which is the main advantage of the design-based approach. The empirical (i.e. 'true') variances are smaller when the fitted model is the true model, as expected.

2. The external variances always underestimate the empirical, i.e. true, variance. The underestimation can be substantial, except for $\hat{Y}_{reg}$.

3. The bootstrap variance estimates are close to the empirical variances.

4. With respect to the nominal 95% confidence intervals the classical regression estimator $\hat{Y}_{reg}$ performs very well in all investigated cases, and can be even further improved by using the g-weight variance instead of the external variance (Mandallaz (2013)). For all other estimators one must use the bootstrap confidence intervals because their external variances clearly underestimates the empirical variances. Slightly higher coverage probabilities can be achieved with the external variance by using the Student’s t-distribution on $n_2 - p$ degrees of freedom, which is recommended for small $n_2$, particularly for small area estimation.

5. On the whole the sophisticated kernel-based or $knn$ estimators do not perform better than the classical regression estimator $\hat{Y}_{reg}$, which is disappointing but worth knowing. The nearest neighbor estimator $\hat{Y}_{knn}$ with the optimal choice $k = 3$ yields a slightly smaller empirical variance (correctly estimated with the bootstrap) when the fitted model is not the true model.

6. The excellent performances of the classical regression estimator $\hat{Y}_{reg}$ is probably due to the fact that the coefficients of determination are rather high under the true and under the working models, even if the value $R^2 = 0.66$ in the later case is not outstanding in the context e.g. of forest inventories with timber volume as response variable and with LiDAR canopy measurements as explanatory variables.

7. Great care is needed with respect to the tuning options and the implementation of the bootstrap procedures.

8. A further theoretical advantage of $\hat{Y}_{reg}$ is that it allows for correct analytical expressions of the asymptotic variances, which seems to be very difficult to obtain for the other estimators.

6 Conclusions

These preliminary comparisons between the various estimators show that the simple classical two-phase regression estimator $\hat{Y}_{reg}$ with the external or g-weight variance estimates performs on the whole at least as well as the more sophisticated kernel-based or $knn$ regression estimators with the bootstrap variance estimates. Also, the choice of the tuning options and re-sampling procedures of the later estimators is not trivial. The $knn$
estimators share with some kernel-based estimators the disadvantage that they cannot extrapolate beyond the range of the observations, which is not the case for $\hat{Y}_{reg}$. The resulting coverage probabilities of the confidence intervals based on the bootstrap are satisfactory, likewise for the simpler external and g-weight variance estimates of $\hat{Y}_{reg}$. A definite advantage of the regression estimator $\hat{Y}_{reg}$ is that it is also available for more complex situations such as three-phase cluster sampling with two-types of auxiliary information, or with two-stage sampling of trees at the plot level, as well as their extension to small-area estimation (Mandallaz (2013); Mandallaz et al. (2013); Mandallaz (2014)). It can reasonably be expected that these conclusions will hold in general if the model is adequate, i.e. if it incorporates the most important explanatory variables (say with an $R^2 \geq 0.6$), which is certainly the case for timber volume with the standard LiDAR explanatory variable (mean canopy height being the most important).

In this context the main advantage of $\hat{Y}_{reg}$ is its simplicity and the uniqueness of the results, independently of the choice of kernels, metrics and further tuning options. It might be wise to use also $\hat{Y}_{knn}$, with well chosen options and bootstrap variance, to have a second opinion.
References


A The Nadaraya-Watson regression estimator

This estimator goes back to Nadaraya (1965), but the original idea is even older, going back to Rosenblatt (1956) and Parzen (1962). The third kernel given in [10] was suggested by Epanechnikov (1969). There is an immense literature on the subject of non-parametric kernel based estimation (for densities as well as for regression) in the mainstream model-dependent literature, but not so many in the design-based approach to survey sampling, and even less so in forest inventory.

In this appendix we summarize the main ideas and results and give informal proofs without going into the details of the regularity assumptions.

In the classical model-dependent approach to non-parametric kernel-based regression (see e.g. Györfi et al. (2002)) one considers \( n \) i.i.d random vectors \((X_i, Y_i)\) with bivariate joint density \(p(x, y)\) and marginal density \(g(x)\) for the \(X_i\). One wants to estimate the conditional expectation \(m(x_0) = \mathbb{E}(Y \mid X = x_0)\), which is the best predictor of \(Y\) given \(X = x_0\). The Nadaraya-Watson estimate of the marginal density \(g(x)\) at the point \(x_0\) is defined by

\[
\hat{g}_\varepsilon(x_0) = \frac{1}{n_2 \varepsilon(n_2, x_0)} \sum_{i=1}^{n_2} K \left( \frac{x_0 - X_i}{\varepsilon(n_2, x_0)} \right)
\]

Under the conditions \( \lim_{n_2 \to \infty} \varepsilon(n_2, x_0) = 0 \) and \( \lim_{n_2 \to \infty} n_2 \varepsilon(n_2, x_0) = \infty \) it is shown below that \( \hat{g}_\varepsilon(x_0) \) is a consistent estimate (convergence in probability since the variance decreases as \( \frac{1}{n_2 \varepsilon(n_2, x_0)} \)). Under the condition \( \lim_{n_2 \to \infty} n_2 \varepsilon^2(n_2, x_0) = \infty \) one has also uniform convergence in probability, i.e. \( \lim_{n_2 \to \infty} \mathbb{P}(\sup_{x \mid \hat{g}_\varepsilon(x_0) - g(x_0) > \eta}) = 0 \ \forall \eta > 0 \), and even almost surely under further regularity conditions.

From a practical point of view we have good consistent estimates of the density functions, albeit in large samples. With respect to the mean square error \( \mathbb{E}_x(\hat{g}_\varepsilon(x_0) - g(x_0))^2 \) the Epanechnikov kernel is optimal and the optimal bandwidth should go to zero as \( n_2^{\frac{1}{5}} \) so that squared bias and variance are of order \( n_2^{\frac{2}{5}} \) and not as usual \( n_2^{-1} \) (see Lehmann (1999), p. 413).

The Nadaraya-Watson estimate of the bivariate density \(p(x, y)\) with the bivariate product kernel \(K(u, v) = K_1(u)K_2(u)\) (\(K_1(u)\) and \(K_2(v)\) are arbitrary kernels, e.g. such as those given in [10]) at the point \((x, y)\) is defined as

\[
\hat{p}_\varepsilon(x, y) = \frac{1}{n_2 \varepsilon_X(n_2, x) \varepsilon_Y(n_2, y)} \sum_{i=1}^{n_2} K_1 \left( \frac{x - X_i}{\varepsilon_X(n_2, x)} \right) K_2 \left( \frac{y - Y_i}{\varepsilon_Y(n_2, y)} \right)
\]

One has consistency under the conditions

\[
\lim_{n_2 \to \infty} \varepsilon_X(n_2, x) = 0, \quad \lim_{n_2 \to \infty} \varepsilon_Y(n_2, y) = 0
\]

and

\[
\lim_{n_2 \to \infty} n_2 \varepsilon_X(n_2, x) \varepsilon_Y(n_2, y) = \infty
\]
It is shown in Appendix B that one has
\[ \int \int \frac{p(x,y)}{g_{\epsilon}(x)} \, dy \]
and using this integration is feasible only with a constant bandwidth \( \varepsilon \), suffices to calculate the integral
\[ \int y \hat{p}_{Y|X}(y, x_0) \, dy \]
which is the celebrated Nadaraya-Watson non-parametric regression estimator.

**Remark**
For \( p \)-dimensional densities the the squared bias and variance term (with the optimal bandwidths) tend to 0 at the rate \( n_2^{-\frac{1}{d+2}} \), which can be viewed as the curse of dimensionality (see Lehmann (1999), p.419-420, for a dramatic example). This is a further motivation for minimizing the dimension of the feature space, i.e. \( p = 1 \), by choosing the metric based on the \( \hat{Y}(x) \) alone.

