Policy Research Working Paper 10429

Poverty Mapping in the Age of Machine Learning

Paul Corral
Heath Henderson
Sandra Segovia

WORLD BANK GROUP
Poverty and Equity Global Practice
May 2023

Policy Research Working Paper 10429

Abstract

Recent years have witnessed considerable methodological advances in poverty mapping, much of which has focused on the application of modern machine-learning approaches to remotely sensed data. Poverty maps produced with these methods generally share a common validation procedure, which assesses model performance by comparing subnational machine-learning-based poverty estimates with survey-based, direct estimates. Although unbiased, survey-based estimates at a granular level can be imprecise measures of true poverty rates, meaning that it is unclear whether the validation procedures used in machine-learning approaches are informative of actual model performance. This paper examines the credibility of existing approaches to model validation by constructing a pseudo-census from the Mexican Intercensal Survey of 2015, which is used to conduct several design-based simulation experiments. The findings show that the validation procedure often used for machine-learning approaches can be misleading in terms of model assessment since it yields incorrect information for choosing what may be the best set of estimates across different methods and scenarios. Using alternative validation methods, the paper shows that machine-learning-based estimates can rival traditional, more data intensive poverty mapping approaches. Further, the closest approximation to existing machine-learning approaches, using publicly available geo-referenced data, performs poorly when evaluated against "true" poverty rates and fails to outperform traditional poverty mapping methods in targeting simulations.

This paper is a product of the Poverty and Equity Global Practice. It is part of a larger effort by the World Bank to provide open access to its research and make a contribution to development policy discussions around the world. Policy Research Working Papers are also posted on the Web at http://www.worldbank.org/prwp. The authors may be contacted at pcorralrodas@worldbank.org

Produced by the Research Support Team

Poverty Mapping in the Age of Machine Learning

Paul Corral,[*] Heath Henderson,[†] and Sandra Segovia[‡][§]

Key words: Small area estimation, Poverty mapping, Machine learning, Satellite imagery

JEL classification: C13, C55, C87, C15

[*] The World Bank Group - Poverty and Equity Global Practice (pcorralrodas@worldbank.org)

[†] Department of Economics and Finance, Drake University, United States of America

[‡] The World Bank Group

[§] The authors acknowledge financial support from the World Bank. We thank Roy van der Weide and Isabel Molina for comments. Additionally, we thank Benu Bidani, Carlos Rodriguez-Castelan, and Johan Mistiaen for providing support and space to pursue this work. Finally, we thank the Global Solutions Group on Data for Policy. Any error or omission is the authors' responsibility alone.

1 Introduction

Poverty maps provide granular estimates of poverty at the sub-national level in order to deepen the understanding of poverty in a given country, better inform the targeting of resources, and support the design of interventions tailored to local needs (Bedi et al. 2007; Elbers et al. 2007). Household surveys provide estimates that are sufficiently reliable for large areas in a given country but often lack the desired precision and coverage to be able to properly inform targeting interventions at a granular geographic level. Thus, it is necessary to rely on small area estimation.

Small area estimation is a branch of statistics focused on obtaining estimates of higher quality than those obtained directly from the household survey. These small area estimation techniques often combine data from household surveys and auxiliary information from censuses, registers, or others to produce estimates of higher quality than what is possible from survey data alone for areas or groups with small sample sizes. Since there is no such thing as a free lunch, to achieve these gains in quality it is necessary to rely on model assumptions which must be thoroughly checked.

The literature on small area estimation is rich and several variations on this basic procedure have been proposed. Unit-level models conduct estimation and prediction at the household level, assuming a linear relationship between the welfare measure and covariates (Hentschel et al., 1998; Elbers et al., 2003; Molina and Rao, 2010). Unit-level models are not well-suited to situations where the survey and census data correspond to different years, which is often the case in developing countries where censuses are conducted infrequently. Area-level models represent a feasible alternative that similarly relies on linear functional forms, but conduct estimation and prediction using only aggregate data for the geographical entities of interest (Fay and Herriot, 1979; Torabi and Rao, 2014). Unit-context models represent another alternative and are characterized by an estimation stage wherein household-level measures are modeled exclusively as a linear function of area-level characteristics (Nguyen, 2012; Lange et al., 2018; Masaki et al., 2020). [1]

Recent developments in poverty mapping have focused on the application of machine learning to remotely-sensed data, largely in response to the issue of out-of-date census information. These approaches also use survey-derived welfare measures but are distinguished by a first stage where a machine-learning model is fit to remotely-sensed covariates (e.g., data derived from satellite imagery or call detail records) rather than census-based covariates. For example, Chi et al. (2022) matched data from several remotely-sensed data sources to survey-based measures of "village" wealth for 56 countries, and then fit a prediction model using gradient boosting machines.[2] They then applied the model to the populated surface of all 135 low- and middle-income countries to develop granular poverty maps for each country. Similar approaches have been used for individual countries, including Rwanda (Blumenstock et al., 2015), Senegal (Pokhriyal and Jacques, 2017),

[1] Despite their desirable features, the gains in precision offered by area-level models is often quite limited (see Molina and Morales (2009)).

[2] Their remotely-sensed data includes high-resolution satellite imagery, mobile phone data, topographic maps, and connectivity data from Facebook.

Bangladesh (Steele et al., 2017), and Belize (Hersh et al., 2021), among others.[3]

While often subsumed under the term "machine learning," this modern approach to poverty mapping actually consists of several practices that differ from more traditional approaches. Most obviously, the modern approach substitutes non-parametric statistical methods for the parametric approaches traditionally used for poverty mapping. In addition, the modern approach relies heavily on remotely-sensed covariates rather than covariates derived from census data. Finally, poverty maps produced with machine-learning methods generally share a common validation procedure. The key feature of this procedure is that model performance is assessed by calculating the R^2 from a regression of the observed poverty measures on the estimated poverty measures for the same geographical units.[4] The poverty measures used in this procedure are most often direct, sample-based estimates rather than the true poverty measures for the regions of interest. While direct estimates are unbiased, they can be imprecise estimates of true poverty measures, meaning that it is unclear whether the validation procedure is informative of actual model performance (Corral et al., 2021b).

In this paper, we provide a more rigorous assessment of the performance of the modern approach to poverty mapping. Our approach consists of constructing a pseudo-census (hereafter "census") that we use to conduct design-based simulation experiments. The census provides the true values we wish to estimate. Our experiments entail repeatedly drawing survey samples from the census where each sample produces direct poverty estimates for the sampled geographic areas. For each direct estimate, the presence of the census means that we observe the true poverty measure corresponding to the direct estimate and we use these true poverty measures to gain insight not only into the credibility of model validation based on direct estimates, but also the performance of machine-learning methods when validated against the true poverty measures. Given that population censuses rarely collect detailed data on income or expenditures, we construct our census from a large-scale household survey conducted in Mexico: the Mexican Intercensal Survey of 2015.

Prior to presenting our simulation results, we discuss three ways in which the R^2 can be misleading in terms of model assessment. Specifically, we show analytically that the R^2 based on direct estimates is biased downward, that it is insensitive to differences in location and scale between the poverty estimates and the reference measures, and that it is context-dependent in the sense that it is influenced by the variance of the true poverty rates. We argue that the mean squared error is a better measure for understanding the strength of the association between the poverty estimates and reference measures. In addition, we argue that the ultimate concern is to understand how different methodological choices affect poverty by influencing the efficiency of targeting. We therefore propose a targeting experiment through which we examine the relative ability of alternative

[3] See Jean et al. (2016), Yeh et al. (2020), Lee and Braithwaite (2020), and Aiken et al. (2022) for examples of other recent applications.

[4] For examples of the above procedure, see Jean et al. (2016), Pokhriyal and Jacques (2017), Steele et al. (2017), Lee and Braithwaite (2020), Yeh et al. (2020), or Chi et al. (2022). Note that a few studies use a similar procedure, but fit the machine-learning model to household-level data and then aggregate the model's predictions to the geographic unit of interest (Blumenstock et al., 2015; Hersh et al., 2021; Aiken et al., 2022). Others rely on reporting correlation coefficients between survey based estimates and model predictions, which is similar (see Smythe and Blumenstock 2022).

mapping methods to alleviate poverty in the context of the Mexican data.

The results of our simulations can be summarized as follows. First, we find that the magnitude of the downward bias of the R^2 based on direct estimates is large, with the true R^2 being around 35 to 50 percent higher depending on the model considered. We further find that this bias has important implications for model selection in that the direct R^2 incorrectly identifies the appropriate level of estimation in the vast majority of our simulations. Second, when assessing model performance based on the true poverty rates using the mean squared error, we find that our closest approximation to the standard machine-learning implementation performs poorly relative to several benchmark models. We find that these performance issues are largely due to the limited predictive power of remotely-sensed covariates relative to census-based covariates. Finally, our targeting simulations show that none of our machine-learning implementations outperform more traditional poverty mapping methods that are feasible with the same data.

Our paper builds on the work of Corral et al. (2021a) and Corral et al. (2022), who similarly used the Mexican Intercensal Survey to examine the performance of several poverty mapping methods. Corral et al. (2021a) focused exclusively on traditional poverty mapping approaches and did not consider the performance of machine-learning methods. While Corral et al. (2022) consider the performance of machine-learning approaches, their implementation only uses census-based covariates and thus does not examine the performance of the standard implementation that relies on remotely-sensed covariates. Importantly, we are not the first paper to evaluate the performance of machine-learning methods relative to census data. While Yeh et al. (2020) and Chi et al. (2022) validate their models relative to censuses, both papers use the R^2 as a performance metric and focus on wealth estimates rather than poverty. Finally, van der Weide et al. (2022) examine how well machine-learning methods using remotely-sensed data can reproduce poverty maps estimated using census data. They do not, however, compare their estimates to a ground truth.

This paper then represents the first attempt to rigorously assess the performance of modern poverty mapping methods relative to true poverty rates. In what follows, Section 2 discusses the data we use for our experiments and Section 3 discusses the various methods we examine, including both machine-learning methods and some traditional poverty mapping methods that we use as performance benchmarks. Section 4 then considers the issue of model validation where we critically assess the R^2 as a validation metric and then propose some alternative metrics. Finally, Section 5 presents the results of our experiments and Section 6 provides some concluding remarks.

2 Data

The Mexican Intercensal Survey was carried out by the Mexican National Institute of Statistics and Geography (INEGI). The sample consists of 5.9 million households and is representative at the national, state (32 states), and municipality level (2,457 municipalities). It is also representative for localities with populations of 50,000 or more inhabitants. The administered questionnaire gathered information on household income, geographic location, household demographics, dwelling characteristics, and economic activities, among others. The size of the dataset is especially important, as it allows us to draw repeated samples that are sufficiently large and diverse. In addition, the fact that the survey gathered detailed income information on all households allows us to reliably calculate the required poverty measures that serve as the basis for our simulation experiments.[5]

Prior to sampling from the Intercensal Survey, we make three modifications to the data. First, as a large number of households reported earning no income, we randomly remove 90 percent of these households.[6] Second, to ensure that all municipalities are sufficiently large, all municipalities with less than 500 households are removed. Finally, to ensure that all primary sampling units (PSUs) are also sufficiently large for sampling purposes, we combine several neighboring PSUs such that all include at least 300 households. Our final census dataset consists of 3.9 million households in 1,865 municipalities and 16,297 PSUs. The resulting census is used to draw 500 survey samples that serve as the basis for our simulation experiments (Tzavidis et al., 2018). In what follows, we describe our approach to constructing these samples.

Our sampling procedure is intended to reflect standard design elements used in many face-to-face household surveys conducted in the developing world (Grosh and Muñoz, 1996). Mexico's 32 states comprise the intended domains of the sample and the indicator of interest is household income per capita. For each sample, we seek to achieve a relative standard error (RSE) of 7.5 percent for average household income in each state.[7] We use a two-stage clustered design wherein PSUs or clusters are selected within each domain and then a sample of 10 households is selected within each cluster. Our decision to select 10 households within each cluster is consistent with the design of other large-scale household surveys conducted in developing countries. Under simple random sampling, the minimum sample size required for each state is as follows:

$$n_s = \left(\frac{\sigma_s}{\bar{w_s} \times \text{RSE}}\right)^2 \tag{1}$$

where $\bar{w_s}$ and σ_s represent average household income per capita and the standard deviation of per capita income for state s, respectively.

