Improved poverty targeting through machine learning: An application to the USAID Poverty Assessment Tools

Linden McBride[a,*] and Austin Nichols[b]

[a]*Cornell University, Ithaca, NY*

[b]*DeBruce Foundation, Washington DC*

21 January 2015

Abstract

Proxy means test (PMT) poverty targeting tools have become common tools for beneficiary targeting and poverty assessment where full means tests are costly. Currently popular estimation procedures for generating these tools prioritize minimization of in-sample prediction errors; however, the objective in generating such tools is out-of-sample prediction. In this paper, we present evidence that application of machine learning algorithms to PMT development can substantially improve the out-of-sample performance of these targeting tools. In particular, we show that stochastic ensemble methods can improve out-of-sample performance by 2 to 18 percent over current methods. While we take the USAID poverty assessment tool and base data for demonstration of these methods, the methods applied in this paper should be considered for PMT and other poverty targeting tool development more broadly.

Keywords: poverty targeting; machine learning; proxy means tests; out-of-sample prediction

JEL: C140, I320, O220, O150

[*] *Corresponding author: lem247@cornell.edu; 410 Warren Hall, Dyson School of Applied Economics and Management, Cornell University, Ithaca, NY 14850. Phone: 240-723-0190*

Introduction

Accurate targeting is one of the most important components of an effective and efficient food security or social safety net intervention (Barrett & Lentz 2013; Coady *et al.* 2004). To achieve accurate targeting, project implementers seek to minimize rates of leakage (benefits reaching those who don't need them) and undercoverage (benefits not reaching those who do need them). Full means tests for identification of project beneficiaries can include detailed expenditure and/or consumption surveys; while effective, such tests are also time consuming and expensive. A short-cut to full means tests, proxy means tests (PMTs), were first developed for the targeting of social programs in Latin American countries during the 1980s. PMTs have become common tools for targeting and poverty assessment where full means tests are costly (Coady *et al.* 2004). Today they are used by USAID microenterprise project implementing partners, the World Food Program, and the World Bank, among many others, for the purpose of poverty assessment, beneficiary targeting, and program monitoring and evaluation in developing countries (PAT 2014, WBG 2011).

PMT tools are typically developed by assignment of weights, or parameters, to a number of easily verifiable household characteristics via either regression or principal components analysis (PCA) in an available, nationally representative data set. In the regression approach, household level income/expenditures or poverty status are regressed on household characteristics with the objective of selecting and parameterizing a subset of those characteristics to explain a significant proportion of the variation in expenditures/income or poverty status. In the PCA approach, the parameters are generated by extracting from a set of variables an orthogonal linear combination of a subset of those variables that captures most of the common variation (Filmer and Pritchett 2001, Hastie *et al*. 2009). While each approach has its advocates, those interested solely in targeting tend to rely on regression approaches while PCA has become popular among those interested in generating asset indices that may or may not be used for targeting. Note that the problem of developing tools for poverty targeting can be a fundamentally different problem from that of generating asset indices;[2] this paper speaks only to the problem of developing targeting tools.

[2] For example, we might be concerned about endogeneity but not concerned about out-of-sample performance when generating an asset index to estimate the relationship between school enrollment and wealth, as in Filmer and

Once a PMT tool has been developed (i.e., once weights have been generated for a set of household characteristics that can account for a substantial amount of the variation in the dependent variable) from a sample from a particular population, the development practitioner can apply the tool to the sub-population selected for intervention to rank or classify households according to PMT score. This process involves implementation of a brief household survey to the targeted subpopulation so as to assign values for each of the household characteristics identified during tool development. The observed household characteristics, x_{ij}, are then multiplied by the PMT tool weights, θ_j, for each characteristic j to generate a PMT score for household i, as shown in EQ1.

$$PMTscore_i = \Sigma_j x_{ij}\theta_j \qquad \text{[EQ1]}$$

In many applications, the calculated PMT scores are used to rank households from poorest to wealthiest[3] and the poorest households are selected as program beneficiaries. In the case of the USAID poverty assessment tools that will be described below, the use is more conservative: the PMT scores are used to quantify the number of households above and below an identified poverty threshold so as to ensure proper allocation of USAID funds (PAT 2014). The methodological improvements we propose in this paper apply to both types of uses for PMT tools.