We now proceed to calculate the overall expectation and variance of [26], i.e. under both random mechanisms generating the \( X_i \) and \( Y_i \). For the expectation we get
\[ \mathbb{E}_{X_i, Y_i}(\hat{m}(x_0)) = \mathbb{E}_{X_i, Y_i} \int \hat{p}_{Y|X}(y, x_0) \, dy = \mathbb{E}_{X_i} \left( \frac{1}{n_{2}(n_2, x_0)} \sum_{i=1}^{n_2} Y_i K \left( \frac{x_0 - X_i}{\varepsilon(n_2, x_0)} \right) \right) \]
It is shown in Appendix B that one has
\[ \mathbb{E}(\hat{m}(x_0)) = m(x_0) + O(\varepsilon^2(n_2, x_0)) \]

For the variance we use the decomposition
\[ \mathbb{V}_{X_i, Y_i}(\hat{m}(x_0)) = \mathbb{V}_{X_i, Y_i} \int \hat{p}_{Y|X}(y, x_0) \, dy + \mathbb{E}_{X_i} \mathbb{V}_{Y_i|X_i}(\hat{m}(x_0)) \]
With \( m(x) = \mathbb{E}(Y | X = x) \) and \( \sigma^2(x) = \mathbb{V}(Y | X = x) \) we see that the first term can be written as
\[ \mathbb{V}_{X_i, Y_i} \int \hat{p}_{Y|X}(y, x_0) \, dy = \mathbb{V}_{X_i} \left( \frac{1}{n_{2}(n_2, x_0)} \sum_{x \in s_2} m(x) K \left( \frac{x_0 - x}{\varepsilon(n_2, x_0)} \right) \right) \]
and the second as

\[ [31] \quad \mathbb{E}_x \left( \frac{1}{n_2 \varepsilon(n_2, x_0)} \sum_{x \in s_2} \sigma^2(x) K^2 \left( \frac{x_0 - x}{\varepsilon(n_2, x_0)} \right) \right) \]

The denominator in [31] tends to \( g^2(x_0) \) and it is shown below that the numerator is asymptotically equivalent to

\[ \frac{1}{n_2 \varepsilon(n_2, x_0)} \sigma^2(x_0) g(x_0) \int_{\mathbb{R}} K^2(u) du \]

so that we finally have under the conditions

\[ \lim_{n_2 \to \infty} \varepsilon(n_2, x_0) = 0 \quad \text{and} \quad \lim_{n_2 \to \infty} n_2 \varepsilon(n_2, x_0) = \infty \]

\[ \mathbb{E}_{Y_i, X_i}(\hat{m}(x_0)) = m(x_0) + O(\varepsilon^2(n_2, x_0)) \]

\[ [32] \quad \forall_{Y_i, X_i}(\hat{m}(x_0)) = \frac{1}{n_2 \varepsilon(n_2, x_0)} \sigma^2(x_0) g(x_0) \int_{\mathbb{R}} K^2(u) du + O \left( \frac{\varepsilon(n_2, x_0)}{n_2} \right) \]

See also Glad (1998) for more details and further developments.

In the next section we shall analyze in more details the case when the \( Y_i \) in the original Nadaraya-Watson estimator [26] are replaced by a deterministic function \( f(x_i) \) at random locations \( X_i = x_i \) with probability density function \( g(x) \), which is sometimes called the random design context.

**B Kernel estimators in the design-based approach**

In the design-based approach and we consider instead the following kernel estimator of a deterministic function \( f : x \in \mathbb{R} \to f(x) \in \mathbb{R} \) at a point \( x_0 \)

\[ [33] \quad \hat{f}_\varepsilon(x_0) = \frac{1}{n_2 \varepsilon(n_2, x_0)} \sum_{x \in s_2} f(x) K \left( \frac{x_0 - x}{\varepsilon(n_2, x_0)} \right) \]

where \( x_0 \) is given and the \( n_2 \) points \( x \in s_2 \) are i.i.d on \( \mathbb{R} \) with density distribution function \( g(x) \) satisfying some regularity conditions (e.g. twice differentiable). \( f(x) \) is known for all \( x \in s_2 \). The expected value of the numerator (under the random location of the observations at points \( x \in s_2 \)) is

\[ \mathbb{E}_x(\hat{\varepsilon}(x_0)) = \frac{1}{\varepsilon(n_2, x_0)} \int_{\mathbb{R}} f(x) K \left( \frac{x_0 - x}{\varepsilon(n_2, x_0)} \right) g(x) dx \]

Changing the integration variable to \( u := \frac{x_0 - x}{\varepsilon(n_2, x_0)} \) we get

\[ \mathbb{E}_x(\hat{\varepsilon}(x_0)) = \int_{\mathbb{R}} h(x_0 - \varepsilon(n_2, x_0)u) K(u) du \]

24
where \( h(x) = f(x)g(x) \). The gist of many arguments (see Lehmann (1999), section 6.4 for details) is to use the Taylor expansion of \( h(x) \) at \( x_0 \). Because by assumption \( \int_{\mathbb{R}} K(u)du = 1, \int_{\mathbb{R}} uK(u)du = 0, \int_{\mathbb{R}} u^2 K(u)du < \infty \), we obtain the following result for the expected value

\[
\mathbb{E}_x(\hat{e}_x(x_0)) = h(x_0) = f(x_0)g(x_0) + O(\varepsilon^2(n_2, x_0))
\]

With similar arguments one obtains for the variance

\[
\mathbb{V}_x(\hat{e}_x(x_0)) = \frac{1}{n_2 \varepsilon(n_2, x_0)} f^2(x_0)g(x_0) \int_{\mathbb{R}} K^2(u)du - \frac{1}{n_2} f^2(x_0)g^2(x_0) + O\left(\frac{\varepsilon(n_2, x_0)}{n_2}\right)
\]

The Nadaraya-Watson estimator of the probability density function \( g(x) \) at the point \( x_0 \) is by definition

\[
\hat{g}_x(x_0) = \frac{1}{n_2 \varepsilon(n_2, x_0)} \sum_{x \in s_2} K\left(\frac{x_0 - x}{\varepsilon(n_2, x_0)}\right)
\]

Using the previous result in the special case \( f(x) \equiv 1 \) we obtain

\[
\mathbb{E}(\hat{g}_x(x_0)) = g(x_0) + O(\varepsilon^2(n_2, x_0))
\]

\[
\mathbb{V}_x(\hat{g}_x(x_0)) = \frac{g(x_0)}{n_2 \varepsilon(n_2, x_0)} \int_{\mathbb{R}} K^2(u)du - \frac{1}{n_2} g^2(x_0) + O\left(\frac{\varepsilon(n_2, x_0)}{n_2}\right)
\]

Hence, we have consistent estimates of \( f(x_0)g(x_0) \). \( g(x_0) \) and therefore of \( f(x_0) = \frac{f(x_0)g(x_0)}{g(x_0)} \) provided that \( \lim_{n_2 \to \infty} \varepsilon(n_2, x_0) = 0 \) and \( \lim_{n_2 \to \infty} n_2 \varepsilon(n_2, x_0) = \infty \).

Recall that the band-width must go to zero but not so fast in order to ensure \( \lim_{n_2 \to \infty} n_2 \varepsilon(n_2, x_0) = \infty \).