The minimum sample size under simple random sampling must be adjusted to account for the clustered design. The design effect due to clustering is accounted for by estimating the intra-cluster correlation of per capita income within each state. The correlation estimates can then be used to adjust the above sample sizes for the clustered design as follows:

$$n_s \times \Big[1 + \rho_s \left(n_{sc} - 1\right)\Big] \tag{2}$$

[5] Income is defined as money received from work performed during the course of the reference period by individuals aged 12 or older.

[6] This is done to ensure some missing values are present in the data used but not as many as in the original data.

[7] The desired precision of 7.5 percent is somewhat arbitrary, but corresponds to precision targets in similar surveys and yields samples of reasonable sizes.

where n_{sc} denotes the number of households selected in cluster c within state s (10 in this case) and ρ_s is the intra-cluster correlation of per capita income. Given the above, we can calculate the number of clusters needed to achieve the minimum sample size for each state and then multiply this number by 10 to find the final household sample size. Taking these sample size requirements as given, each of our samples is then drawn in accordance with the two-stage design.

Specifically, the clusters within each state are selected with probability proportional to size (without replacement), where the measure of size is the number of households within each cluster. The households within each cluster are then selected via simple random sampling. According to this design, the inclusion probability for a given household is then approximately:

$$\frac{\tilde{n}_{sc} N_s}{\tilde{n}_s} \times \frac{n_{sc}}{\tilde{n}_{sc}} \tag{3}$$

where $\tilde{n}_{sc}$ is the total number of households in cluster c within state s, N_s is the number of clusters selected in state s, and $\tilde{n}_s$ is the total number of households in state s.[8] The sample size across the 500 samples is roughly 23,500 households. Under this sampling scenario, not all municipalities are included, and the number of municipalities included varies from sample to sample, ranging between 951 to 1,020 municipalities. The median municipality included in a given sample is represented by a single cluster and thus its sample size is 10 households.

In addition to the Intercensal Survey, INEGI has also released geo-referenced data that is similar to the remotely-sensed data often used in machine-learning applications. These geographic variables were calculated by INEGI for the year 2015 (the same year as the Intercensal Survey) and are only available at the municipality level. The geo-referenced data includes 105 different indicators. Specifically, the dataset includes information from 21 different dimensions and each dimension is associated with five different summary measures. For example, we summarize the atmospherically resistant vegetation index for each municipality using the minimum, maximum, mean, sum, and standard deviation of the index. The same procedure is applied to all dimensions.[9] We then have two alternative sets of covariates that we use in our simulations: the geo-referenced data and the standard set of sociodemographic characteristics from the census.

[8] The equation used to calculate inclusion probabilities assumes sampling with replacement, but is used here as an approximation of inclusion probabilities under proportional selection without replacement. This should provide a reasonable approximation in this case since there are a relatively large number of clusters present in the frame. The design weight for each household is simply the inverse of the inclusion probability. In a typical survey, the design weights would be further adjusted for nonresponse and calibrated to known population characteristics. However, since the sampling is only a simulation exercise, there is no nonresponse and thus no nonresponse adjustment is required. Calibration or post-stratification could be performed, but was not implemented to simplify the process.

[9] The 21 dimensions are as follows: enhanced vegetation index, normalized difference vegetation index, normalized difference built-up index, built-up index, simple ratio, atmospherically resistant vegetation index, urban index, index of biological integrity, normalized difference water index, modified normalized difference water index, new built-up index, band ratio for built-up area, normalized built-up area index, built-up area extraction index, normalized difference snow index, visible atmospherically resistant index, soil-adjusted vegetation index, optimized soil-adjusted vegetation index, normalized difference moisture index, digital altitude model, and digital slope model.

3 Methods

This section describes the methods we apply to the Mexican Intercensal Survey. The first subsection discusses machine learning, with a special focus on gradient boosting, as it is among the most popular machine-learning methods for poverty mapping. The second subsection then discusses more traditional approaches to poverty mapping, which we use as the benchmark in our performance assessment of the machine-learning methods.

3.1 Machine learning

The goal of machine learning is to develop high-performance algorithms for prediction, classification, and clustering/grouping tasks (Varian, 2014; Athey, 2018; Athey and Imbens, 2019). Such tasks can be divided into supervised and unsupervised learning problems. While unsupervised learning focuses on identifying clusters of observations that are similar in terms of their features, supervised learning uses a set of features to predict some outcome of interest. Supervised learning can be further divided into regression and classification tasks, where regression is concerned with predicting continuous outcomes and classification focuses on categorical outcomes. While there are many machine-learning methods used for regression and classification tasks (e.g., lasso, random forests, or support vector machines), here we fix ideas by focusing on a method that has been particularly popular for poverty mapping: gradient boosting.

Originally developed by Friedman (2001), gradient boosting machines are a family of machine-learning techniques that combine a large number of weak learners to form a stronger ensemble prediction. Unlike some common ensemble techniques (e.g., random forests) that simply average models in the ensemble, gradient boosting adds models to the ensemble sequentially by fitting new weak learners to the negative gradient of some chosen loss function. With the classic squared-error loss function, this amounts to sequentially fitting new models to the current residuals of the ensemble.[10] Extreme gradient boosting (XGBoost) is a particularly popular implementation of gradient boosting that uses classification and regression trees as the base models or weak learners. The popularity of XGBoost is due to the fact that it is fast, scalable, and has been shown to outperform competing methods across a wide range of problems (Chen and Guestrin, 2016).

Classification and regression trees are based on a sequential binary splitting of the covariate space that serves to partition the data into subsamples that are increasingly homogeneous in terms of the outcome variable. A tree begins with a single root node that is split into two child nodes according to some decision rule. A decision rule pertains to a single explanatory variable and observations are assigned to the child nodes depending on whether they meet some condition associated with the explanatory variable.[11] The child nodes can be further split into new child nodes based on

[10] See Natekin and Knoll (2013) for an accessible introduction to gradient boosting.

[11] For example, for a continuous explanatory variable, one determines whether a given observation falls above or below some numeric cutoff.

different decision rules that involve other explanatory variables and cutoff points. Any child node that is not further split is referred to as a terminal node or leaf, and each leaf is associated with a parameter that provides a prediction for the subsample associated with that leaf. For any given observation, the ensemble prediction used by XGBoost is the sum of that observation's predictions across all trees in the ensemble.

In what follows, we discuss the key features of the XGBoost algorithm, drawing on the presentation provided in Chen and Guestrin (2016). Let y_i^d denote the outcome of interest (e.g., direct estimates of poverty rates) for observation or entity i, and let x_i denote a vector of explanatory variables. In line with the above, the predicted value of the outcome y_i^p is given by the sum of predictions across the individual trees in the ensemble:

$$y_i^p = \sum_t f_t(x_i) \tag{4}$$

where the trees are indexed by t and $f_t(x_i)$ gives the prediction of a given tree as a function of x_i. To build the trees used in the model, XGBoost minimizes the following regularized objective function:

$$\mathcal{L} = \sum_i l(y_i^d, y_i^p) + \sum_t r(f_t) \tag{5}$$

where $l(y_i^d, y_i^p)$ is the loss associated with the i^{th} observation and $r(f_t)$ is a regularization term that serves to mitigate overfitting. In our implementation, we use the classic squared-error loss such that $l(y_i^d, y_i^p) = (y_i^d - y_i^p)^2$.

XGBoost uses the following specification of the regularization term:

$$r(f_t) = \gamma M_t + \frac{1}{2}\lambda \sum_j m_{tj}^2 \tag{6}$$

where M_t is the total number of leaves in tree t, m_{tj} is the prediction associated with the j^{th} leaf in a given tree, and γ and λ are hyperparameters that must be chosen by the user (we discuss our selection of hyperparameters below). The regularization term serves to mitigate overfitting by penalizing complicated trees and smoothing the predictions associated with each tree. That is, the method relies on a large number of weak learners and the regularization term helps control the complexity of the learners. It is important to note that the above specification of the regularization term is only one of many possible specifications, though Chen and Guestrin (2016) claim that it is simpler than the leading alternatives and easier to parallelize. Further note that when the hyperparameters are set to zero, the objective function becomes equivalent to that used in traditional gradient boosting.

To minimize the objective function, the model is trained in a sequential manner, building one tree at a time. The objective function at iteration t can be written as follows:

$$\mathcal{L}_t = \sum_i l(y_i^d, y_{i,t-1}^p + f_t(x_i)) + r(f_t) \tag{7}$$

where $y_{i,t}^p = y_{i,t-1}^p + f_t(x_i)$. Each iteration then adds the tree $f_t(x_i)$ that greedily minimizes the objective function at that iteration. More specifically, XGBoost uses a second-order Taylor series expansion to approximate the loss function at any given iteration:

$$\tilde{\mathcal{L}}_t = \sum_i \left\{ g(y_i^d, y_{i,t-1}^p) f_t(x_i) + \frac{1}{2} h(y_i^d, y_{i,t-1}^p) f_t(x_i)^2 \right\} + r(f_t) \tag{8}$$

where $g(\cdot,\cdot)$ and $h(\cdot,\cdot)$ denote the first- and second-order derivatives of the loss function $l(y_i^d, y_{i,t-1}^p)$ with respect to $y_{i,t-1}^p$. That is, the point $y_{i,t-1}^p$ is taken as the basis for the expansion. The above conveniently omits the first term of the series $l(y_i^d, y_{i,t-1}^p)$ because it is a constant.

The XGBoost algorithm is based on a re-expression of Eq. (8). Let z_j denote the set of observations that belong to the j^{th} leaf of the tree being built at iteration t. Substituting in the explicit form of the regularization term and noting that all observation that belong to the same leaf receive the same prediction m_{tj}, we can write

$$\tilde{\mathcal{L}}_t = \sum_j \sum_{i \in z_j} \left\{ g(y_i^d, y_{i,t-1}^p) m_{tj} + \frac{1}{2} h(y_i^d, y_{i,t-1}^p) m_{tj}^2 \right\} + \gamma M_t + \frac{1}{2} \lambda \sum_j m_{tj}^2 \tag{9}$$

or, equivalently,

$$\tilde{\mathcal{L}}_t = \sum_j \left\{ G_j m_{tj} + \frac{1}{2} (H_j + \lambda) m_{tj}^2 \right\} + \gamma M_t \tag{10}$$

where G_j and H_j represent the sums over $g(\cdot,\cdot)$ and $h(\cdot,\cdot)$ for all observations in z_j. Note that the summation in the above is over j, which indexes the leaves in the tree.

For a given tree structure, we can find the optimal prediction values by minimizing the objective function with respect to m_{tj}, which yields $m_{tj}^* = -G_j/(H_j + \lambda)$. With a squared-error loss function, where $g(y_i^d, y_{i,t-1}^p) = -2(y_i^d - y_{i,t-1}^p)$ and $h(y_i^d, y_{i,t-1}^p) = 2$, the optimal prediction values are effectively the (penalized) average of the residuals associated with a given leaf. We can then substitute the optimal prediction values into the objective function to obtain the following:

$$\tilde{\mathcal{L}}_t^* = -\frac{1}{2} \sum_j \frac{G_j^2}{H_j + \lambda} + \gamma M_t \, , \tag{11}$$

which gives us the minimal objective function for a given tree structure. Ideally, one would enumerate all possible tree structures and then use the tree associated with the smallest value of the (minimized) objective function, but this is often intractable. XGBoost instead implements an alternative approach that seeks to minimize the objective at a given iteration by greedily optimizing one level of the tree at a time.