Overall, the objective of a PMT tool is to quickly and accurately identify households meeting particular criteria in a new setting (but under the same data generating process) using a model parameterized with previously available data. Therefore, for PMT tools to serve their purpose, it is important that they perform well not only within the data set or sample in which they were parameterized but also, especially, within the new data set or sample. In other words, high out-of-sample prediction accuracy must be prioritized in the development of PMT tools. In the fields

---

Pritchett (2001). We have no such endogeneity concern when generating targeting tools because we are not attempting causal inference; however, out-of-sample performance is a primary concern.

[3] There are several long-standing debates as to whether PCA type asset indices and/or the use of consumption or income data in the regression approach capture long run economic status, permanent income, current consumption levels, current welfare, non-food spending, or something else altogether. Lee (2014) points out that much of the theoretical support for these various claims is dubious and offers a theoretically grounded approach to the development of asset indices to measure poverty. As much as possible, we remain agnostic on the particular type of well-being that PMT tools capture while noting that the methods we discuss and the way in which we discuss them (e.g., their interpretation as capturing household poverty status) are standard in the literature and in practice.

of machine learning and predictive analytics, stochastic ensemble methods have been shown to perform very well out-of-sample due to the bias and variance reducing features of such methods.

In this paper, we present evidence that the application of machine learning methods to PMT development can substantially improve the out-of-sample performance of these targeting tools. We illustrate the potential of machine learning algorithms for the improvement of PMT tool development by applying stochastic ensemble algorithms such as random forests to a set of PMT tools that have been developed by the University of Maryland IRIS Center for the purpose of USAID poverty assessment. While we take the USAID poverty assessment tool and base data for demonstration of these methods, the methods applied in this paper should be considered for PMT and other poverty targeting tool development more broadly.

We next present the USAID poverty assessment tool development and accuracy evaluation criteria; we then introduce the stochastic ensemble algorithms, regression forests and quantile regression forests, that we apply to the problem of developing more accurate out-of-sample targeting tools; an explanation of our data and methods follows. We close with results and conclusions.

The USAID Poverty Assessment Tool

The development of the USAID poverty assessment tool (PAT) dates from 2000, when the U.S. Congress passed the Microenterprise for Self-Reliance and International Anti-Corruption Act, mandating that half of all USAID microenterprise funds benefit the very poor (PAT 2014). In the context of this legislation, the very poor are defined as those households living on less than the equivalent of a dollar per day or those households considered "among the poorest 50 percent of households below the country's own national poverty line" (IRIS Center 2005). Subsequent legislation required USAID to develop and certify low-cost tools to enable its microenterprise project-implementing partners[4] to assess the poverty status of microenterprise beneficiaries.

[4] The implementing partners who are required to make use of the PAT include "all projects and partner organizations receiving at least US$100,000 from USAID in a fiscal year for microenterprise activities in countries with a USAID-approved tool" (PAT 2014). In 2013 this entailed 71 partners receiving a total of 110 million dollars (USAID MMR).

USAID engaged the IRIS Center at the University of Maryland in 2003 to create the tools. To date, the IRIS Center has developed, and USAID has certified, tools for 38 countries.[5]

Using existing Living Standards Measurement Study (LSMS) data as well as survey data collected by IRIS, the IRIS Center developed country-specific PAT tools following the general PMT development procedure: they first identified a subset of household characteristics (approx. 15) from the larger dataset of 70-125 available observables that accounted for the greatest variation in household level income via an R-squared maximization routine, SAS MAXR;[6] they then selected for the final tool the parameters identified by the statistical model—whether OLS, quantile regression, logit, or probit—that produced the highest prediction accuracy.