With the same techniques and after some tedious but simple algebra one obtains for the covariance

\[
\mathbb{COV}(\hat{e}_x(x_0), \hat{g}_x(x_0)) = \frac{f(x_0)g(x_0)}{n_2 \varepsilon(n_2, x_0)} \int_{\mathbb{R}} K^2(u)du - \frac{1}{n_2} f(x_0)g^2(x_0) + O\left(\frac{\varepsilon(n_2, x_0)}{n_2}\right)
\]

Using the well-known approximation formula for the variance of a ratio of two random variables \( R \) and \( S \), we get

\[
\mathbb{V}\left(\frac{R}{S}\right) \approx \frac{\mathbb{E}^2(R)}{\mathbb{E}^2(S)}\left(\frac{\mathbb{V}(R)}{\mathbb{E}^2(R)} - 2\frac{\mathbb{COV}(R, S)}{\mathbb{E}(R)\mathbb{E}(S)} + \frac{\mathbb{V}(S)}{\mathbb{E}^2(S)}\right)
\]

and after some algebra we see that the Nadaraya-Watson non-parametric regression estimator

\[
\hat{f}_x(x_0) = \frac{1}{n_2 \varepsilon(n_2, x_0)} \sum_{x \in s_2} f(x)K\left(\frac{x_0 - x}{\varepsilon(n_2, x_0)}\right)
\]

\[
\frac{1}{n_2 \varepsilon(n_2, x_0)} \sum_{x \in s_2} K\left(\frac{x_0 - x}{\varepsilon(n_2, x_0)}\right)
\]
under the conditions
\[
\lim_{n_2 \to \infty} \varepsilon(n_2, x_0) = 0 \quad \lim_{n_2 \to \infty} n_2 \varepsilon(n_2, x_0) = \infty
\]
is a consistent estimator of \( f(x_0) \) satisfying
\[
\mathbb{E}(\hat{f}_x(x_0)) = f(x_0) + O(\varepsilon^2(n_2, x_0))
\]
\[
\mathbb{V}_x(\hat{f}_x(x_0)) = O\left(\frac{\varepsilon(n_2, x_0)}{n_2}\right)
\]
[41]

It is interesting to note that the terms in \( \frac{1}{n_2(\varepsilon(n_2, x_0))} \) and \( \frac{1}{n_2} \) occurring in [37] and [38] cancel out and that the design-based variance goes to zero as \( O\left(\frac{\varepsilon(n_2, x_0)}{n_2}\right) \), whereas in the model-dependent approaches the term of order \( \frac{1}{n_2(\varepsilon(n_2, x_0))} \) remains as in [32] above. The above design-based result [41] corresponds formally, as expected, to the model-dependent case [32] with \( \sigma^2(x_0) = 0 \).

In the forest inventory context we estimate the probability density function of the predictions with
\[
\hat{f}_Y(\hat{y}_0) = \frac{1}{n_2 \varepsilon(n_2, x_0)} \sum_{x \in s_2} K\left(\frac{\hat{Y}(x_0) - \hat{Y}(x)}{\epsilon(n_2, x_0)}\right)
\]
We see from the above arguments, after the substitution \( x \to \hat{Y}(x) \) and \( dx \to f_Y(y)dy \), that \( \hat{f}_Y(\hat{y}_0) \) is an asymptotically design unbiased estimate of the density function \( f_Y(y) \) at the point \( y_0 = \hat{Y}(x_0) \). Likewise, we see that
\[
\frac{1}{n_2 \varepsilon(n_2, x_0)} \sum_{x \in s_2} \hat{Y}(x)K\left(\frac{\hat{Y}(x_0) - \hat{Y}(x)}{\epsilon(n_2, x_0)}\right)
\]
converges towards \( \hat{y}_0 f_Y(y_0) \) (as usual convergence in the mean, in probability or almost surely depending on the regularity assumptions). Likewise, we have also shown that
\[
\hat{Y}_c(x_0) := \frac{1}{n_2 \varepsilon(n_2, x_0)} \sum_{x \in s_2} \hat{Y}(x)K\left(\frac{\hat{Y}(x_0) - \hat{Y}(x)}{\epsilon(n_2, x_0)}\right)
\]
will converge towards \( \hat{Y}(x_0) \).

However, we are here primarily interested in estimators of the form
\[
\hat{Y}^{(1)}_c(x_0) := \frac{1}{n_2 \varepsilon(n_2, x_0)} \sum_{x \in s_2} Y(x)K\left(\frac{\hat{Y}(x_0) - \hat{Y}(x)}{\epsilon(n_2, x_0)}\right)
\]
[43]
and
\[
\hat{R}^{(2)}_c(x_0) := \frac{1}{n_2 \varepsilon(n_2, x_0)} \sum_{x \in s_2} \hat{R}(x)K\left(\frac{\hat{Y}(x_0) - \hat{Y}(x)}{\epsilon(n_2, x_0)}\right)
\]
[44]
This is only meaningful if we can write the local density $Y(x)$ as a function of its prediction $\hat{Y}(x)$, i.e. $Y(x) = \hat{Y}(\hat{Y}(x))$ (and likewise $R(x) = \hat{R}(\hat{Y}(x))$). This is not always possible, e.g. if the $\{\hat{Y}(x) \mid x \in F\}$ is a discrete set of points as with post-stratification, where for one $\hat{Y}(x)$ one has infinitely many $Y(x)$. In the continuous case the exact regression line (i.e. with the true optimal $\beta$) of $Y(x)$ (ordinate) on $\hat{Y}(x)$ (abscissa) is the diagonal (because $Y(x) = \hat{Y}(x) + R(x)$ and to each $\hat{Y}(x)$ one has one $Y(x)$, with probability 1). Neglecting the correlation between predictions (or using an independent training data set to fit the model) the mean and variance of $\hat{Y}^{(1)}(x_0)$ will be given by [41]. The convergence seems to be fast, but recall that the recommended optimal bandwidth is of order $n^{-\frac{1}{2}}$.

**Remark:**
Since $\hat{Y}^{(1)}(x_0)$ is asymptotically unbiased for $Y(x_0)$ it is tempting to consider the following estimator

\[ \hat{Y}^{(1)}_{e,reg} = \frac{1}{n_1} \sum_{x_0 \in s_1} \hat{Y}^{(1)}_e(x_0) \]

The covariance $\mathbb{C}O\mathbb{V}(\hat{Y}^{(1)}_e(x_1), \hat{Y}^{(1)}_e(x_2))$ is of order $O\left(\frac{\epsilon_{(n_2)}}{n_2}\right)$, so one may hope that the standard variance estimate

\[ \frac{1}{n_1(n_1 - 1)} \sum_{x_0 \in s_1} (\hat{Y}^{(1)}_e(x_0) - \bar{\hat{Y}}^{(1)}_e)^2 \]

is consistent. Let us denote with $R^2_{(1)}$ the R-squared of the kernel-based predictions $\hat{Y}^{(1)}_e(x_0)$ (regression line of the $Y(x_0)$ on the $\hat{Y}^{(1)}_e(x_0)$). The variance under the external model-approach is (see e.g. Mandallaz (2008), eqn. 5.4)

\[ \mathbb{V}(\hat{Y}^{(1)}_{e,reg}) = \mathbb{V}_x(Y(x)) \left( \frac{1 - R^2_{(1)}}{n_2} + \frac{R^2_{(1)}}{n_1} \right) \]

whereas the above proposal yields

\[ \mathbb{V}(\hat{Y}^{(1)}_{e,reg}) = \mathbb{V}_x(Y(x)) \frac{R^2_{(1)}}{n_1} \]

The same difficulty occurs with the classical two-phase regression estimator. In this case, the mean of the empirical residuals is zero and the empirical design-based covariance between predictions and residuals is zero by the orthogonality properties of ordinary least squares, so that one could naively think that the estimated variance is

\[ \hat{\mathbb{V}}(\hat{Y}_{reg}) = \frac{1}{n_1} \hat{\mathbb{V}}_x(\hat{Y}) \]