To this end, consider splitting a node in a given tree into "left" and "right" child nodes based on some candidate splitting rule. Let G_L and G_R represent G_j for the left and right child nodes, respectively. Similarly, let H_L and H_R represent H_j for the child nodes. We can then find the reduction in the objective function as follows:

$$\Delta \tilde{\mathcal{L}}_t^* = \frac{1}{2}\left\{\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right\} - \gamma \tag{12}$$

where the first two terms in parentheses capture the loss associated with the child nodes and the third term captures the loss associated with the parent node being split. When splitting a node in any given tree, XGBoost then evaluates Eq. (12) for every possible split rule for every available explanatory variable and chooses the split associated with the maximal reduction in the objective function.[12] XGBoost then continues to split nodes in a given tree until a stopping criterion is met (e.g., a user-specified maximum tree depth). Once a given tree is built, XGBoost then moves to building the next tree in the ensemble and continues to do so until reaching the user-specified number of trees to build.

The above references several hyperparameters that must be chosen by the user, including the hyperparameters in the regularization term (i.e., γ and λ), the maximum tree depth, and the number of trees in the ensemble. There are two additional hyperparameters that can be used to further prevent overfitting. First, one can use column or feature subsampling where each tree in the ensemble is built based on a (uniformly) random subset of all explanatory variables. This procedure can also speed up the algorithm because it reduces the number of covariates that must be searched across when creating new splits. Second, one can use shrinkage by scaling the predictions associated with each tree by a factor η, which is often called the learning rate. The predicted value of the outcome at iteration t is then given by $y_{i,t}^p = y_{i,t-1}^p + \eta f_t(x_i)$. Shrinkage reduces the influence of each tree so that no individual tree has too much influence on the ensemble prediction.

One possible approach to hyperparameter selection is to use a grid search where the user (1) specifies the domain of the hyperparameters, (2) applies the model to every combination of hyperparameters, and (3) selects the hyperparameters that perform the best in terms of some metric (e.g., mean squared prediction error in the validation dataset). Grid search, however, is computationally inefficient because hyperparameter proposals are uninformed by previous evaluations. We instead use a Bayesian hyperparameter optimization procedure, which sequentially updates a probability model that relates the hyperparameter space to the performance metric and then uses the model to intelligently explore the hyperparameter space (Bergstra et al., 2011). Specifically, our implementation of gradient boosting relies on the open-source library XGBoost, which we combine with the Hyperopt library that conducts Bayesian hyperparameter optimization using the tree-structured Parzen estimator.

Finally, regarding our application to the Mexican data, recall that mapping procedures entail

[12] This is known as the "exact greedy algorithm." XGBoost also has an approximate algorithm that can be used for large datasets. See Chen and Guestrin (2016) for details.

an estimation stage and a prediction stage. For the machine-learning methods, we perform this procedure at two different levels of analysis, namely the PSU and municipality levels. Our focus is nevertheless on obtaining estimates of poverty rates at the municipality level, meaning that when we conduct estimation and prediction at the PSU level, we further aggregate the predictions to the municipality level using population-weighted averaging. Further, recall that we have two alternative sets of covariates available, including geo-referenced and census-based variables. Given that the geo-referenced data is only available at the municipality level, we conduct estimation and prediction at the municipality level for any implementation that uses these covariates. When using census-based covariates, however, we consider implementations at both the PSU and municipality levels.[13]

3.2 Traditional small-area methods

We consider three alternative traditional methods for obtaining small area estimates of poverty as a way to benchmark the performance of the non-parametric machine-learning approach.[14] Specifically, we consider area-level models, unit-level models, and a method that attempts to combine unit- and area-level models. First proposed by Fay and Herriot (1979), area-level models conduct estimation and prediction using data aggregated to the geographic area of interest (e.g., the PSU or municipality level).

The area-level model consists of two basic parts. The first part of the model links the actual or true poverty rates y_i^a for the i^{th} entity to a vector of area-level covariates x_i as follows:

$$y_i^a = x_i\beta + \varepsilon_i \tag{13}$$

where β is a vector of parameters to be estimated and ε_i represents random area-level effects that are assumed to be mean zero with a constant variance σ_ε^2. Since we lack information on the true poverty rates, this model cannot be immediately fit.

Instead, direct estimates of poverty rates from survey data $\hat{y}_i^d$ are used as the outcome variable. These estimates are nevertheless subject to sampling error because they are based on the area-specific sample data. The second part of the model, thus, assumes that the direct estimates are centered around the true poverty rates as follows:

$$\hat{y}_i^d = y_i^a + \omega_i \tag{14}$$

[13] In all cases, we pass every eligible variable to the machine-learning model and let the algorithm conduct variable selection.

[14] For a thorough review of small area estimation methods refer to Rao and Molina (2015), for a review of methods in the context of poverty mapping refer to Molina et al. (2022) as well as reviews by Pfeffermann (2013) on other developments in the area.

This model is referred to as the sampling model, and is assumed to have heteroscedastic error variance $\text{var}(\omega_i|y_i) = \sigma^2_{\omega,i}$. These variances are assumed to be known although in reality these are estimated using the survey data. The best linear unbiased predictor (BLUP) is obtained by predicting y_i through the model, $\hat{y}^d_i = x_i\hat{\beta} + \hat{\varepsilon}_i$. The $\hat{\beta}$ are the weighted least squares estimator of β under the Fay-Herriot model and, $\hat{\varepsilon}_i = \psi_i\left(\hat{y}^d_i - x_i\hat{\beta}\right)$ is also the BLUP of the area effect ε_i. Additionally, $\psi_i = \sigma^2_u / \left(\sigma^2_u + \sigma^2_{\omega,i}\right)$. In essence, the final estimates for areas contained in the survey are the weighted average between the direct estimate $\hat{y}^d_i$ and the synthetic estimator $x_i\hat{\beta}$, where the weights, ψ_i and $(1 - \psi_i)$, are determined by the quality of the model and the quality of the direct estimator. For the non-sampled entities, the area-level model simply uses the synthetic estimates to produce predictions. As Molina et al. (2022) notes, normality assumptions are not required for the estimates to be BLUP, but assumptions are needed for estimation of the mean squared error (MSE). Finally, similar to the machine-learning approach, the area-level model can be implemented at the PSU or municipality levels.[15]

Regarding unit-level models, the well-known methods of Elbers et al. (2003) and Molina and Rao (2010) are based on the nested error model originally proposed by Battese et al. (1988). This model assumes that (transformed) household income relates linearly to a vector of household-level covariates according the following model:

$$w_{iv} = x_{iv}\delta + \mu_i + \xi_{iv} \tag{15}$$

where w_{iv} denotes the transformed equivalized income of household v in location i, x_{iv} represents a vector of household-specific characteristics, δ is a vector of coefficients on the household characteristics, μ_i is location-specific random effect such that $\mu_i \sim N(0, \sigma^2_\mu)$, and ξ_{iv} is a household-level error term such that $\xi_{iv} \sim N(0, \sigma^2_\xi)$. The two errors are assumed to be independent from each other.[16] We use a particular version known as the "census empirical best" method (CensusEB), which is a variant of the "empirical best" method of Molina and Rao (2010).[17]

It is evident that the dependent variable of the model is not the poverty status of a given household. Instead, the method attempts to replicate the welfare distribution based on an assumed data generating process. The parameter estimates obtained when fitting model (15) to the survey data ($\hat{\delta}, \hat{\sigma}^2_\mu$, and $\hat{\sigma}^2_\xi$) are applied to the census household level data to generate a simulated value of w_{iv} for each household in the census. From the simulated census welfare vector it is possible to obtain estimates of any welfare indicator, not just poverty headcount rates.

The parameter estimates are fit through a suitable method such as REML or Henderson's Method

[15] Estimates at the PSU level may be aggregated to a higher level, see Rao and Molina (2015) for more details.

[16] Elbers et al. (2003) offer the option of accommodating for non-normally distributed errors – however, despite this flexibility Corral et al. (2021b) illustrate how even under non-normality the method lags Molina and Rao's (2010) EB approach.

[17] As opposed to empirical best estimates, census empirical best estimates do not include survey observations when calculating area-level poverty rates. Corral et al. (2021b) show that when the sample size is small relative to the population the difference between the two methods is negligible. For additional discussion, see Molina (2019).

III (Henderson 1953).[18] The w_{iv} for each household is simulated M times using Monte Carlo simulation where each simulated welfare m follows:

$$w^*_{ivm} = x_{iv}\hat{\delta} + \mu^*_{im} + \xi^*_{ivm}$$

where μ^*_{im} and ξ^*_{ivm} are drawn from their assumed distributions.[19] Here, μ^*_{im} is generated as $\mu^*_{im} \sim N(\hat{\mu}_i, \hat{\sigma}^2_\mu(1-\hat{\kappa}_i))$ for sampled municipalities, where $\hat{\mu}_i = \hat{\kappa}\left(\bar{w}_i - x_{iv}\hat{\delta}\right)$ and $\hat{\kappa}_i = \hat{\sigma}^2_\mu / \left(\hat{\sigma}^2_\mu + \hat{\sigma}^2_\xi / n_i\right)$, n_i is the sample size for municipality i. For municipalities that are not part of the sample, then μ^*_{im} is generated as $\mu^*_{im} \sim N(0, \hat{\sigma}^2_\mu)$ which corresponds to the method used in ELL for all municipalities regardless of whether or not they are in the sample. Thus, EB makes more use of the survey data for predictions corresponding to sampled areas. Note that EB methods only ensure that the area level means of the dependent variable are Empirical Best Linear Unbiased Predictors (EBLUP); it does not guarantee the same for poverty which is a non-linear parameter. Hence, the importance of replicating the welfare distribution.

One major aspect of concern for small area estimation is the estimation of noise. A statistical agency in a given country may seek to undertake a small area estimation application because the quality of the estimates derived from survey data may not be sufficiently precise or estimates may not even be possible since the area was not sampled. Hence, a big concern surrounding small area estimation is the estimation of MSEs for each area's estimate. Estimates of noise are obtained via bootstrap replications where the model's assumptions are exploited in order to obtain an estimated MSE (González-Manteiga et al., 2008).[20] This aspect is one of the biggest differences between poverty maps based on small area estimation and those based on machine learning. Statistical agencies across the globe will assess an estimate's noise before deciding to publish it or not, thus the importance of such estimates.

The key assumption of the unit-level model is that the distribution of the explanatory variables is similar between the survey used for estimation and the census used for prediction. This assumption is generally violated if the census is outdated – which is often the case in developing countries – and in this event the application of the unit-level model will lead to similarly outdated poverty maps (Lange et al., 2018). Importantly, machine-learning approaches to poverty mapping are commonly motivated by the fact that censuses are outdated and are thus viewed as a way to produce timely poverty maps in a data-constrained environment. Stated differently, given the differing data requirements between the unit-level model and the machine-learning approach, the two methods are not generally considered to be direct competitors. The unit-level model nevertheless serves as a useful performance benchmark, as it is often viewed as the "gold standard" of poverty mapping.

Unlike unit-level models, unit-context models were specifically developed to be used in off-census

[18] Henderson's method III is coupled with GLS to obtain the regression coefficients.

[19] See Corral et al. (2021b) for a more detailed description.

[20] For a much more detailed discussion on the estimation of noise refer to Rao and Molina (2015) and for a discussion focused on ELL and EB estimates refer to Corral et al. (2021b) and Corral et al. (2022).

years. Initially developed by Nguyen (2012) and further refined by Masaki et al. (2020), these models are based on a simple modification of the unit-level approach: rather than modeling household-level income as a function of household-level characteristics, unit-context models instead use only area-level covariates from the census. Specifically, the unit-context model replaces the survey-based covariates x_{iv} in Eq. (15) with census aggregates at the municipality and the PSU level, x_i and x_{ic}, and thus relaxes the assumption that the survey-based covariates match the moments of those from the census. Like the area-level model, the unit-context model is a direct competitor to the machine-learning approach because it is feasible to implement with the same data. For this reason, we use the unit-context model as reference model as we evaluate the performance of machine-learning methods. However, it is worth noting that the unit-context model has been shown to produce biased and noisy poverty estimates, largely due to its inability to explain between-household variation in income (Corral et al., 2022).