The predictive ability of the resulting coefficients was evaluated against a number of accuracy criteria—total accuracy, poverty accuracy, undercoverage, leakage, and the balanced poverty accuracy criterion—each of which is defined below. These criteria allow for ex-ante evaluation of the generated poverty assessment tools via systematic consideration of each possible outcome/error type as presented in the confusion matrix in Table 1: True Positive (the true very poor, P=1, are identified by the tool as very poor, $\hat{P} = 1$); False Negative (the true very poor, P=1, are identified by the tool as non very poor, $\hat{P} = 0$); False Negative (the true non very poor, $P = 0$, are identified by the tool as very poor, $\hat{P} = 1$); True Negative (and the true non very poor, $P = 0$, are identified by the tool as non very poor, $\hat{P} = 0$). As defined, the elements of Table 1 are mutually exclusive and sum to one.

Table 1. Poverty prediction outcomes

	$P = 1$	$P = 0$
$\hat{P} = 1$	True positive (TP)	False positive (FP)
$\hat{P} = 0$	False negative (FN)	True negative (TN)

[5] Albania, Azerbaijan, Bangladesh, Bolivia, Bosnia and Herzegovina, Cambodia, Colombia, East Timor, Ecuador, El Salvador, Ethiopia, Ghana, Guatemala, Haiti, India, Indonesia, Jamaica, Kazakhstan, Kenya, Kosovo, Liberia, Madagascar, Malawi, Mexico, Nepal, Nicaragua, Nigeria, Paraguay, Peru, The Philippines, Rwanda, Senegal, Serbia, Tanzania, Tajikistan, Uganda, Vietnam, and the West Bank.

[6] The MAXR procedure operates by selecting and rejecting variables one by one with the objective of maximizing the improvement in a model's R^2 (SAS 2009).

The classification literature has developed many metrics based on this confusion matrix for the assessment of classification accuracy. Following the IRIS Center, and relying on the categories given in Table 1, the accuracy criteria used to assess PAT performance are defined as follows: total accuracy (TA) is the sum of the correctly predicted very poor and the correctly predicted non very poor as a percentage of the total sample, (TA=(TP+TN)/(TP+TN+FP+FN)). Poverty accuracy (PA) is the correctly predicted very poor as a percentage of the total true very poor, (PA=TP/(TP+FP)). The undercoverage rate is the ratio of true very poor incorrectly predicted as non very poor to total true very poor, (UC=FN/(TP+FN)) while the leakage rate is the ratio of true non very poor incorrectly identified as very poor to total true very poor, (LE=FP/(TP+FN)). Finally, the balanced poverty accuracy criterion (BPAC) is the correctly predicted very poor as a percentage of the true very poor minus the absolute difference between the undercoverage and leakage rates, (BPAC=TP/(TP+FP)-|FN/(TP+FP)-FP/(TP+FP)|). These accuracy criteria are summarized in Table 2.

Total accuracy, or one minus the mean squared error, is very familiar to economists as a metric for model assessment. However, there are several reasons why total accuracy might not be an adequate metric for assessing the accuracy of a poverty tool. Consider an example wherein a population of 100 includes 10 poor households. A tool that simply classifies the entire population as non-poor would have a total accuracy rate of 90 percent, which seems quite good. However, this tool would have failed to identify a single poor household. Therefore, metrics beyond total accuracy are necessary for assessment of poverty tool performance; these additional metrics include poverty accuracy (also known as *precision* in the classification and predictive analytics literature) and undercoverage (*false negative*) and leakage (*false positive*) rates. In the example just given, the poverty accuracy of the tool would be zero percent, and the undercoverage rate would be 100 percent. These additional metrics offer a better picture of the tool's performance than does total accuracy alone. The BPAC combines these three metrics—poverty accuracy, undercoverage, and leakage—by penalizing the poverty accuracy rate with the extent to which the leakage and undercoverage rates exceed one another. The BPAC is an innovation of the IRIS Center; it was created to balance "the stipulations of the Congressional Mandate against the practical implications of the assessment tools" (IRIS 2005). The other criteria are standard in