However, $\hat{Y}_{reg} = \hat{Z}^{s_1}_s \hat{\beta}_{s_2}$ and one obtains, using Taylor expansion and the estimated covariance matrix $\Sigma_{\hat{\beta}_{s_2}}$ given in [7], the correct answer, i.e. the g-weight based variance

\[ \hat{\mathbb{V}}(\hat{Y}_{reg}) = \frac{1}{n_1} \hat{\mathbb{V}}_x(\hat{Y}) + \hat{\beta}_{s_2}' \Sigma_{\hat{\beta}_{s_2}} \hat{\beta}_{s_2}^t \]
which can be shown to be asymptotically equivalent (see Mandallaz (2013) for examples and Mandallaz (2012) for detailed proofs) to the classical external variance estimate

\[
\hat{V}(\hat{Y}_{reg}) = \frac{1}{n_1} \hat{V}_x(\hat{Y}(x)) + \frac{1}{n_2} \hat{V}_x(\hat{R}(x))
\]

[50]

\[
= \frac{1}{n_1} \hat{V}_x(Y(x)) + (1 - \frac{n_2}{n_1}) \frac{1}{n_2} \hat{V}_x(\hat{R}(x))
\]

The second version can be expected to be better for kernel estimators since predictions and residuals do not have an exact zero design-based empirical covariance. Obviously, the same holds for the other estimators. Intuitively the result is obvious: there is no free lunch; one cannot expect to have with \( n_2 << n_1 \) costly terrestrial observations \( Y(x) \) (plus the comparatively low cost of the remote sensing data) the same accuracy as with \( n_1 \).
C Calibration model

C.1 Background

In the recent literature on finite population surveys, the concepts of calibration (calage in French, see Tille (2001)) and balanced sampling have played, and still play an important role. The former does not deal with the given inclusion probabilities, i.e., the design, but determines weights \( w_i \) in estimators of the type \( \sum_{i \in s} w_i Y_i \) such that the total distance \( \sum_{i=1}^{N} d(w_i, d_i) \) to the original weights \( d_i = \pi_i^{-1} \), with respect to some metric \( d(\cdot, \cdot) \), is minimized under the constraint for the auxiliary variables \( \sum_{i=1}^{N} Z_i = \sum_{i \in s} w_i Z_i \). As the original weights (i.e. \( d_i = \pi_i^{-1} \)) lead to the Horwitz-Thompson estimators (HT), the bias of the resulting estimator, i.e., calibration estimator, is small because it is close to HT. It can be expected to have good properties because it is exact for the auxiliary variables, assumed to be strongly related to the response variable. It can be shown that the calibration estimator is essentially asymptotically equivalent to linear regression estimators (i.e., linear prediction models, see Deville and Sarndal (1992)). In balanced sampling, on the other hand, one constructs a design with given inclusion probabilities \( \pi_i \) such that the constraint \( \sum_{i \in s} \pi_i / \sum_{i=1}^{N} \pi_i = \sum_{i \in s} w_i Z_i \) is satisfied, which can be done by the so-called cube algorithm (Tille (2006)). Hence, the two techniques achieve the same objective, calibration via the choice of the estimator and balanced sampling via the choice of the design.

The paper Wu and Sitter (2001) is an important contribution as it generalizes the original calibration techniques to non-linear and generalized linear models used for the predictions. The inference is design-based but the construction of the calibration estimators is based on considerations coming from a super-population.

C.2 Calibration estimators in the Monte-Carlo approach

So far, the calibration estimators are only available in the finite population framework with exhaustive auxiliary information. We shall give below an informal proof of the main result given in Wu and Sitter (2001) in the infinite population framework germane to the Monte-Carlo approach. One formal difficulty is that the inclusion probabilities of the sample points \( s_2 \subset F \) are constant equal to 0.

We shall consider non-linear models of the form

\[
Y(x) = \mu(Z(x), \beta) + R(x) =: \mu(x, \beta) + R(x)
\]

or generalized linear models \( g(\mu(x, \beta)) = Z'(x) \beta \) with a given link function \( g(\cdot) \). The ordinary non-linear least square uses the identity function \( g(u) = u \). The estimating equation [7] in Wu and Sitter (2001) (with the choice \( q_i \equiv 1 \)) reads

\[
\int_F Z(x) \phi(\mu(x, \beta)) v^{-1}(\mu(x, \beta) (Y(x) - \mu(x, \beta)) = 0
\]
with $\phi(u) = \left( \frac{d\phi(u)}{du} \right)^{-1}$ and an arbitrary weight function $v(\cdot)$. $\hat{\beta}$ can be obtained via sample copy version.

In non-linear least squares one considers the function

\[ Q(\beta) = \frac{1}{\lambda(F)} \int_F (Y(x) - \mu(Z(x), \beta))^2 dx \]

achieving is absolute minimum at $\beta_0$ and its sample copy version

\[ Q_n(\beta) = \frac{1}{n} \sum_{x \in s_2} (Y(x) - \mu(Z(x), \beta))^2 \]

$Q_n(\beta)$ is a random function of $\beta$, with expected value equal to $Q(\beta)$ and finite variance, which by the Law of Large Numbers converges in probability and also almost surely pointwise to the non-random function $Q(\beta)$. Under further regularity conditions the convergence will also be uniform in $\beta$ towards the function $Q(\beta)$. Ideally the argument $\hat{\beta}_s$ at the minimum of the random function $Q_n(\beta)$ should also converge to the argument at the minimum of $Q(\beta)$, which is by definition the true value $\beta_0$. This will be the case if the convergence is uniform in a neighborhood $U_0$ of $\beta_0$. We can also reasonably assume that the functions $Q_n(\beta)$ and $Q(\beta)$ are twice differentiable in $U_0$. The derivatives (gradients in the multidimensional case) of $Q_n(\beta)$ at $\hat{\beta}_s$ and of $Q(\beta)$ at $\beta_0$ both vanish because they are calculated at the respective minima. Using the second order Taylor expansion at both minima we get (with 1-d notation) the asymptotic relation

\[ Q_n(\beta) - Q(\beta) = Q_n(\hat{\beta}_s) - Q(\beta_0) + \frac{(\beta - \hat{\beta}_s)^2}{2} Q''(\hat{\beta}_s) - \frac{(\beta - \beta_0)^2}{2} Q''(\beta_0) \]

Since the term on the left of [54] tends to zero almost surely, we see that for $\beta \to \beta_0$ the above expansion leads to a contradiction if $\hat{\beta}_s$ does not tend to $\beta_0$ almost surely (note that the second derivatives at the minima are strictly positive, or positive definite matrices). Also, by Markov’s inequality, the left term in [54] is of order $\frac{1}{n_2}$. Hence, as $\beta \to \beta_0$, we must have on the right hand side the same order of magnitude for the term $(\hat{\beta}_s - \beta_0)^2$. This implies that $\hat{\beta}_s - \beta_0$ is of order $\frac{1}{\sqrt{n_2}}$ in probability, which is precisely the assumption made in the appendix of Wu and Sitter (2001). Consequently, one has

\[ \frac{1}{n_1} \sum_{x \in s_1} \hat{Y}(x, \hat{\beta}_s) = \frac{1}{n_1} \sum_{x \in s_1} \hat{Y}(x, \beta_0) + O_p\left( \frac{1}{\sqrt{n_2}} \right) \]

where $\hat{Y}(x, \beta) = \mu(Z(x), \beta)$. Likewise, one has for the residuals $\hat{R}(x, \beta) = Y(x) - \hat{Y}(x, \beta)$

\[ \frac{1}{n_2} \sum_{x \in s_2} \hat{R}(x, \hat{\beta}_s) = \frac{1}{n_2} \sum_{x \in s_2} \hat{R}(x, \beta_0) + O_p\left( \frac{1}{\sqrt{n_2}} \right) \]
Hence, the external model assumption (i.e. that the predictions and residuals at a sample point \(x\) depend only on the auxiliary information \(Z(x)\) at point \(x\) and not on all points in the sample \(s_2\)) holds asymptotically also for non-linear predictions. Note that formal proofs of similar results for finite populations are more intricate because both the population and the sample sizes have to tend to infinity. In the Monte-Carlo approach the population is infinite from scratch.