4 Model Validation

The coefficient of determination or R^2 is the most commonly used metric for validating poverty maps generated using machine-learning methods (see Jean et al. (2016), Pokhriyal and Jacques (2017), Steele et al. (2017), Lee and Braithwaite (2020), Yeh et al. (2020), or Chi et al. (2022)). In this section, we first critically assess this validation approach by discussing three ways in which the R^2 can be misleading in terms of model assessment. We then argue that a better alternative is to validate poverty estimates using the mean squared error, which is not subject to many of the limitations of the R^2. Finally, we outline another validation procedure that focuses more directly on what is of ultimate concern with poverty mapping, namely informing the targeting of resources for the purposes of poverty alleviation.

Let y_i^a denote the actual or true poverty indicator for the i^{th} geographical entity. Further, let y_i^d and y_i^p denote the direct estimate and model-based prediction of the poverty indicator, respectively. The standard method for assessing the predictive performance of model-based poverty estimates calculates the R^2 from a regression like the following:

$$y_i^d = \alpha + \delta y_i^p + \zeta_i \tag{16}$$

where α and δ are the regression coefficients and ζ represents the error term.[21] The R^2 for the above regression can be written as follows:

$$R_d^2 = \frac{\sigma_d^2 - \sigma_\zeta^2}{\sigma_d^2} \tag{17}$$

where σ_d^2 is the variance of the direct estimate and σ_ζ^2 is the variance of ζ. R_d^2 thus conveys

[21] Note that any regression of this form can only be estimated for the sampled regions with direct estimates.

information about how much of the variance of the direct estimate is explained by the model-based prediction.

While unbiased, direct estimates can be imprecise estimates of true poverty indicators (Corral et al., 2021b). Ideally, one would assess the performance of model-based predictions on the basis of the true poverty indicators themselves. That is, model-based predictions are more appropriately assessed using the R^2 from the following regression:

$$y_i^a = \alpha + \delta y_i^p + \upsilon_i \tag{18}$$

where υ represents the error term. The R^2 for this regression can written as

$$R_a^2 = \frac{\sigma_a^2 - \sigma_\upsilon^2}{\sigma_a^2} \tag{19}$$

where σ_a^2 is the variance of the true poverty indicator and σ_υ^2 is the variance of υ. We are specifically interested in characterizing the bias of R_d^2 when it is used to assess true quantity of interest, R_a^2.

Let $y_i^d = y_i^a + \omega_i$ where ω is a mean-zero error term with variance σ_ω^2. That is, we view this as a classical measurement error problem where the direct estimate of the poverty indicator is a noisy estimate of the true poverty indicator. Note that the preceding equation implies that $\zeta_i = \upsilon_i + \omega_i$ and $\sigma_\zeta^2 = \sigma_\upsilon^2 + \sigma_\omega^2$. Following Majeske et al. (2010), we then define the bias as:

$$B = R_a^2 - R_d^2 \,. \tag{20}$$

Substituting Eqs. (17) and (19) into our expression for the bias, we have:

$$B = \frac{\sigma_a^2 - \sigma_\upsilon^2}{\sigma_a^2} - \frac{\sigma_d^2 - \sigma_\upsilon^2 - \sigma_\omega^2}{\sigma_d^2} = \frac{\sigma_\omega^2(\sigma_a^2 - \sigma_\upsilon^2)}{\sigma_d^2\sigma_a^2} = \frac{\sigma_\omega^2}{\sigma_d^2} R_a^2 \,. \tag{21}$$

Finally, we can substitute Eq. (21) into Eq. (20) and rearrange, which gives

$$R_d^2 = R_a^2 \left(\frac{\sigma_d^2 - \sigma_\omega^2}{\sigma_d^2} \right) \tag{22}$$

and establishes a basic relationship between R_d^2 and R_a^2. See Majeske et al. (2010) for additional mathematical details.

In Eq. (22) above, we see that $R_d^2 = R_a^2$ when $\sigma_\omega^2 = 0$ or, equivalently, when $y_i^d = y_i^a$ for all i. In the presence of measurement error or when $\sigma_\omega^2 > 0$, we then see that R_d^2 will be biased downward and the magnitude of the bias will be determined by the size of $(\sigma_d^2 - \sigma_\omega^2)/\sigma_d^2$. Validation exercises based on R_d^2 will thus tend to mischaracterize the performance of model-based approaches to poverty mapping, with the resulting estimates of R^2 being biased downward. One implication of the above is that R_d^2 is context-dependent and will generally be influenced by the design of the survey data (e.g., larger sample sizes will tend to reduce measurement error, all else equal). In particular, a poorly performing model applied to a low bias setting may yield an estimate of R_d^2 that is higher than an estimate of R_d^2 from a strong model applied to a high bias setting. The presence of bias in the measures of R^2 can thus reverse the rank ordering of models when comparisons are made across contexts.[22]

There is an additional form of bias in R^2 that can lead to overoptimistic conclusions related to model validity. To illustrate this, note that R_a^2 in Eq. (19) is mathematically equivalent to the squared (Pearson) correlation coefficient between y_i^a and y_i^p. A well-known property of the correlation coefficient is that it is invariant to any positive affine transformation of its arguments, implying that the *squared* correlation coefficient is invariant to *any* affine transformation (Gujarati, 2003).[23] We can thus write

$$R_a^2(y_i^a, y_i^p) = R_a^2(y_i^a, q_1 + q_2 y_i^p) \tag{23}$$

where q_1 and q_2 are some arbitrary constants. The implication of the above is that the R^2 is insensitive to systematic bias in the model-based poverty estimates. For example, consider some unbiased poverty estimate y_i^u, meaning that $E(y_i^u - y_i^a) = 0$. Now consider another estimate y_i^b that is identical to y_i^u up to some locational shift, implying that $y_i^b = y_i^u + q$ and $E(y_i^b - y_i^a) = q$. While y_i^b is biased (i.e., it will systematically over- or under-estimate poverty), it will achieve an R^2 that is identical to y_i^u.

A final issue related to R^2 as a performance metric is that it is affected by the variance of the true poverty measures. To see this, consider the following re-expression of R_a^2:

$$R_a^2 = 1 - \frac{\frac{1}{N}\sum_i (y_i^a - \alpha - \delta y_i^p)^2}{\frac{1}{N}\sum_i (y_i^a - \bar{y}^a)^2} = 1 - \frac{\text{MSE}(y_i^a, \alpha + \delta y_i^p)}{\text{Var}(y_i^a)} \tag{24}$$

where N is the number of geographical entities. The above shows that R^2 can be viewed as a normalized version of the mean squared error with the variance of the true poverty indicator used as the normalization factor. This approach to normalization produces an additional source of context-dependence in R^2. Specifically, consider two models that perform equally well in that they achieve the same mean squared error. However, assume that one model is applied to a context where the variance of y_i^a is large and the other is applied to a low variance context. The above then implies that the model applied to the high variance context will achieve a higher R^2, even though the performance of the two models is arguably identical.

Rather than relying on R^2 as a performance metric, one can avoid regression-based model assessment altogether by calculating the mean squared error directly:

$$\text{MSE}(y_i^a, y_i^p) = \frac{1}{N}\sum_i (y_i^a - y_i^p)^2 , \tag{25}$$

[22] See Yeh et al. (2020), Chi et al. (2022), or Aiken et al. (2022) for this type of cross-context comparison of R^2 values.

[23] More specifically, the correlation coefficient is invariant to any *positive* affine transformation because it changes signs when one of its arguments is scaled by some negative constant. The *squared* correlation coefficient is then invariant to any affine transformation.

which avoids the normalization issue and provides a more direct understanding of the expected squared error of the estimates. The mean squared error in Eq. (25) is sensitive to systematic biases in the predictions. Reconsidering the biased and unbiased poverty estimates described above (i.e., y_i^u and y_i^b), we can write $\text{MSE}(y_i^b, y_i^a) = \text{MSE}(y_i^u + q, y_i^a) = \text{MSE}(y_i^u, y_i^a) + q^2$. Systematic biases in the estimates will thus be penalized by the mean squared error. Further, with multiple estimates for each entity, one can calculate entity-specific mean squared errors, which can be decomposed into parts associated with the bias and variance of the estimates. Where y_{ik}^p denotes estimate $k = 1, 2, \ldots, K$ for the i^{th} entity, we have

$$\text{MSE}_i = \frac{1}{K}\sum_k (y_i^a - y_{ik}^p)^2 = \frac{1}{K}\sum_k (y_{ik}^p - \bar{y}_i^p)^2 + (\bar{y}_i^p - y_i^a)^2 \tag{26}$$

where the first term on the right-hand side of the above captures the variance of the estimates for entity i and the second term captures the (squared) bias.[24]

The primary reason for obtaining granular poverty estimates is to improve the targeting of resources. While understanding the error of the poverty estimates is useful in this regard, the ultimate concern is to understand how different methodological choices affect poverty alleviation by influencing the efficiency of targeting. Consequently, we also consider how machine-learning methods perform in terms of reducing poverty in the context of the Census created from the Mexican Intercensal Survey. Similar to Elbers et al. (2007), we propose simulating a hypothetical poverty-alleviation program that distributes resources to households according to a given set of small area estimates.[25] In addition to the estimates, the procedure requires true household-level incomes from a census-type dataset.

The procedure we use in our simulations is as follows:

1. Using the true household-level incomes, calculate the poverty headcount and poverty gap for the country as a whole for some chosen poverty line. Using the poverty gap measure, determine the budget required to completely eradicate poverty.[26] Calculate an individual-level transfer amount by dividing the total budget by the number of people living in poverty.

2. Using a given set of small area estimates, order the municipalities according to their estimated poverty headcounts, from most to least poor. For the poorest municipality, augment the true per capita income of all households by an amount equal to the transfer (i.e., each household receives an amount equal to the transfer times the number of household members). Continue this distribution procedure for the second poorest municipality, the third poorest municipality, and so on until there are not enough resources to cover all households in a given municipality. For the marginal municipality, distribute the remaining budget equally among individuals in that municipality.

3. Using the post-transfer household incomes, calculate an updated poverty headcount to assess the effects of the distribution scheme on poverty alleviation.

[24] See Hastie et al. (2009) for derivation and detailed discussion of the bias-variance decomposition.

[25] See Yeh et al. (2020),Chi et al. (2022), and Aiken et al. (2022) for examples of other studies that have conducted targeting simulations.

[26] This requires knowing who is poor and how poor they are. The transfer is enough to bring every poor individual up to the poverty threshold.

4. Repeat steps 2 and 3 for each estimation method for all 500 sets of small area estimates corresponding to each sample.

The benchmark poverty alleviation from this scheme corresponds to the results of applying the above to the true municipal poverty rankings. The results for each method are compared to that result. In our implementation, we fix the poverty line at the 25th percentile of per capita income prior to transfers. The benchmark poverty rate, after delivering transfers to people in municipalities ranked by their true poverty rate, is 19.9 percent.