PMT development. However, it should be noted that IRIS computes leakage in an unconventional manner.[7]

Table 2. Targeting accuracy metrics

Total accuracy	TA=(TP+TN)/(TP+TN+FP+FN)=1-MSE
Poverty accuracy	PA=TP/(TP+FP)
Leakage	LE=FP/(TP+FN)
Undercoverage	UC=FN/(TP+FN)
Balanced poverty accuracy criterion	BPAC=TP/(TP+FP)-|FN/(TP+FP)-FP/(TP+FP)|

PAT model selection for each country was ultimately made by IRIS based on the BPAC criteria. While we follow the prioritization of these criteria in the analysis that follows, the methods we propose can be used to meet other prioritized accuracy criteria as well.

Stochastic Ensemble Methods: Regression and Quantile Regression Forests

Classification and regression trees are a class of supervised learning methods that produce predictive models via stratification of a feature (in the case of poverty tool development, a feature is a variable or characteristic) space into a number of regions following a decision rule (Hastie *et al*. 2009). A canonical and intuitive example of a classification tree is that of predicting, based on a number of features such as age, gender, and class, who survived the sinking of the Titanic.[8] While both classification and regression trees can be used to make predictions regarding the poverty status of households based on observable household characteristics, this paper focuses on regression and, in particular, quantile regression trees and

[7] Whereas leakage rates are commonly computed as FP/(TP+FP), IRIS computes leakage rates as FP/(TP+FN). This adjustment to the denominator in the calculation of leakage rates has two consequences: it can lead to calculated leakage rates that are greater than one, producing a heavy penalty in the calculation of BPAC where such leakage occurs (it is not clear that IRIS intended for this outcome); it keeps constant the denominator across poverty accuracy, undercoverage, and leakage rates, allowing IRIS to easily perform the addition and subtraction necessary for the BPAC calculation (we assume this was IRIS's purpose in modifying the denominator).

[8] See Varian (2014) for an example. Many examples and data are also available at The Comprehensive R Archive Network at http://cran.r-project.org.

forests, due to the advantages the latter offer in terms of making predictions about households concentrated at the lower end of the income distribution.

Regression trees operate via a recursive binary splitting algorithm as follows (Hastie *et al*. 2009): for *N* observations of response variable, y_i, and a vector of characteristics, $\boldsymbol{x}_{ij}$, where $i = 1,2, \ldots N$ is the number of observations and j=1,2,…J is the number of features, consider the splitting variable, x_j, and the split point, where $x_{ij} = s$, that define the half planes, R_1 and R_2 as indicated in EQ2,

$$R_1(j,s) = \{x_{ij} | x_{ij} \leq s\} \text{ and } R_2(j,s) = \{x_{ij} | x_{ij} > s\} \qquad \text{[EQ2]}$$

The algorithm selects x_j and s to solve the minimization problem,

$$\min_{js}[\min_{c_1} \Sigma_{x_i \in R_1(j,s)}(y_i - c_1)^2 + \min_{c_2} \Sigma_{x_i \in R_2(j,s)}(y_i - c_2)^2] \qquad \text{[EQ3]}$$

Where the inner minimizations are solved by

$$c_1 = \frac{1}{n}\Sigma_i(y_i | x_i \in R_1(j,s)) \text{ and } c_2 = \frac{1}{n}\Sigma_i(y_i | x_i \in R_2(j,s)) \qquad \text{[EQ4]}$$

In words, the regression tree algorithm chooses the variable, x_j (the splitting variable), and the value of that variable, s (the split point), that minimizes the summed squared distance between the mean response variable and the actual response variables for the observations found in each of the resulting regions. The algorithm is effectively weighting the response variables by the predictive value of the observations within each region.