We can obtain formally the results given in Wu and Sitter (2001) (specialized to the case of equal inclusions probabilities \(\pi_i\) and weights \(q_i\)) by adapting the notation according to the following rules

\[
\begin{align*}
N &\rightarrow \lambda(F) \\
Y_i &\rightarrow Y(x) \\
\sum_{i=1}^{N} Y_i &\rightarrow \int_F Y(x) dx \\
\pi_i &\equiv \pi \quad \rightarrow \quad \pi(x) \equiv \frac{\lambda(F)}{n_2} \\
d_i = \pi_i^{-1} &\equiv \pi^{-1} \quad \rightarrow \quad d(x) \equiv \pi(x)^{-1} = \frac{\lambda(F)}{n_2} \\
w_i &\rightarrow w(x) \\
q_i &\equiv q \quad \rightarrow \quad q(x) \equiv q \\
\sum_{i \in s} w_i = N &\rightarrow \quad \sum_{x \in s_2} w(x) = \lambda(F) \\
\sum_{i \in s} \mu(x_i, \hat{\beta}) = \sum_{i=1}^{N} \mu(x_i, \hat{\beta}) &\rightarrow \quad \sum_{x \in s_2} w(x) \hat{Y}(x) = \int_F \hat{Y}(x) dx \\
\min \sum_{i \in s} \frac{(d_i - w_i)^2}{q_i d_i} &\rightarrow \quad \min \sum_{x \in s_2} (w(x) - d(x))^2 \\
\hat{Y}_{HT} &\rightarrow \quad \frac{1}{n_2} \sum_{x \in s_2} Y(x) \\
\hat{B}_N &\rightarrow \quad \frac{1}{n_2} \sum_{x \in s_2} (\hat{Y}(x) - \hat{Y}_{s_2})(Y(x) - \hat{Y}_{s_2})^2 \\
\end{align*}
\]

With this notation one has \(\mathbb{E}_x(\sum_{x \in s_2} \frac{Y(x)}{\pi(x)}) = \int_F Y(x) dx\), which is the standard Horwitz-Thompson form.

The optimal weights \(w_{opt}(x)\) minimize \(\sum_{x \in s_2} (w(x) - d(x))^2\) subject to the constraints \(\sum_{x \in s_2} w(x) = \lambda(F)\) and \(\sum_{x \in s_2} w(x) \hat{Y}(x) = \int_F \hat{Y}(x) dx\). The resulting \textbf{calibration estimator} \(\hat{Y}_{MC} = \sum_{x \in s_2} w_{opt}(x) Y(x)\) can be rewritten, according to equation [10] in Wu and Sitter (2001), as

\[
\hat{Y}_{MC} = \frac{1}{n_2} \sum_{x \in s_2} Y(x) + \left( \frac{1}{\lambda(F)} \int_F \hat{Y}(x) - \frac{1}{n_2} \sum_{x \in s_2} \hat{Y}(x) \right) \hat{B}
\]
If the first phase is not exhaustive the obvious choice is to set

\[ \hat{Y}_{MC} = \frac{1}{n_2} \sum_{x \in s_2} Y(x) + \left( \frac{1}{n_1} \sum_{x \in s_1} \hat{Y}(x) - \frac{1}{n_2} \sum_{x \in s_2} \hat{Y}(x) \right) \hat{B} \]

For linear prediction with intercept term the regression line of \( Y(x) \) on \( \hat{Y}(x) \) has intercept 0 and slope 1 so that \( \hat{B} = 1 \) so that the calibration estimator is the same as the regression estimator. Note that with linear models the regression estimator, because of zero mean residuals, can also be written as a weighted sum of the observations, namely \( \sum_{x \in s_2} \omega(x)Y(x) \) with \( \omega(x) = \frac{1}{n_2} g(x) \) and \( g(x) = Z' A g(x) \). \( g(x) \) are the so-called g-weights. One has \( \sum_{x \in s_2} \omega(x)Z(x) = \bar{Z} \), which in this case is equivalent to the calibration constraint \( \sum_{x \in s_2} \omega(x)\hat{Y}(x) = \int_F \hat{Y}(x) \).

In the general case one has

\[ \hat{B} = \frac{\text{Cov}(Y(x), \hat{Y}(x))}{\text{Var}(\hat{Y}(x))} = \hat{\rho} \sqrt{\frac{\text{Var}(Y(x))}{\text{Var}(\hat{Y}(x))}} = \hat{\rho} \hat{R} \]

where \( \hat{\rho} \) is the empirical correlation between \( Y(x) \) and \( \hat{Y}(x) \) and \( \hat{R}^2 = \frac{\text{Var}(\hat{Y}(x))}{\text{Var}(Y(x))} \) is the empirical coefficient of determination. If the model is adequate one will also have \( \sqrt{\hat{R}^2} = \hat{R} \approx \hat{\rho} \) so that in general \( \hat{B} \) will be very close to one.

Summarizing the results, we can say that the regression estimator can also be based on non-linear predictions models and that the external variance formula remains asymptotically valid. Furthermore, the regression estimator is asymptotically equivalent to a special case of the calibration estimator, but it does not require the first-phase to be exhaustive.
### D Tables

#### D.1 Case study

Results when the optimal bandwidth is recalculated for each bootstrap sample.

**Table 7: Global estimation for the case study in Canton Grisons**

<table>
<thead>
<tr>
<th>Estimator</th>
<th>point estimate</th>
<th>standard error</th>
<th>s.e. bootstrap</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{Y}_{\text{lokern}}^{(1)}$</td>
<td>396.30</td>
<td>15.98</td>
<td>35.82</td>
</tr>
<tr>
<td>$\hat{Y}_{\text{glkern}}^{(1)}$</td>
<td>389.88</td>
<td>16.08</td>
<td>24.76</td>
</tr>
<tr>
<td>$\hat{Y}_{\text{lokern}}^{(2)}$</td>
<td>395.71</td>
<td>15.82</td>
<td>35.26</td>
</tr>
<tr>
<td>$\hat{Y}_{\text{glkern}}^{(2)}$</td>
<td>389.77</td>
<td>16.08</td>
<td>24.78</td>
</tr>
</tbody>
</table>

**Table 8: Small area estimation for the case study in Canton Grisons**

<table>
<thead>
<tr>
<th>small area</th>
<th>$G_1$</th>
<th>$G_2$</th>
<th>$G_3$</th>
<th>$G_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_1 : n_2$</td>
<td>94:19</td>
<td>81:17</td>
<td>66:15</td>
<td>65:16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Estimator</th>
<th>$G_1$</th>
<th>$G_2$</th>
<th>$G_3$</th>
<th>$G_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{Y}_{\text{lokern}}^{(1)}$</td>
<td>(27.89)</td>
<td>(33.93)</td>
<td>(29.94)</td>
<td>(33.66)</td>
</tr>
<tr>
<td>[51.75]</td>
<td>[41.04]</td>
<td>[42.67]</td>
<td>[44.22]</td>
<td></td>
</tr>
<tr>
<td>$\hat{Y}_{\text{glkern}}^{(1)}$</td>
<td>(28.82)</td>
<td>(34.63)</td>
<td>(29.65)</td>
<td>(34.65)</td>
</tr>
<tr>
<td>[48.97]</td>
<td>[38.34]</td>
<td>[35.79]</td>
<td>[37.48]</td>
<td></td>
</tr>
<tr>
<td>$\hat{Y}_{\text{lokern}}^{(2)}$</td>
<td>(27.89)</td>
<td>(33.93)</td>
<td>(29.95)</td>
<td>(33.67)</td>
</tr>
<tr>
<td>[51.56]</td>
<td>[37.72]</td>
<td>[42.81]</td>
<td>[44.40]</td>
<td></td>
</tr>
<tr>
<td>$\hat{Y}_{\text{glkern}}^{(2)}$</td>
<td>(28.81)</td>
<td>(34.62)</td>
<td>(29.67)</td>
<td>(34.64)</td>
</tr>
</tbody>
</table>

The standard error based on the external model assumption are given in () and the standard errors based on the bootstrap variance estimates are given in [].