5 Results

Recall from Section 4 that the R^2 measure often used to assess the Recall that the R^2 measure often used to assess the performance of machine-learning models is biased downward.[27] To understand the extent of this bias in practice, we have calculated the R^2 associated with gradient boosting when using both the direct estimates and true poverty rates as the basis for performance assessment. We conduct this comparison at two different levels of estimation, namely the PSU and municipality levels. All R^2 calculations are nevertheless made at the municipality level, meaning that the PSU-level model is estimated at that level and then the poverty estimates are aggregated to the municipality level. For each of our 500 samples, we thus calculate four R^2 measures: (1) PSU-level estimates evaluated against the direct poverty estimates, (2) PSU-level estimates evaluated against the true poverty rates, (3) municipality-level estimates evaluated against the direct poverty estimates, and (4) municipality-level estimates evaluated against the true poverty rates. All models use only census-based covariates.[28]

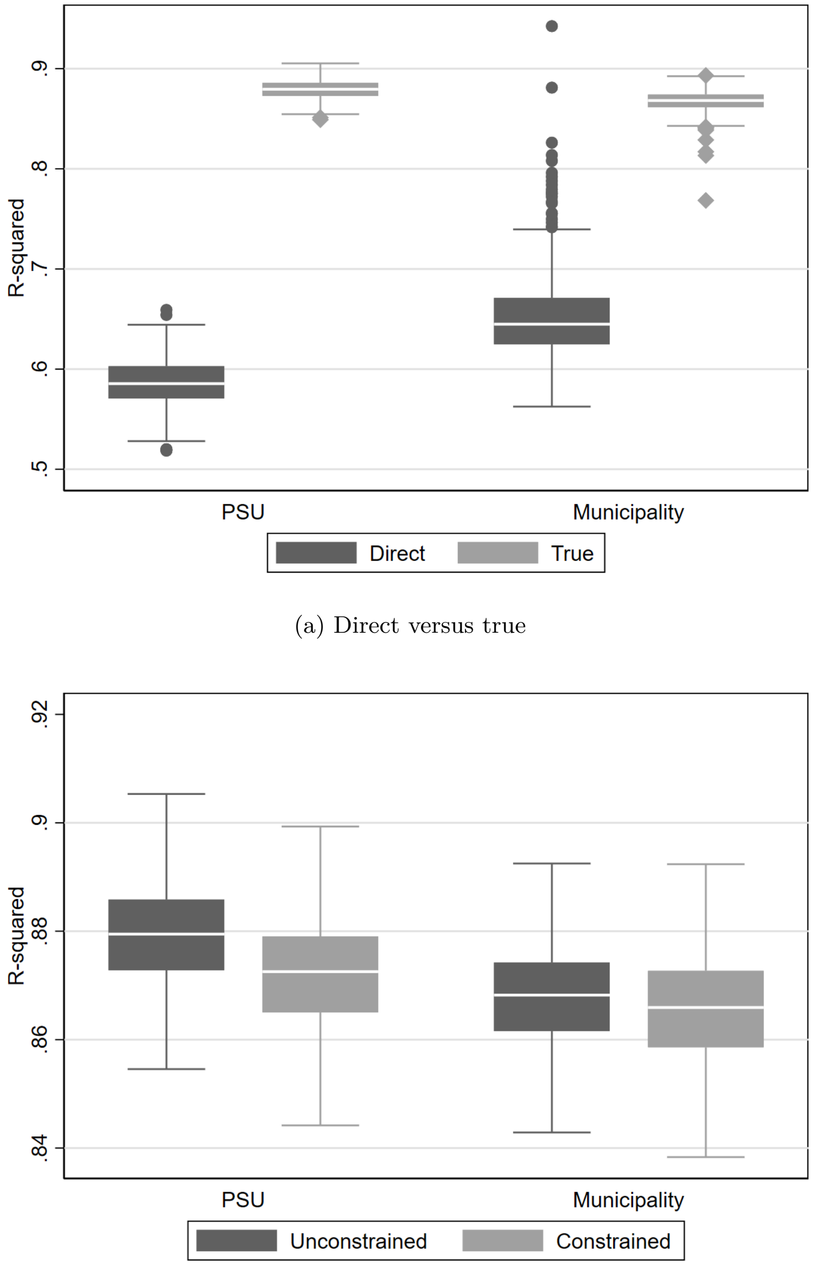
Panel (a) of Figure 1 presents these results and we find evidence of considerable downward bias for both the PSU- and municipality-level models. For example, for the PSU-level model, we find that the median "true" R^2 is 0.88 whereas the median "direct" R^2 is 0.59. Interestingly, the true R^2 tends to be higher for the PSU-level model than the municipality-level model, but the opposite is the case for the direct R^2. This suggests that model selection based on the direct R^2 can be misleading in that the direct R^2 incorrectly suggests that the municipality is the preferred level of estimation. More specifically, we find that the true R^2 identifies the PSU as the preferred level of estimation for 437 of the 500 samples, but the direct R^2 only selects the PSU level for four of the 500 samples. The direct R^2 then identifies the incorrect level of estimation for 433 of the 500 samples.

In Section 4, we discussed two other ways that the R^2 can be misleading for assessing model performance: it is invariant to locational shifts and influenced by the variance of the true poverty rates. Using the Mexican data, we can assess the relevance of the former issue by comparing the

[27] For examples, see Jean et al. (2016), Pokhriyal and Jacques (2017), Steele et al. (2017), Lee and Braithwaite (2020), Yeh et al. (2020), or Chi et al. (2022).

[28] To ensure that our results are not affected by the composition of the sample, we calculate all R^2 measures for each sample using only those municipalities with direct poverty estimates.

Figure 1: R^2 estimates at the PSU and municipality levels



(a) Direct versus true

(b) Unconstrained versus constrained

Note: This figure presents the R^2 from a regression where on the right-hand side we have the model-based estimates and on the left-hand side we either have the true poverty rate or the direct estimates of poverty. Since we have 500 samples drawn from our census, we have 500 sets of model-based estimates and, consequently, 500 R^2 measures.

true R^2 to a constrained version of the true R^2 that fixes the intercept and coefficient to zero and one, respectively. By constraining the R^2 in this manner, we remove its ability to adjust for systematic differences between the poverty estimates and the true poverty rates. Panel (b) of Figure 1 presents the results and we see that, as expected, the unconstrained R^2 overstates the model's performance, albeit only to a small degree. We nevertheless find that even these small differences can lead to model selection issues. In particular, the constrained R^2 identifies the PSU level as the appropriate level of estimation for 355 of the 500 samples whereas the unconstrained version, as mentioned, selects the PSU level for 437 out of the 500 samples. The commonly used unconstrained R^2 then selects the incorrect model for 82 of the 500 samples.

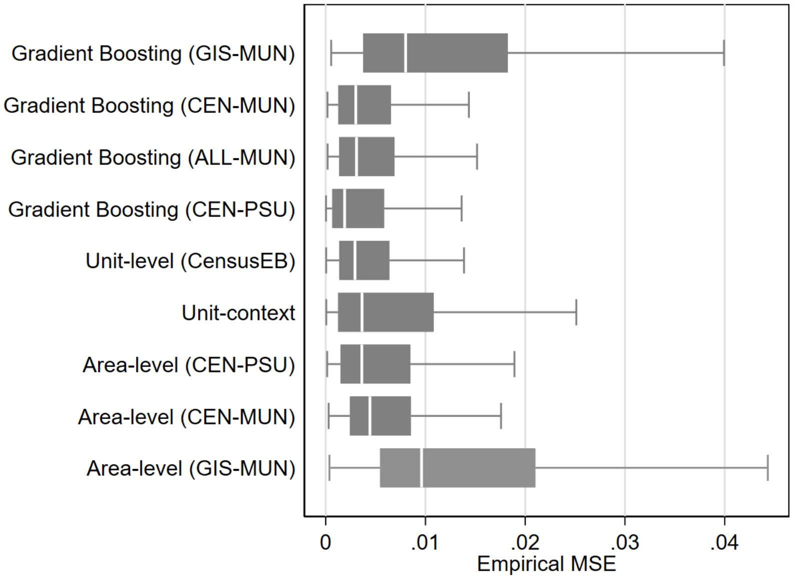
Given the shortcomings of the R^2, we have argued that the empirical MSE of the poverty estimate at the area level represents a better metric for validating the performance of a particular set of estimates. Panel (a) of Figure 2 thus presents our empirical MSE results for several alternative models. The first three models apply gradient boosting at the municipality level using three alternative covariate sets: geo-reference covariates ("GIS-MUN"), census-based covariates ("CEN-MUN"), and then a model with all available covariates ("ALL-MUN"). The fourth model applies gradient boosting at the PSU level and only uses census-based covariates ("CEN-PSU").[29] The final three models are all traditional methods that we use to benchmark the performance of gradient boosting. We present unit-level (CensusEB), unit-context, and area level models. We consider three alternative implementations of the area-level model that, much like for gradient boosting, vary the level of analysis and the covariates used. In particular, the empirical best (CensusEB) model is often considered the "gold standard" of small-area estimation and corresponds to an ideal scenario where up-to-date, micro-level census data are available. For these traditional models, all available covariates enter the models linearly (i.e., we do not include interaction or quadratic terms).

One advantage of the MSE is that it can be calculated for each geographical entity. As such, each box plot in panel (a) of Figure 2 summarizes the distribution of municipality-level MSE estimates rather than the aggregate MSE for each sample (i.e., the results are based on Eq. [26]). It is useful to first consider how gradient boosting performs relative to the traditional models when using the same census-based covariates. As is evident in the results from panel (a) and (b) of Figure 2 the gradient boosting methods using census aggregate data – Gradient boosting (CEN-PSU) and Gradient boosting (CEN-MUN) – perform better than unit-context and Fay-Herriot models, despite using similar data, and approximate the performance of unit-level (CensusEB) estimates. The median MSE for the CensusEB model is 0.003 while that of gradient boosting methods is 0.002 and both methods have similar interquartile ranges. The results suggest that gradient boosting methods can approximate CensusEB unit-level methods and will likely outperform unit-context and Fay-Herriot methods by a considerable margin, which is quite encouraging for cases where estimates are needed when the survey and census data are not contemporaneous.[30]

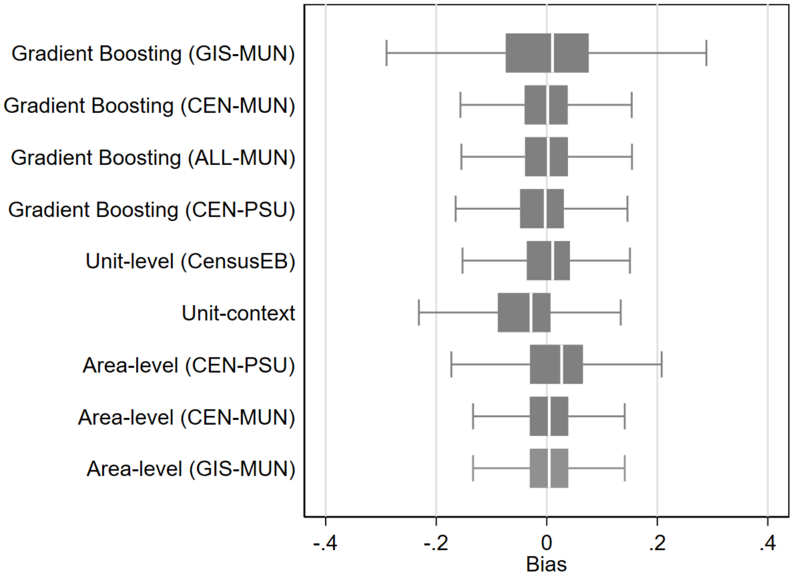
[29] Recall that the geo-referenced covariates are only available at the municipality level.

[30] The unit-context models include covariates at the PSU and municipality level. The location effects are specified at the municipality level.

Figure 2: Empirical design MSE and bias for select models



(a) Design MSE



(b) Design bias

Note: Figures presents the design MSE and bias for each applied method. The design MSE is approximated as the mean squared difference between the model based estimate for a given municipality (obtained from each of the 500 samples drawn from the census) and the municipality's true poverty rate. The bias is the average difference between the model based estimate for a given municipality (obtained from each of the 500 samples drawn from the census) and the municipality's true poverty rate. The box-plots shows the municipality level spread of the design MSE and bias of the method's estimates.

In line with the findings and recommendations from Corral et al. (2022), gradient boosting methods appear to outperform the traditional small area methods recommended for off-census years. The unit-context and Fay-Herriot models are direct competitors to the machine-learning models, which are often motivated by a lack of up-to-date census data. Also aligned to Corral et al. (2022), area-level models outperform unit-context models in terms of bias and MSE.

Now that we have established that in the context of the Mexican data used and with covariates derived from census data, gradient boosting methods can perform just as well as CensusEB methods our attention is turned to gradient boosting's performance under geo-referenced data – Gradient Boosting (GIS-MUN).[31] The gradient boosting model with GIS-based covariates is our closest approximation to the standard implementation of machine learning in the context of poverty mapping, the method is contrasted to an application of an area-level model with similar data – Area-level (GIS-MUN).[32] The median MSE for the gradient boosting model with GIS-based covariates is 0.008 (with a root MSE of about 0.09), meaning that the median municipality's error is roughly nine percentage points. The area-level model using geo-referenced covariates achieves a median MSE of 0.01.