Once the optimal split in EQ3 is identified, the algorithm proceeds within the new partitions. The recursive binary splitting process can continue until a stopping criterion is reached; however, larger trees may overfit the data. In the case that we want to bootstrap over this algorithm—a good idea, as the algorithm may make different splitting decisions in different subsets of the data—it becomes apparent that a bias for variance trade-off is made as we allow the trees to grow large.[9] A collection of larger trees will have high variance but low bias while a collection of smaller trees will have low variance but high bias.

[9] A variety of options for "pruning" trees exist to address these issues in a regression tree framework (Hastie *et al.* 2009). We don't discuss these here but move on instead to random forests, which address the problem without pruning.

Fortunately, in this setting, the bias-variance trade off can be somewhat overcome via a process called bootstrap aggregation, or bagging. Bagging involves bootstrapping with replacement a number of approximately unbiased and identically distributed regression trees and then averaging across them so as to reduce the variance of the predictor. However, bagging cannot address the persistent variance that arises due to the fact that the trees themselves are correlated, as they were generated over the same feature space. Consider, for example, a set of B identically distributed but correlated regression trees, each with variance σ^2. If ρ represents the pairwise correlation between the trees, then the variance of the average of these trees is $\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$. As B grows large, the term $\frac{1-\rho}{B}\sigma^2$ will approach zero, reducing the overall variance. However, the first term, $\rho\sigma^2$, persists (Hastie *et al.* 2009).

Reducing this persistent variance component of the bagged predictor is the innovation of random forests. Introduced by Breiman (2001a), regression forests improve the variance reduction feature of bagged regression trees by de-correlating the trees via random selection of the features (variables) over which the algorithm may split. Random selection of the features over which any split can take place reduces ρ for an overall reduction in variance. The number of random features available to the algorithm at any split is typically limited to 1/3 of the total number of features (Hastie *et al.* 2009).

Finally, in a random forest algorithm, the mean squared error of the prediction is estimated in the "out of bag" sample (OOB), the (on average) third of the training data set on which any given tree has not been built, in a manner similar to leave-one-out cross validation (Breiman 2001a). The OOB sample can be used to assess model performance as well as variable importance.

The random forest training algorithm produces a collection of B trees, denoted $\{T(x;\Theta_b)\}_1^B$, where Θ_b indicates the bth tree. The regression forest predictor is then the bagged prediction,

$$\hat{f}(x_i) = \frac{1}{B}\sum_{b=1}^{B} T(x_i;\Theta_b) \qquad \text{[EQ5]}$$

The regression forest algorithm is detailed in the appendix.

It has been shown that regression forests offer consistent and approximately unbiased estimates of the conditional mean of a response variable (Breiman 2004, Hastie *et al.* 2009). However, as elaborated by Koenker (2005), among others, the conditional mean tells only part of the story of

the conditional distribution of y given X. Therefore, we also apply quantile regression forests, as developed by Meinshausen (2006), to our PMT tool development.

Meinshausen (2006) draws on insights from Lin and Jeon (2006) who show that random forest predictors can be thought of as weighted means of the response variable, y_i, as shown in EQ6.

$$\hat{f}(x_i) = \frac{1}{B}\sum_{b=1}^{B} T(x_i; \Theta_b) = \sum_{i=1}^{N} \frac{\sum_{b=1}^{B} w_i(x_i; \Theta_b)}{\beta} y_i \qquad \text{[EQ6]}$$

In EQ6, $w_i(x_i; \Theta)$ represents the weight vector obtained by averaging over the observed values in a given region R_l, $(l = 1 \ldots L)$. Application of the weight vector to the response variable is simply another way of considering the conditional averaging of the response variable, as represented in EQ4 above and shown in EQ7.

$$w_i(x_i; \Theta) y_i = \frac{1}{n} \Sigma_i (y_i | x_i \in R_l(j, s)) \qquad \text{[EQ7]}$$