33
D.2 Simulations

Results when the optimal bandwidth is recalculated for each bootstrap sample. The legends are the same as in Tables 3, 4, 5, 6. The ? mark in the column \( \mathbb{E}^*(\hat{\tilde{V}}_{\text{boot}}()) \) indicates that the value is corrupted by some extreme outliers as illustrated in Figures 4 and 5.

Table 9: \( n_1 = 400, n_2 = 100 \): fitted model is the true model

<table>
<thead>
<tr>
<th>Estimator</th>
<th>( \mathbb{E}^<em>(\mathbb{V}^</em>) )</th>
<th>( \mathbb{V}^*(\mathbb{V}()) )</th>
<th>( \mathbb{E}^*(\tilde{V}_{\text{boot}}()) )</th>
<th>( \mathbb{E}^*(P) )</th>
<th>( \mathbb{E}^*(P_{\text{boot}}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{Y}_1 ) lokern</td>
<td>39.16</td>
<td>0.20</td>
<td>0.18</td>
<td>0.24</td>
<td>0.94</td>
</tr>
<tr>
<td>( \hat{Y}_1 ) glkern</td>
<td>39.16</td>
<td>0.19</td>
<td>0.19</td>
<td>0.22</td>
<td>0.94</td>
</tr>
<tr>
<td>( \hat{Y}_2 ) lokern</td>
<td>39.16</td>
<td>0.20</td>
<td>0.18</td>
<td>0.24</td>
<td>0.94</td>
</tr>
<tr>
<td>( \hat{Y}_2 ) glkern</td>
<td>39.16</td>
<td>0.19</td>
<td>0.19</td>
<td>0.22</td>
<td>0.94</td>
</tr>
</tbody>
</table>

Table 10: \( n_1 = 200, n_2 = 50 \): fitted model is the true model

<table>
<thead>
<tr>
<th>Estimator</th>
<th>( \mathbb{E}^<em>(\mathbb{V}^</em>) )</th>
<th>( \mathbb{V}^*(\mathbb{V}()) )</th>
<th>( \mathbb{E}^*(\tilde{V}_{\text{boot}}()) )</th>
<th>( \mathbb{E}^*(P) )</th>
<th>( \mathbb{E}^*(P_{\text{boot}}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{Y}_1 ) lokern</td>
<td>39.17</td>
<td>0.43</td>
<td>0.36</td>
<td>3.24?</td>
<td>0.93</td>
</tr>
<tr>
<td>( \hat{Y}_1 ) glkern</td>
<td>39.18</td>
<td>0.42</td>
<td>0.36</td>
<td>0.52</td>
<td>0.94</td>
</tr>
<tr>
<td>( \hat{Y}_2 ) lokern</td>
<td>39.17</td>
<td>0.43</td>
<td>0.36</td>
<td>0.60</td>
<td>0.93</td>
</tr>
<tr>
<td>( \hat{Y}_2 ) glkern</td>
<td>39.18</td>
<td>0.43</td>
<td>0.36</td>
<td>0.51</td>
<td>0.94</td>
</tr>
</tbody>
</table>
Table 11: \( n_1 = 400, n_2 = 100 \): fitted model is not the true model

<table>
<thead>
<tr>
<th>Estimator</th>
<th>( \mathbb{E}(\cdot) )</th>
<th>( \mathbb{V}(\cdot) )</th>
<th>( \mathbb{E}(\mathbb{V}(\cdot)) )</th>
<th>( \mathbb{E}(\mathbb{V}_{\text{boot}}(\cdot)) )</th>
<th>( \mathbb{E}(P) )</th>
<th>( \mathbb{E}(P_{\text{boot}}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{Y}_{\text{lokern}}^{(1)} )</td>
<td>39.17</td>
<td>0.28</td>
<td>0.27</td>
<td>0.39</td>
<td>0.94</td>
<td>0.96</td>
</tr>
<tr>
<td>( \hat{Y}_{\text{glkern}}^{(1)} )</td>
<td>39.18</td>
<td>0.28</td>
<td>0.26</td>
<td>0.33</td>
<td>0.94</td>
<td>0.96</td>
</tr>
<tr>
<td>( \hat{Y}_{\text{lokern}}^{(2)} )</td>
<td>39.18</td>
<td>0.28</td>
<td>0.26</td>
<td>0.38</td>
<td>0.94</td>
<td>0.96</td>
</tr>
<tr>
<td>( \hat{Y}_{\text{glkern}}^{(2)} )</td>
<td>39.18</td>
<td>0.28</td>
<td>0.26</td>
<td>0.33</td>
<td>0.94</td>
<td>0.96</td>
</tr>
</tbody>
</table>

Table 12: \( n_1 = 200, n_2 = 50 \): fitted model is not the true model

<table>
<thead>
<tr>
<th>Estimator</th>
<th>( \mathbb{E}(\cdot) )</th>
<th>( \mathbb{V}(\cdot) )</th>
<th>( \mathbb{E}(\mathbb{V}(\cdot)) )</th>
<th>( \mathbb{E}(\mathbb{V}_{\text{boot}}(\cdot)) )</th>
<th>( \mathbb{E}(P) )</th>
<th>( \mathbb{E}(P_{\text{boot}}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{Y}_{\text{lokern}}^{(1)} )</td>
<td>39.20</td>
<td>0.61</td>
<td>0.47</td>
<td>1.78?</td>
<td>0.92</td>
<td>0.96</td>
</tr>
<tr>
<td>( \hat{Y}_{\text{glkern}}^{(1)} )</td>
<td>39.20</td>
<td>0.58</td>
<td>0.48</td>
<td>0.76</td>
<td>0.92</td>
<td>0.95</td>
</tr>
<tr>
<td>( \hat{Y}_{\text{lokern}}^{(2)} )</td>
<td>39.20</td>
<td>0.61</td>
<td>0.47</td>
<td>1.02?</td>
<td>0.91</td>
<td>0.96</td>
</tr>
<tr>
<td>( \hat{Y}_{\text{glkern}}^{(2)} )</td>
<td>39.20</td>
<td>0.58</td>
<td>0.48</td>
<td>0.77</td>
<td>0.92</td>
<td>0.95</td>
</tr>
</tbody>
</table>
E Figures

E.1 Case study

Figure 1: Predictions $\tilde{Y}_{gI-lo Kern}(x_0)$ versus $\hat{Y}(x_0)$
Figure 2: Variance as a function of $k$

**Influence of choice of k-neighbors**

Case study in canton of Grisons with $n_1=306$ and $n_2=67$

Estimated external vs. bootstrap standard errors (9999 replicates)

- **External standard error (gaussian kernel)**
- **Bootstrapped standard error (gaussian kernel)**
E.2 Simulation