The bias associated with poverty estimates is especially important due to the policy relevance of understanding absolute living standards. In panel (b) of Figure 2, we thus present box plots for the municipality-level bias estimates associated with each model (see Eq. [26] for the bias-variance decomposition of the MSE). The median bias for all models is approximately zero, with the possible exception of the unit-context model and the area-level model estimated at the PSU level.[33] Perhaps the more interesting result is that the bias estimates from the gradient boosting model with geo-referenced covariates exhibits considerable variation, with minimum and maximum biases reaching -0.43 and 0.43, respectively. That is, the standard implementation with remotely-sensed covariates is severely biased for some municipalities, even though its median bias is approximately zero. All other models, including the competing unit-context and area-level models, exhibit much less variation.

The performance of machine learning appears to depend heavily on the covariates used in the model, with census-based covariates considerably improving model performance. Another way to see that the census-based covariates have greater predictive power than the remotely-sensed covariates is by looking at feature importance. One popular measure of feature importance summarizes the gain (i.e., reduction of the loss function) associated with any given feature. That is, for a given run of the model, feature importance captures the average gain of a given feature, where the gains are averaged across all trees in the model.[34] Figure 3 plots the 20 covariates with the highest feature importance

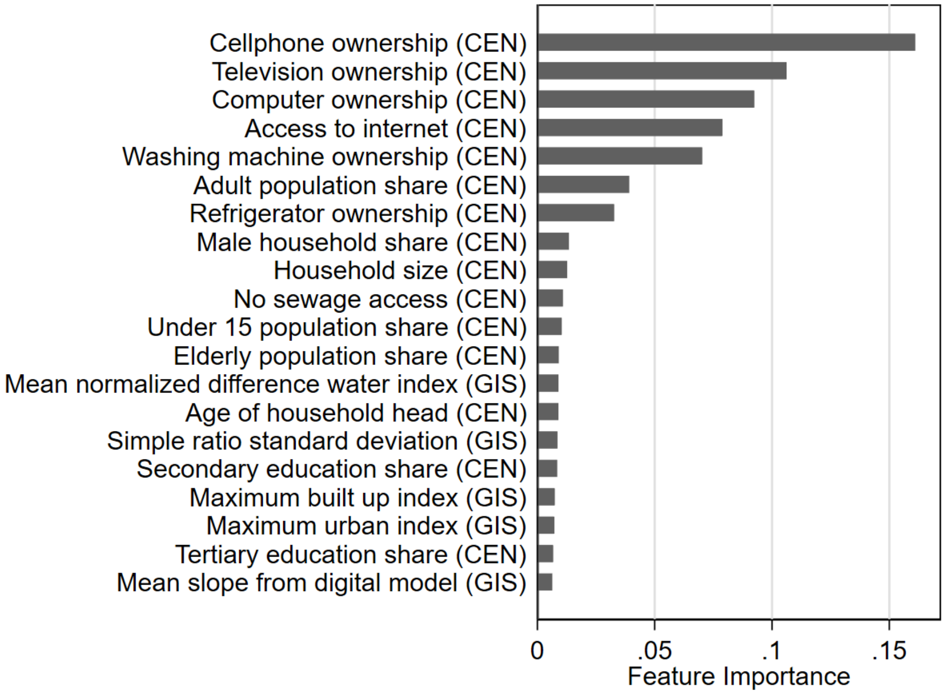
[31] Note that the underlying model for CensusEB estimates relies on household level microdata. Consequently, the data requirements for CensusEB are even higher.

[32] A notable aspect of Fay-Herriot and Gradient Boosting models is their relative ease of application and the fact that these usually take considerably less time in implementing as opposed to CensusEB and unit-context models.

[33] The same issue for unit-context models is highlighted by Corral et al. (2021a) and Corral et al. (2022).

[34] In addition, the average gains for a given run of the model are normalized by dividing each by the sum of all average gains for that run.

Figure 3: Feature importance



Note: Figure illustrates the average feature importance across all 500 models fit to the 500 samples of the census data. The importance corresponds to the reduction of the loss function associated to each feature.

after running gradient boosting with all covariates on each sample and averaging each covariate's importance across the 500 samples. We find that all of the top 10 covariates are census-based, with only five geo-referenced covariates entering the top 20.

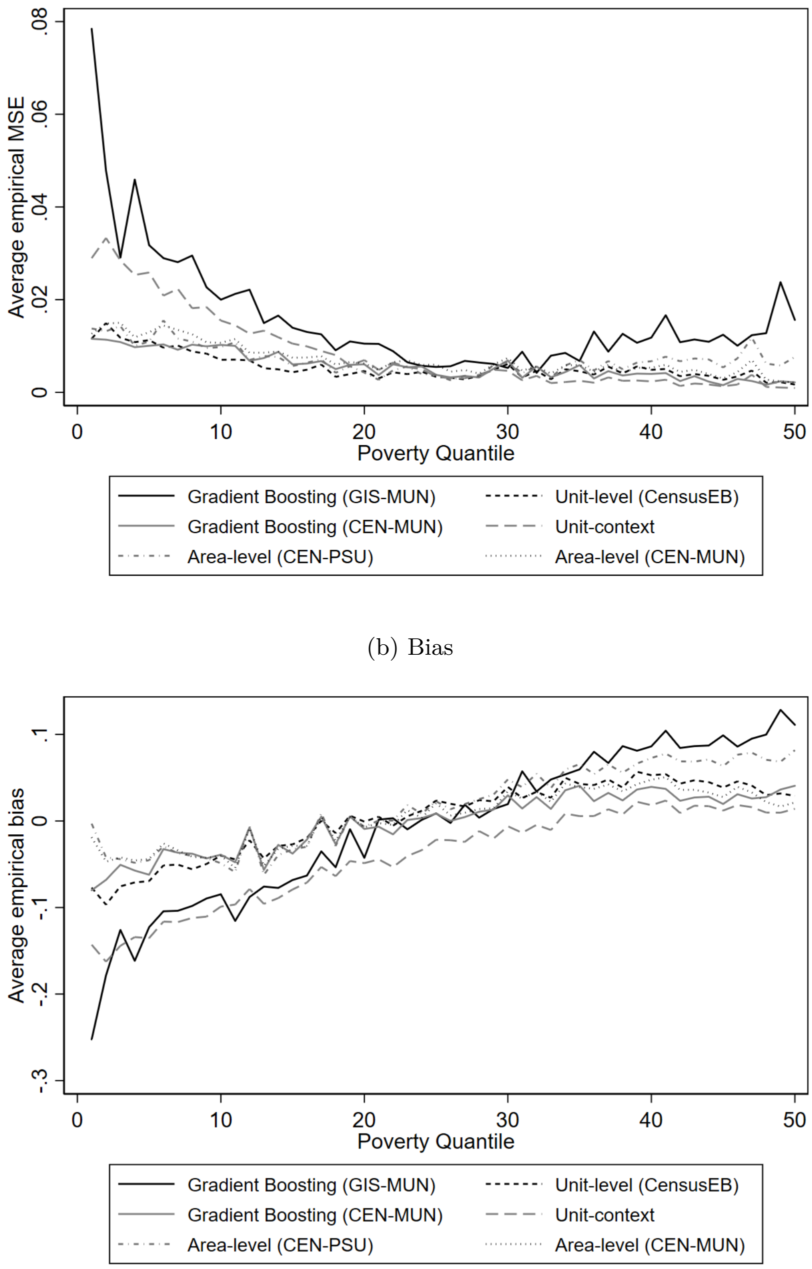
Figure 4 presents MSE and bias estimates for various models across different "poverty quantiles." That is, we order all municipalities by their true poverty rates, divide them into 50 quantiles, and calculate the average MSE and bias for each quantile. For the sake of simplicity, we only present results for two machine-learning models – the municipality-level models using only geo-referenced and census-based covariates – and select traditional methods. Panel (a) presents the MSE results and panel (b) presents the bias results. In both plots, gradient boosting with census-based covariates performs quite similarly to the "gold standard" unit-level model across all quantiles. However, gradient boosting with geo-referenced covariates again departs considerably from this benchmark, particularly at the lowest and highest quantiles. The bias results are particularly notable: the model with geo-referenced covariates systematically underestimates poverty at the lowest quantiles and overestimates poverty at the highest quantiles. The standard implementation thus understates the spatial distribution of poverty.

An additional consideration is how the methods perform for predictions which are out of sample. Because the experiment consists of 500 samples taken from the census data, a given municipality may be present in one of the samples and may be absent in others. Consequently, to evaluate out of sample properties we rank municipalities by their likelihood of selection across the 500 samples. The likelihood of being present ranges from 0.09 to 1, with a median of 0.48 and a mean of 0.53. Among the 20 percent least likely to be sampled municipalities, the gradient boosting model with GIS-based covariates is the worst performing in terms of MSE followed closely by unit-context models (Figure 5). In terms of bias among the 20 percent of municipalities that are least likely to be sampled, the gradient boosting model with the geo-referenced covariates shows very wide variability, while unit context models show downward bias. The other gradient boosting models perform similarly to CensusEB models in terms of bias and MSE. The performance, in terms of MSE, of the different methods seems to improve with the likelihood of being selected. Among municipalities most likely to be selected, the gradient boosting model with the geo-referenced covariates is still the worst performing (Figure 6). However, in terms of bias, among the most likely to be selected municipalities, the bias of Fay-Herriot methods at the municipality level is the best performing illustrating the importance of the methods EBLUP.

While we have focused on the performance of gradient boosting up to this point, there are nevertheless several other machine-learning models that can be used for poverty mapping. Leading alternatives include the least absolute shrinkage and selection operator (lasso) (Tibshirani, 1996), the random forest (Breiman, 2001), and Bayesian additive regression trees (BART) (Chipman et al., 2010). While the lasso is a simple way to conduct variable selection and regularization in the context of linear regression, the random forest and BART models are similar to gradient boosting in that they all rely on regression tree frameworks. We ask whether these alternative models can perform better than gradient boosting within the confines of the standard implementation, which

Figure 4: Empirical average MSE and bias by poverty quantiles for select models

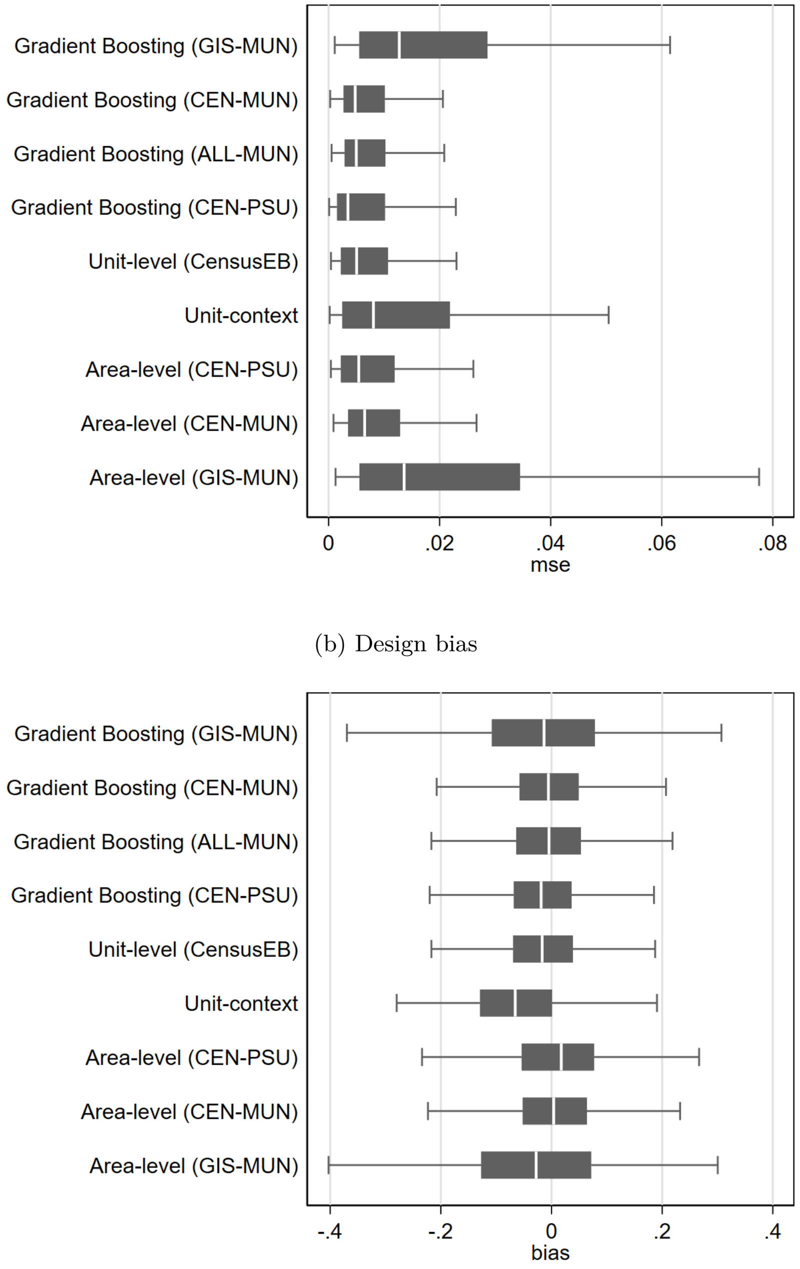
(a) MSE



(b) Bias

Note: Figure illustrates the average empirical bias and MSE for municipalities corresponding to each quantile. Municipality quantiles correspond to the ordering of municipalities by their true poverty rate.