With this insight, Meinshausen (2006) produces quantile regression forests, as a generalization of regression forests in which not only the conditional mean, but the entire conditional distribution of the response variable is estimated (EQ8).

$$\hat{f}_y(x_i) = \sum_{i=1}^{N} \frac{\sum_{b=1}^{B} w_i(x_i; \Theta_b)}{\beta} 1\{y_i \leq y\} \qquad \text{[EQ8]}$$

Meinshausen (2006) provides a proof for the consistency of this method and demonstrates the gains in predictive performance of quantile regression forests over linear quantile regression. A quantile approach is particularly useful for the purposes of PMT tool development due to the fact that the very poor are often concentrated at one end of the conditional income distribution, far from the conditional mean. The quantile regression forest algorithm is detailed in the appendix.

Using regression forest and quantile regression forest algorithms, we expect to realize improvements in out-of-sample targeting accuracy. We note, however, that this methodology requires the critical assumption that the data generating process remains unchanged between tool development and tool application. That is, the algorithm can perform well out of sample but not out of population. This limitation plagues any sample based estimation routine.

Empirical Method and Data

We produce a set of country-specific examples from the same survey data used by the IRIS Center to construct their PATs. We replicate the PAT development process by extracting the

same variables that IRIS has extracted from the same data sets and then generating identical estimation models. We are limited in our replication process to the use of LSMS data sets that are publicly available. We have additionally constrained ourselves to the LSMS data sets for which income or expenditure aggregates are also publicly available due to the challenges of precisely replicating an income or expenditure aggregate that IRIS may have generated.

From the publicly available data sets meeting these criteria, we selected three nearly arbitrarily: the 2005 Bolivia Encuesta de Hogares (EH), the 2001 Timor Leste Living Standards Survey (TLSS), and the 2004-2005 Malawi Second Integrated Household Survey (IHS2). These data sets present a reasonable representation of the settings in which PATs have been developed. Each data set differs in number of observations, poverty level, and IRIS selected household characteristics. The data are summarized in Table 3 where we can see that the number of household level observations range from 1,800 in East Timor to 11,280 in Malawi. Likewise, the USAID defined poverty rates range considerably, from 24.2 percent in Bolivia to 90.4 percent in Malawi.

The fourth column of Table 3 displays the household level characteristics selected by IRIS for PAT tool development; many characteristics such as household size, age of household head, household construction materials, and material possessions are common across data sets.

Table 3. LSMS surveys used in PAT development and replicated by authors

County	Data	Obs.	IRIS selected variables	Poverty rate
Bolivia	2005 Encuesta de Hogares (EH)	4,086	hhsize, hhsize2, age head, age head2, regions, rural, sublease, brick wall, wood wall, dirt floor, cement floor, fridge, radio, tv, dvd, fan, car, number beds, number kitchens, number computers, sheep	24.20%
Malawi	2004-2005 Second Integrated Household Survey (IHS2)	11,280	hhsize, hhsize2, age head, age head2, regions, rural, never married, share of adults with out education, share of adults who can read, number of rooms, cement floor, electricity, flush toilet, soap, bed, bike, music player, coffee table, iron, garden, goats	90.40%
East Timor	2001 Timor Leste Living Standards Survey (TLSS)	1,800	hhsize, hhsize2, age head, age head2, regions, rattantin wall, leaf roof, concreter or tile roof, number rooms, private water, shared water, toilet is a bowl or bucket, electricity light, private light, fan, number of adults who read, farmland, number of axes number of baskets, number of chickens	44.70%

We provide the IRIS reported in-sample-accuracy estimates for each country level dataset in each row 1 of appendix Table A1. These are the estimates on which the IRIS model selection was made. We replicate these models and report the replication estimates in each row 5 of Table A1. Within-country comparisons of our replication estimates (Table A1, row 5) with the estimates reported by IRIS (Table A1, row 1) serve as a check on how well we have replicated the PAT tool development process. In the case of Bolivia, our replication estimates do not perform as well as those of IRIS; however, it should be noted that IRIS built the Bolivia PAT

tools on a randomly selected subset of the data. We cannot replicate precisely the same random draw and so report the full sample estimates. In the case of East Timor and Malawi, our replication estimates are very close to those reported by IRIS.