Figure 3: Local density $Y(x)$
Figure 4: Bootstrap variances for $n_1 = 400, n_2 = 100$
Figure 5: Bootstrap variances for $n_1 = 200$, $n_2 = 50$
Figure 6: Variance as a function of $k$

Influence of choice of k-neighbors

Empirical vs. mean of estimated external variances

Simulation under true model with $n_1=400$ and $n_2=100$ (3000 iterations)
E.3 R-Code

Below we give the R-code to run a single iteration of the simulation to produce an artificial data set and to calculate with it all the estimators discussed in this report, in particular the bootstrap variance estimates. Potential users can easily change the variables names to run the program on their own data sets.

```
######################
### SAMPLE R CODE ###
######################

#LOAD LIBRARIES
library(kknn)
library(lokern)
library(np)
library(boot)

### CREATE SINGLE EXAMPLE DATA (CALLED "s1") SET FROM SIMULATION SURFACE
set.seed(7)
n1 <- 200
n2 <- 50
s1 <- data.frame(s2.indicator = c(rep(1, times=n2), rep(0, times=n1-n2)), x=2*runif(n1), y=3*runif(n1))
s1$xx <- s1$x^2
s1$yy <- s1$y^2
s1$response <- ifelse(s1$s2.indicator==1, 30 + 13*s1$x - 6*s1$y - 4*s1$x^2 + 3*s1$x*s1$y + 2*s1$y^2 + 6*cos(pi*s1$x)*sin(2*pi*s1$y), NA)
head(s1, 60)

### CREATE TRAINING SET
# - if true external model is used we need to select independent training set from s1
training.set <- data.frame(x=2*runif(n2), y=3*runif(n2))
training.set$xx <- training.set$x^2
training.set$yy <- training.set$y^2
training.set$response <- 30 + 13*training.set$x - 6*training.set$y - 4*training.set$x^2 + 3*training.set$x*training.set$y + 2*training.set$y^2 + 6*cos(pi*training.set$x)*sin(2*pi*training.set$y)
training.set <- training.set[training.set$s2.indicator==1,]

### DEFINE FUNCTIONS ###

twophase_est <- function(ground.truth.s2, predictions.s1, residuals.s2){
  n1<- length(predictions.s1)
  n2<- length(residuals.s2)
  Est<- mean(predictions.s1) + mean(residuals.s2)
  Var<- (1/n1)*var(ground.truth.s2) + (1-(n2/n1))*(1/n2)*var(residuals.s2)
  Err<- sqrt(Var)
  out<- list(Est=Est, Err=Err)
  return(out)
}

#special leave-one-out-cross validation function for npreg(2) bandwidth selection

cv_bws_npreg <- function(x, y, bandwidths=c(1,2,4,6,10,15,20,30,40,50)) {
  n <- length(x)
  fold_MSEs <- matrix(0, nrow=n, ncol=length(bandwidths))
  colnames(fold_MSEs) = bandwidths
  case.folds <- 1:n
  for (loo in 1:n) {
    x.train = x[-loo]
    y.train = y[-loo]
    x.test = x[loo]
    y.test = y[loo]
    x.df <- as.data.frame(x.test)
    reg <- lm(y ~ x)
    pred <- reg$fitted.values
    res <- reg$residuals
    hat <- npreg(res ~ pred, regtype = "lc", bws = 1)$mean
    reg <- lm(y~train * x.train)
    pred.train <- reg$fitted.values
    pred.test <- predict(object = reg, newdata = x.df)
    pred.test.df <- as.data.frame(pred.test)
    names(pred.test.df) <- "pred.train"
    fold_MSEs[loo,] <- (1/(n-1)) * (sum((y.train - reg$fitted.values)^2) + sum((hat - res)^2))
  }
  out <- list(Est=Est, Err=Err)
  return(out)
}
```

42
```r
resid <- y.train - pred.train
for (bw in bandwidths) {
  fit <- pred.test + npreg(resid ~ pred.train, regtype = "lc", newdata = pred.test.df, bw = bw)$mean
  fold_MSEs[[loca,paste(bw)]] <- (y.test - fit)^2
}
CV_MSEs = colMeans(fold_MSEs)
best_bw = bandwidths[which.min(CV_MSEs)]
return(best_bw)
}

# CHOOSE DESIRED MODEL

# FIT LINEAR MODEL AND ADD REGRESSION PREDICTIONS TO s1
model <- lm(model.formula, data=training.set, y=TRUE) #note: the training set is used here
summary(model)
s1$lm.predictions <- predict(object=model, newdata=s1)
s1 <- s1[sort.list(s1$lm.predictions),] # SOME PROCEDURES REQUIRE SORTING
s1$lm.residuals <- s1$response - s1$lm.predictions

# MAKE PREDICTIONS FOR GLKERN(1), NPREG(1), GLKERN(2), NPREG(2), KNN(multidim), and KNN(pred)

# npreg1 <- npreg(response ~ lm.predictions, data=s1[s1$s2.indicator==1,], newdata=s1, bwmethod="cv.ls")
npreg1_band <- npreg1$bw #note: npreg's internally produced bandwidths have different scaling than cv_bws_npreg()
s1$npreg1 <- npreg1$mean

# npreg2_band <- cv_bws_npreg(x=s1$lm.predictions[s1$s2.indicator==1], y=s1$response[s1$s2.indicator==1], bandwidths=seq(0.5, 10, 0.5)) # 2
s1$npreg2 <- npreg(lm.residuals ~ lm.predictions, data=s1[s1$s2.indicator==1,], newdata=s1, bws=npreg2_band)$mean + s1$lm.predictions

# leave one out cross validation to choose optimal k (k=3 in this run of the code)
knn_multi_band <- train.kknn(formula=model.formula, train=s1[s1$s2.indicator==1,], test=s1, kernel="gaussian", kmax=15)$best.parameters$k
knn_multi <- kknn(formula=model.formula, train=s1[s1$s2.indicator==1,], test=s1, kernel="gaussian", k=knn_multi_band)
s1$knn_multi <- knn_multi$fitted.values

# leave one out cross validation to choose optimal k (k=12 in this run of the code)
knn_pred_band <- train.kknn(response ~ lm.predictions, data=s1[s1$s2.indicator==1,], kernel="gaussian", kmax=15)$best.parameters$k
knn_pred <- kknn(formula=response ~ lm.predictions, train=s1[s1$s2.indicator==1,], test=s1, kernel="gaussian", k=knn_pred_band)
s1$knn_pred <- knn_pred$fitted.values

# Use difference estimator on above predictions

# CALCULATE SAMPLE MEAN AND ITS STANDARD ERROR
mean(s1$response, na.rm=TRUE) # 40.52217
sqrt(var(s1$response, na.rm=TRUE)/n2) # 1.130878

# CALCULATE REG ESTIMATOR
REG <- twophase_est(ground.truth.s2 = s1[s1$s2.indicator==1, "response"],
predictions.s1 = s1[, "lm.predictions"],
residuals.s2 = s1[s1$s2.indicator==1, "lm.residuals"])
REG$Est # 39.75979
REG$Err # 0.7247854

# CALCULATE GLKERN1 ESTIMATOR
```
GLKERN1 <- twophase_est(ground.truth.s2 = s1[s1$s2.indicator==1, "response"], predictions.s1 = s1[, "glkerns1"], residuals.s2 = s1[s1$s2.indicator==1, "response"] - s1[s1$s2.indicator==1, "glkerns1"])
GLKERN1$Est # 39.43728
GLKERN1$Err # 0.694061

# CALCULATE NPREG1 ESTIMATOR
NPREG1 <- twophase_est(ground.truth.s2 = s1[s1$s2.indicator==1, "response"], predictions.s1 = s1[, "npreg1"], residuals.s2 = s1[s1$s2.indicator==1, "response"] - s1[s1$s2.indicator==1, "npreg1"])
NPREG1$Est # 39.55139
NPREG1$Err # 0.7038791