Figure 5: Empirical design MSE and bias with alternative machine-learning models among 20 percent least likely to be sampled municipalities

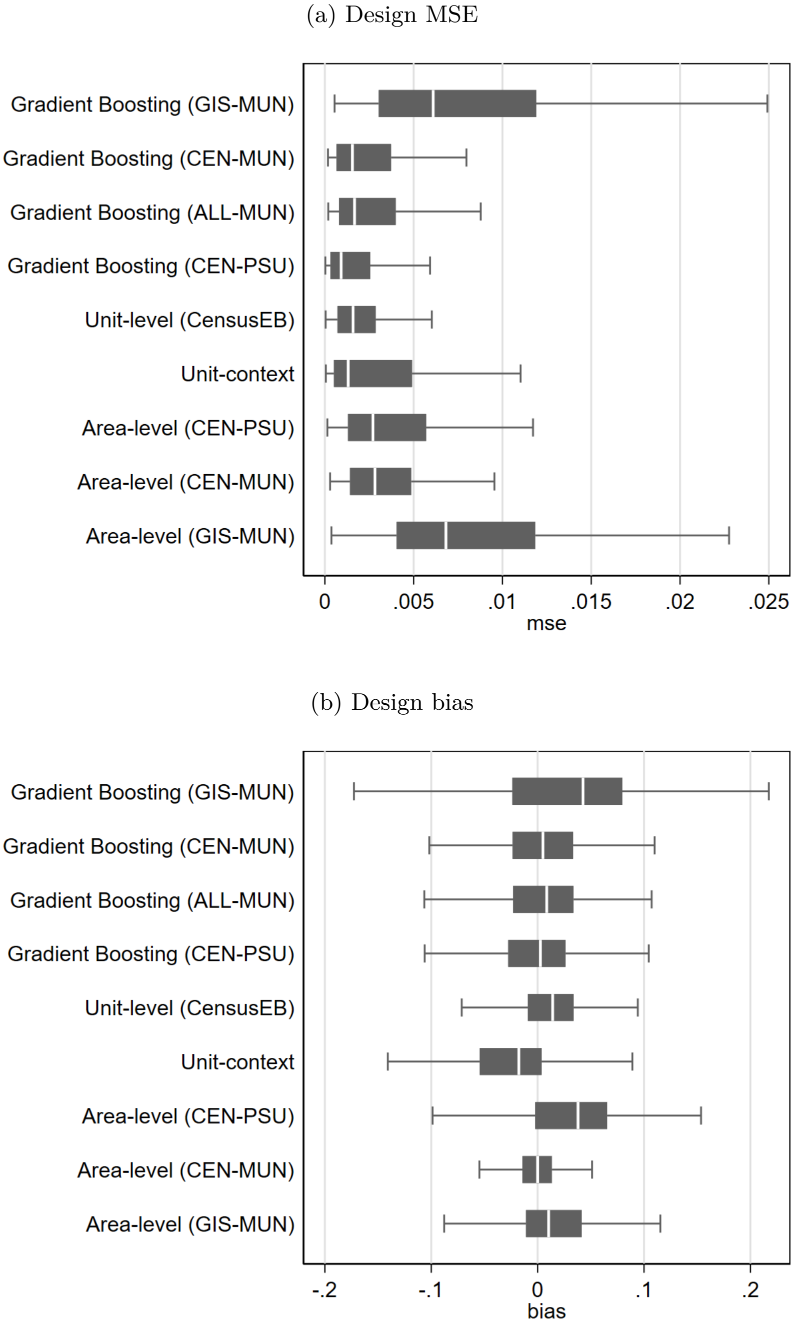


(a) Design MSE

(b) Design bias

Note: Figures presents the design MSE and bias for each applied method. The design MSE is calculated as the mean squared difference between the model based estimate for a given municipality (obtained from each of the 500 samples drawn from the census) and the municipality's true poverty rate. The bias is just the average difference between the model based estimate for a given municipality (obtained from each of the 500 samples drawn from the census) and the municipality's true poverty rate. The box-plots shows the municipality level spread of the design MSE and bias of the method's estimates for the least likely to be sampled municipalities.

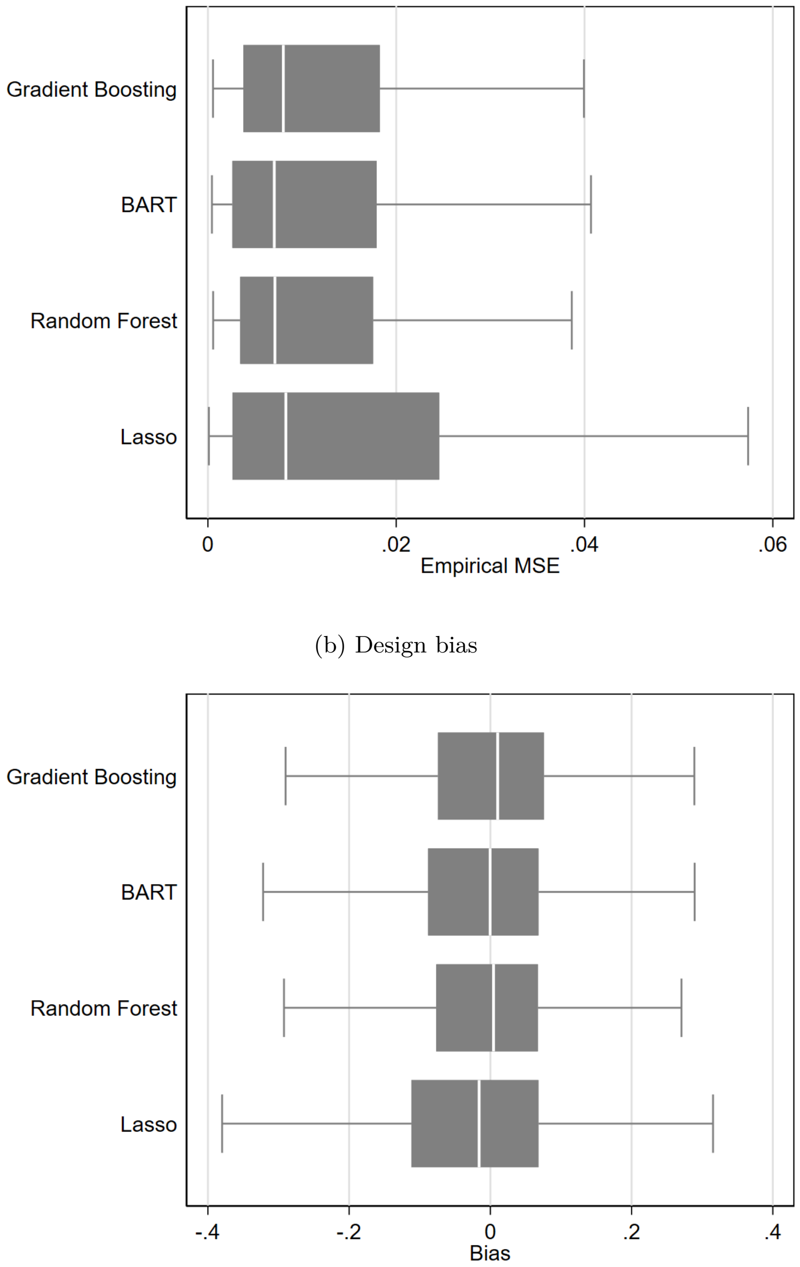
Figure 6: Empirical design MSE and bias with alternative machine-learning models among 20 percent most likely to be sampled municipalities


(a) Design MSE

(b) Design bias

Note: Figures presents the design MSE and bias for each applied method. The design MSE is calculated as the mean squared difference between the model based estimate for a given municipality (obtained from each of the 500 samples drawn from the census) and the municipality's true poverty rate. The bias is just the average difference between the model based estimate for a given municipality (obtained from each of the 500 samples drawn from the census) and the municipality's true poverty rate. The box-plots shows the municipality level spread of the design MSE and bias of the method's estimates for the most likely to be sampled municipalities.

Figure 7: Empirical design MSE and bias with alternative machine-learning models

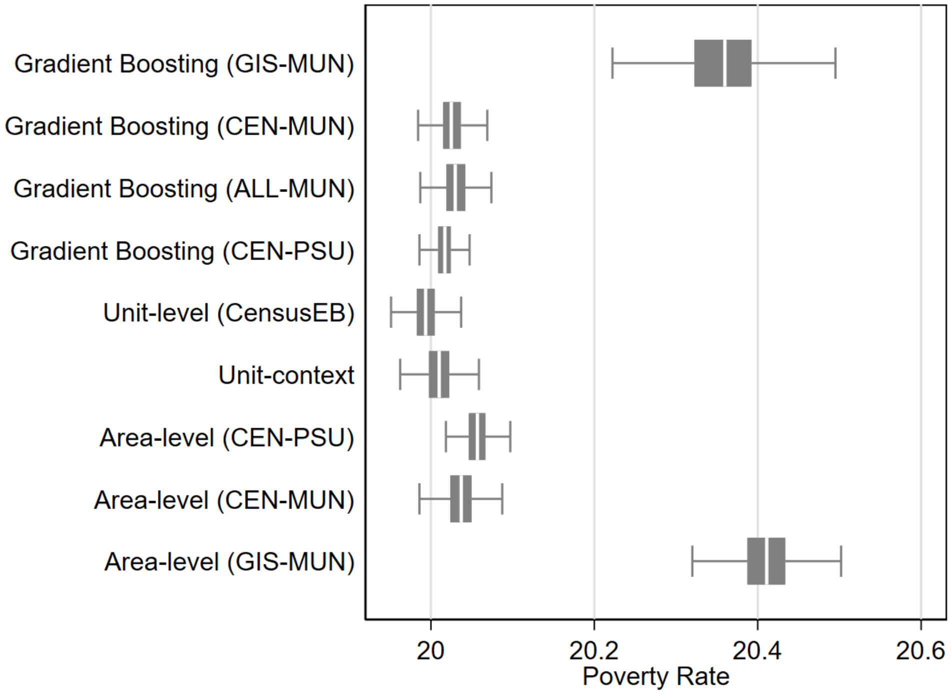


(a) Design MSE

(b) Design bias

Note: Figures presents the MSE and bias for each applied method. The MSE is calculated as the mean squared difference between the model based estimate for a given municipality (obtained from each of the 500 samples drawn from the census) and the municipality's true poverty rate. The bias is just the average difference between the model based estimate for a given municipality (obtained from each of the 500 samples drawn from the census) and the municipality's true poverty rate. The box-plots shows the municipality level spread of the MSE and bias of the method's estimates. All methods rely on GIS covariates only and are fit at the municipality level.

Figure 8: Poverty targeting



Note: Figure illustrates the poverty reduction potential of relying on estimates from each method to rank municipalities and prioritize the poorest municipalities in a cash transfer scheme described in section 4. The box-plot show the spread of the 500 resulting targeting simulations conducted with each methods estimates.

relies exclusively on remotely-sensed covariates. Panels (a) and (b) of Figure 7 present the MSE and bias results for all models. With the possible exception of the lasso, we find that all models perform comparably, meaning that the performance of gradient boosting when using geo-referenced covariates cannot be remedied by simply adopting a different machine-learning model.