Our empirical approach is to randomly draw, with replacement, two samples of size N/2 from each country level data set, producing a training sample and a testing sample. The random forest models are built in the training sample where, for any given $(\boldsymbol{x}_i, y_i)$, an average of two thirds of the training data are used to build bagged regression trees and the remaining third is reserved for out of bag, and therefore unbiased, running estimates of the prediction error over a forest of 500[10] trees. The resulting model is then taken to the testing sample to assess classification accuracy. Note that, in principle, the division into training and validation sets is unnecessary using a stochastic ensemble method, since unbiased out-of-sample statistics are always produced in the training data. By only using half the data, we are stacking the deck against the stochastic ensemble method. Following the methodology for out-of-sample testing used by the IRIS center, we test the model using 1000 bootstrapped samples of the testing sample.[11] We run both regression forest and quantile regression forest algorithms in R using packages developed by Liaw and Wiener (2002) and the R Development Core Team (2005). We select and report values for the model that offers the greatest BPAC prediction accuracy.

Results

Results are displayed graphically in Figures 1, 2, and 3 and numerically in appendix Table A1. In both formats we compare the IRIS out-of-sample bootstrap accuracy estimates with the out-of-sample accuracy metrics for the stochastic ensemble generated targeting tools. The confidence

[10] 500 trees is the default setting in the randomForest package in R. From casual observation, the OOB error has largely stabilized by the time the forest has reached 200-300 trees; this is consistent with the literature (Hastie *et al.* 2009).

[11] According to their documentation, the IRIS Center builds a PAT model in one half of the data and then bootstrap samples the remaining half to estimate out-of-sample performance of the tool. While this approach to training and testing may approximate the real world use of the tool, the resulting tool and its reported performance accuracy are necessarily some function of this first random split of the data. A more conservative approach to tool training and testing would be to iterate over, as opposed to within, this first random split, training the model in one half of the data and then testing it in the remaining half in each iteration. So as to produce accuracy estimates that could be compared with those of IRIS, we have followed their procedure. However, we also report the more conservative estimates, which still outperform those of IRIS, in appendix Table A2.

bars in each figure display the non-parametric bootstrap confidence intervals, where the lower bound is the 2.5th percentile and upper bound is the 97.5th percentile bootstrap estimate. Standard errors are reported in Table A1.

While stochastic ensemble methods do not improve on the total accuracy of the IRIS generated tools (Figure 1, first graph), gains in poverty accuracy, ranging from a 2 percent improvement in Malawi to an 8 percent improvement in Bolivia, are observed across all countries (Figure 1, second graph). Parametric and nonparametric tests of equivalence of the bootstrapped means, reported in the first and third columns of Table 4, indicate that these gains are highly statistically significant.

From Figure 2 (first graph), we can see that these gains in poverty accuracy are not without trade-offs: the leakage rates for the stochastic ensemble generated tools are significantly greater than those reported for the IRIS generated tools in both Bolivia and East Timor. However, the stochastic ensemble generated tool performs much better than IRIS's in terms of undercoverage rates; this error is decreased in each country (Figure 2, second graph).