# CALCULATE GLKERN2 ESTIMATOR
GLKERN2 <- twophase_est(ground.truth.s2 = s1[s1$s2.indicator==1, "response"], predictions.s1 = s1[, "glkerns2"], residuals.s2 = s1[s1$s2.indicator==1, "response"] - s1[s1$s2.indicator==1, "glkerns2"])
GLKERN2$Est # 39.43422
GLKERN2$Err # 0.6943259

# CALCULATE NPREG2 ESTIMATOR
NPREG2 <- twophase_est(ground.truth.s2 = s1[s1$s2.indicator==1, "response"], predictions.s1 = s1[, "npreg2"], residuals.s2 = s1[s1$s2.indicator==1, "response"] - s1[s1$s2.indicator==1, "npreg2"])
NPREG2$Est # 39.56989
NPREG2$Err # 0.7031609

# CALCULATE KNN(multidim) ESTIMATOR
KNN_multidim <- twophase_est(ground.truth.s2 = s1[s1$s2.indicator==1, "response"], predictions.s1 = s1[, "knn_multi"], residuals.s2 = s1[s1$s2.indicator==1, "response"] - s1[s1$s2.indicator==1, "knn_multi"])
KNN_multidim$Est # 39.60094
KNN_multidim$Err # 0.6325992
#Note: when expect this to be underestimation because we treated an internal model as external

# CALCULATE NPREG2 ESTIMATOR
KNN_pred <- twophase_est(ground.truth.s2 = s1[s1$s2.indicator==1, "response"], predictions.s1 = s1[, "knn_pred"], residuals.s2 = s1[s1$s2.indicator==1, "response"] - s1[s1$s2.indicator==1, "knn_pred"])
KNN_pred$Est # 39.63425
KNN_pred$Err # 0.698619

# EXAMPLE OF BOOTSTRAP ROUTINE
#

# Define function that inputs a data.frame with index vector and outputs estimates for above estimators.
# When bandwidths are allowed to be reoptimized within the following routine instability can result.
# One must also pass the desired model formula to this function
boot.vector <- function(data, i, glkerns1_band=glkerns1_band, glkerns2_band=glkerns2_band, npreg1_band=npreg1_band, npreg2_band=npreg2_band, knn_multi_band=knn_multi_band, knn_pred_band=knn_pred_band, model.formula=model.formula){
  df=data[i,]
  training.set <- df[df$s2.indicator==1,] #you change this for true external model approach (not done here)
  model <- lm(model.formula, data=training.set, y=TRUE) #note: the internal training set is used here
  df$lm.predictions <- predict(object=model, newdata=df)
  df <- df[sort.list(df$lm.predictions),] # SOME PROCEDURES REQUIRE SORTING
  df$lm.residuals <- df$response - df$lm.predictions
  MEAN <- mean(df$response, na.rm=TRUE)
  #---
  df$lm.predictions <- predict(object=model, newdata=df)
  df$lm.predictions <- df$lm.predictions
  df$lm.residuals <- df$response - df$lm.predictions
  MEAN <- mean(df$response, na.rm=TRUE)
  #---
  for (i in 1:1000){
    df$glkerns1 <- glkerns(x=df[df$s2.indicator==1, "lm.predictions"], y=df[df$s2.indicator==1, "response"], x.out=df[, "lm.predictions"], bandwidth=glkerns1_band)
    df$glkerns2 <- glkerns(x=df[df$s2.indicator==1, "lm.predictions"], y=df[df$s2.indicator==1, "response"], x.out=df[, "lm.predictions"], bandwidth=glkerns2_band, best=best
    GLKERN1 <- twophase_est(ground.truth.s2 = df[df$s2.indicator==1, "response"], predictions.s1 = df[, "glkerns1"], residuals.s2 = df[df$s2.indicator==1, "response"] - df[df$s2.indicator==1, "glkerns1"], method=twophase_est)
    GLKERN1$Est # 39.43728
    GLKERN1$Err # 0.694061
  }
}
predictions.s1 = df[, "glkerns1"],
residuals.s2 = df[df$s2.indicator==1, "response"] - df[df$s2.indicator==1, "glkerns1"]
#----#
npreg1 <- npreg(response ~ lm.predictions, data=df[df$s2.indicator==1,],
newdata=df, bws=npreg1_band)$mean
NPREG1 <- twophase_est(ground.truth.s2 = df[df$s2.indicator==1, "response"],
predictions.s1 = df[, "npreg1"],
residuals.s2 = df[df$s2.indicator==1, "response"] - df[df$s2.indicator==1, "npreg1"],
predictions.s1 = df[, "npreg1"],
residuals.s2 = df[df$s2.indicator==1, "response"] - df[df$s2.indicator==1, "npreg1"])
#----#
glkerns2 <- glkerns(x=df[df$s2.indicator==1, "lm.predictions"], y=df[df$s2.indicator==1, "lm.residuals"],
x.out=df[, "lm.predictions"], bandwidth=glkerns2_band)
glkerns2est <- glkerns$est + df$lm.predictions
GLKERN2 <- twophase_est(ground.truth.s2 = df[df$s2.indicator==1, "response"],
predictions.s1 = df[, "glkerns2"],
residuals.s2 = df[df$s2.indicator==1, "response"] - df[df$s2.indicator==1, "glkerns2"])
#----#
npreg2 <- npreg(lm.residuals ~ lm.predictions, data=df[df$s2.indicator==1,],
newdata=df, bws=npreg2_band)$mean + df$lm.predictions
NPREG2 <- twophase_est(ground.truth.s2 = df[df$s2.indicator==1, "response"],
predictions.s1 = df[, "npreg2"],
residuals.s2 = df[df$s2.indicator==1, "response"] - df[df$s2.indicator==1, "npreg2"])$Est
#----#
knn_multi <- kknn(formula=model.formula, train=df[df$s2.indicator==1,], test=df, kernel="gaussian", k=knn_multi_band)$fitted.values
KNN_multidim <- twophase_est(ground.truth.s2 = df[df$s2.indicator==1, "response"],
predictions.s1 = df[, "knn_multi"],
residuals.s2 = df[df$s2.indicator==1, "response"] - df[df$s2.indicator==1, "knn_multi"])
#----#
knn_pred <- kknn(formula=response ~ lm.predictions, train=df[df$s2.indicator==1,], test=df, kernel="gaussian", k=knn_pred_band)$fitted.values
KNN_pred <- twophase_est(ground.truth.s2 = df[df$s2.indicator==1, "response"],
predictions.s1 = df[, "knn_pred"],
residuals.s2 = df[df$s2.indicator==1, "response"] - df[df$s2.indicator==1, "knn_pred"])$Est
return(c(MEAN, REG, GLKERN1, NPREG1, GLKERN2, NPREG2, KNN_multidim, KNN_pred))

#Call bootstrap (uncomment "parallel="multicore", ncpus=4" if parallel processing on linux or Mac OSx)
bootvec <- boot(s1, boot.vector, R = 10000,
parallel="multicore", ncpus=4, # uncomment for multicore processing on linux or Mac OSx
glkerns1_band=glkerns1_band, glkerns2_band=glkerns2_band,
npreg1_band=npreg1_band, npreg2_band=npreg2_band,
npreg1_band=npreg1_band, npreg2_band=npreg2_band,
nk_multi_band=knn_multi_band, knn_pred_band=knn_pred_band,
model.formula=model.formula))

bootvec
#Note: These bootstrap results (based on 10000 replicates) may differ from simulation results because the
# original sample here is a SINGLE sample, not the mean and variance of 10000 samples. That type of
# simulation is possible because it is an artificial example where we know the truth.