Finally, recall from Section 4 that we argued that the ultimate concern with poverty mapping is to understand how different methodological choices affect poverty alleviation. Figure 8 thus compares the machine-learning and traditional approaches in terms of their ability to reduce poverty in the context of our targeting simulations. In these simulations, a hypothetical poverty-alleviation program targeted on the basis of *true* municipality-level poverty rates is able to reduce the overall poverty headcount to just under 20 percent. While the achieved poverty rates of all models fall within the 20 percent range, we find that gradient boosting with geo-referenced covariates is one of the worst performing models. With the exception of the area-level model using geo-referenced covariates, the standard implementation is thus outperformed by all traditional methods, including those that are direct competitors. However, when incorporating census-based covariates, we find that gradient boosting achieves poverty rates that are comparable to the traditional methods, yet again showing that an exclusive reliance on remotely-sensed data can limit the performance of the machine learning.

The results of the targeting simulation are relevant to cases where targeting is based solely on the ranking of municipalities – all methods appear to do a solid job. Nevertheless, there are many instances where the targeting mechanism may rely on the method's poverty estimates. For example, Senegal's Unique National Registry (RNU - in French) of vulnerable households relies on actual estimates from poverty maps to determine the quotas for sampling by municipality (Ferre 2018). It is in those cases where a method that yields unbiased and precise estimates should be employed.

6 Conclusion

This paper sets out to validate traditional small area estimation and machine learning approaches for poverty mapping. Machine learning approaches have become popular across the globe for producing highly disaggregated estimates of poverty which often feed into policy making. The methods are particularly popular in the most data deprived scenarios where census data are not frequently collected, and even when collected may not be contemporaneous to a household survey. Hence, the methods are also a popular alternative for cases where the census data is too old for use under a traditional unit-level small area estimation exercise. To the best of our knowledge, a rigorous validation of machine learning methods like the one done here, where machine learning methods are compared to traditional small area estimates and to a true welfare measure, has not been done before.

We first showed analytically that that the R^2 can be a misleading measure of model performance: it is biased downward when direct estimates are used as the reference measure of model performance, it is insensitive to differences in location and scale between the reference measure and the model predictions, and it is influenced by the variance of the reference measure. One practical issue this raises is that it can be misleading to compare the performance of poverty mapping methods across different datasets, as is done in Yeh et al. (2020), Chi et al. (2022), and Aiken et al. (2022), among others. Most notably, survey design elements can affect measurement error in the direct estimates (e.g., through different sample sizes), which in turn influences the magnitude of bias in the R^2. All else equal, models applied to low-bias settings will appear to perform better than those applied to high-bias settings.

Our simulations built on these analytical results. First, we showed empirically that the direct R^2 exhibits considerable downward bias, with the true R^2 being roughly 35 to 50 percent higher than the direct R^2. We also showed that this bias has implications for model selection, as the direct R^2 commonly identified the incorrect level of estimation in our data. Second, we assessed the performance of machine learning methods and found that they may outperform or achieve similar bias and MSE as the best performing traditional small area estimation approaches, particularly in out-of-sample predictions. Third, we present a comparison on the potential benefits of publicly available GIS covariates. We show how GIS covariates under the data used here do not outperform census aggregate data. This result suggests that there may be considerable gains in linking census and administrative data for poverty mapping in off census years. Fourth, we expand the validations from Corral et al. (2022) and illustrate how Fay-Herriot models are still the preferred approach for off-census years among traditional small area estimation methods. Finally, we present evidence on how, despite different performance across models, all models are beneficial for targeting. While some models may outperform others in their potential poverty reduction when relying on estimates for targeting, even the worst performing models achieve decent poverty reduction, but may come at a considerable cost.

The results found here should be caveated by the fact that they are dependent on the Mexican Intercensal data used. Results and model performance may differ considerably in a different country or using different data. It is quite possible that in less-developed economies the geo-referenced covariates perform better than what is observed here, but there is no guarantee this will be the case. Validating the usefulness of a particular type of data is not straight forward as it may not work as well everywhere, it is for this reason that comparisons here are made in a more holistic manner. When comparing the performance of geo-referenced covariates to other type of data one should use the same method, as is done here. On the one hand, we show how using census aggregates at the municipality level relying on gradient boosting comes very close to the best performing method's estimates. On the other hand, using the same method with GIS covariates we show how it falls short of competing with a model where only census aggregates are used.

The paper opens considerable avenues for future research. One avenue is related to methodology. Specifically, how the noise of machine learning methods can be accurately estimated since the noise measures presented here rely on having the true poverty rates which is not possible in the vast majority of contexts. Second, the results suggest that most methods would do a solid job for spatial targeting type exercises. However, a key question is whether or not this will hold elsewhere and under what circumstances will one method outperform others.

References

Aiken, E., Bellue, S., Karlan, D., Udry, C., and Blumenstock, J. E. (2022). Machine learning and phone data can improve targeting of humanitarian aid. *Nature*, 603:864–870.

Athey, S. (2018). The impact of machine learning on economics. In Agrawal, A., Gans, J., and Goldfarb, A., editors, *The economics of artificial intelligence: An agenda*, pages 507–547. University of Chicago Press, Chicago, IL.

Athey, S. and Imbens, G. W. (2019). Machine learning methods that economists should know about. *Annual Review of Economics*, 11:685–725.

Battese, G. E., Harter, R. M., and Fuller, W. A. (1988). An error-components model for prediction of county crop areas using survey and satellite data. *Journal of the American Statistical Association*, 83(401):28–36.

Bedi, T., Coudouel, A., and Simler, K. (2007). *More than a pretty picture: using poverty maps to design better policies and interventions*. World Bank Publications.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, volume 24.

Blumenstock, J., Cadamuro, G., and On, R. (2015). Predicting poverty and wealth from mobile phone metadata. *Science*, 350(6264):1073–1076.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Chi, G., Fang, H., Chatterjee, S., and Blumenstock, J. E. (2022). Microestimates of wealth for all low-and middle-income countries. *Proceedings of the National Academy of Sciences*, 119(3):1–11.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298.

Corral, P., Himelein, K., McGee, K., and Molina, I. (2021a). A map of the poor or a poor map? *Mathematics*, 9(21):2780.

Corral, P., Molina, I., Cojocaru, A., and Segovia, S. (2022). *Guidelines to small area estimation for poverty mapping*. The World Bank, Washington, DC.

Corral, P., Molina, I., and Nguyen, M. (2021b). Pull your small area estimates up by the bootstraps. *Journal of Statistical Computation and Simulation*, 91(16):3304–3357.

Elbers, C., Fujii, T., Lanjouw, P., Özler, B., and Yin, W. (2007). Poverty alleviation through geographic targeting: How much does disaggregation help? *Journal of Development Economics*, 83(1):198–213.

Elbers, C., Lanjouw, J. O., and Lanjouw, P. (2003). Micro-level estimation of poverty and inequality. *Econometrica*, 71(1):355–364.

Fay, R. E. and Herriot, R. A. (1979). Estimates of income for small places: An application of James-Stein procedures to census data. *Journal of the American Statistical Association*, 74(366a):269–277.

Ferre, C. (2018). Sénégal - résultats de l'enquête de mise à jour du rnu: Registre national unique. Washington, D.C. : World Bank Group.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232.

González-Manteiga, W., Lombardía, M. J., Molina, I., Morales, D., and Santamaría, L. (2008). Bootstrap mean squared error of a small-area eblup. *Journal of Statistical Computation and Simulation*, 78(5):443–462.

Grosh, M. E. and Muñoz, J. (1996). *A manual for planning and implementing the Living Standards Measurement Study Survey.* The World Bank, Washington, DC.

Gujarati, D. N. (2003). *Basic Econometrics.* McGraw-Hill, New York, NY.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction.* Springer, New York, NY.

Henderson, C. R. (1953). Estimation of variance and covariance components. *Biometrics*, 9(2):226–252.

Hentschel, J., Lanjouw, J. O., Lanjouw, P., and Poggi, J. (1998). Combining census and survey data to study spatial dimensions of poverty. World Bank Policy Research Working Paper Series No. 1928.

Hersh, J., Engstrom, R., and Mann, M. (2021). Open data for algorithms: Mapping poverty in Belize using open satellite derived features and machine learning. *Information Technology for Development*, 27(2):263–292.

Jean, N., Burke, M., Xie, M., Davis, W. M., Lobell, D. B., and Ermon, S. (2016). Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794.

Lange, S., Pape, U. J., and Pütz, P. (2018). Small area estimation of poverty under structural change. World Bank Policy Research Working Paper No. 8472.

Lee, K. and Braithwaite, J. (2020). High-resolution poverty maps in sub-Saharan Africa. arXiv preprint arXiv:2009.00544.

Majeske, K. D., Lynch-Caris, T., and Brelin-Fornari, J. (2010). Quantifying R^2 bias in the presence of measurement error. *Journal of Applied Statistics*, 37(4):667–677.

Masaki, T., Newhouse, D., Silwal, A. R., Bedada, A., and Engstrom, R. (2020). Small area estimation of non-monetary poverty with geospatial data. World Bank Policy Research Working Paper No. 9383.

Molina, I. (2019). Desagregación de datos en encuestas de hogares: metodologías de estimación en áreas pequeñas.

Molina, I., Corral, P., and Nguyen, M. (2022). Estimation of poverty and inequality in small areas: Review and discussion. *TEST*, pages 1–24.

Molina, I. and Morales, D. (2009). Small area estimation of poverty indicators. *Estadistica e Investigacion Operativa*, 25(3).

Molina, I. and Rao, J. (2010). Small area estimation of poverty indicators. *Canadian Journal of Statistics*, 38(3):369–385.

Natekin, A. and Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, 7(21):1–21.

Nguyen, V. C. (2012). A method to update poverty maps. *The Journal of Development Studies*, 48(12):1844–1863.

Pfeffermann, D. (2013). New important developments in small area estimation. *Statistical Science*, 28(1):40–68.

Pokhriyal, N. and Jacques, D. C. (2017). Combining disparate data sources for improved poverty prediction and mapping. *Proceedings of the National Academy of Sciences*, 114(46):E9783–E9792.

Rao, J. and Molina, I. (2015). *Small area estimation.* John Wiley & Sons, Hoboken, NJ, 2nd edition.

Smythe, I. S. and Blumenstock, J. E. (2022). Geographic microtargeting of social assistance with high-resolution poverty maps. *Proceedings of the National Academy of Sciences*, 119(32):e2120025119.

Steele, J. E., Sundsøy, P. R., Pezzulo, C., Alegana, V. A., Bird, T. J., Blumenstock, J., Bjelland, J., Engø-Monsen, K., de Montjoye, Y.-A., Iqbal, A. M., Hadiuzzaman, K. N., Lu, X., Wetter, E., Tatem, A. J., and Bengtsson, L. (2017). Mapping poverty using mobile phone and satellite data. *Journal of The Royal Society Interface*, 14(127):20160690.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Torabi, M. and Rao, J. (2014). On small area estimation under a sub-area level model. *Journal of Multivariate Analysis*, 127:36–55.

Tzavidis, N., Zhang, L.-C., Luna, A., Schmid, T., and Rojas-Perilla, N. (2018). From start to finish: A framework for the production of small area official statistics. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 181(4):927–979.

van der Weide, R., Blankespoor, B., Elbers, C., and Lanjouw, P. (2022). How accurate is a poverty map based on remote sensing data? World Bank Policy Research Working Paper Series No. 10171.

Varian, H. R. (2014). Big data: New tricks for econometrics. *Journal of Economic Perspectives*, 28(2):3–28.

Yeh, C., Perez, A., Driscoll, A., Azzari, G., Tang, Z., Lobell, D., Ermon, S., and Burke, M. (2020). Using publicly available satellite imagery and deep learning to understand economic well-being in Africa. *Nature Communications*, 11(1):1–11.