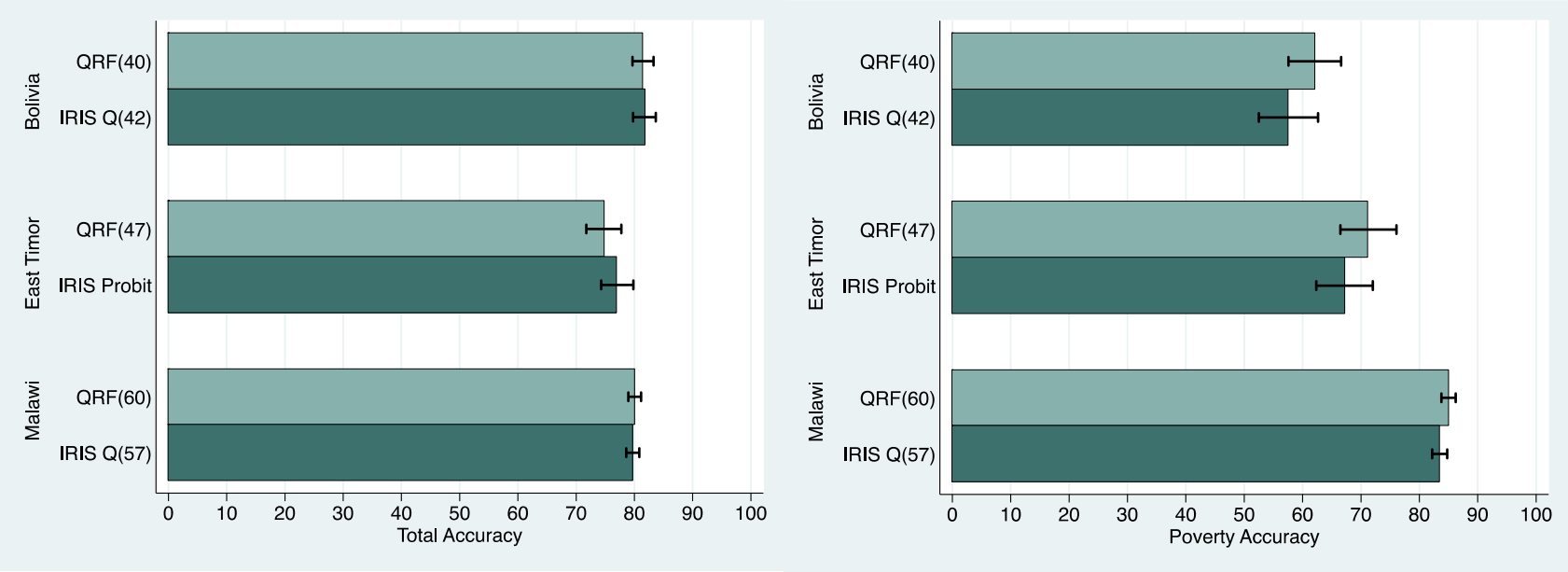
Finally, for ex-ante evaluation of tool performance, we assess how these trade-offs net out in terms of USAID's key accuracy metric, the BPAC. Figure 3 demonstrates that the accuracy of the stochastic ensemble generated tool out-performs that of the IRIS generated tool in each country. Improvements range from 2 percent in Malawi to 18 percent in Bolivia. Parametric and nonparametric tests again support the statistical significance of these gains (Table 4, columns 2 and 4).

Table 4. Tests of equality of bootstrap Poverty Accuracy and BPAC means across estimates

	t-test		Mann-Whitney *U* test	
Country	$H_0: \mu^*_{PA_{AE}} = \mu^*_{PA_{IRIS}}$	$H_0: \mu^*_{BPAC_{AE}} = \mu^*_{BPAC_{IRIS}}$	$H_0: \mu^*_{PA_{AE}} = \mu^*_{PA_{IRIS}}$	$H_0: \mu^*_{BPAC_{AE}} = \mu^*_{BPAC_{IRIS}}$
Bolivia	*t*=41.0584 (0.000)	*t*=37.8474 (0.000)	*z*=25.194 (0.000)	*z*=22.150 (0.000)
East Timor	*t*=35.1057 (0.000)	*t*=41.7830 (0.000)	*z*=27.046 (0.000)	*z*=29.279 (0.000)
Malawi	*t*=55.5455 (0.000)	*t*=47.1786 (0.000)	*z*=16.578 (0.000)	*z*=10.364 (0.000)

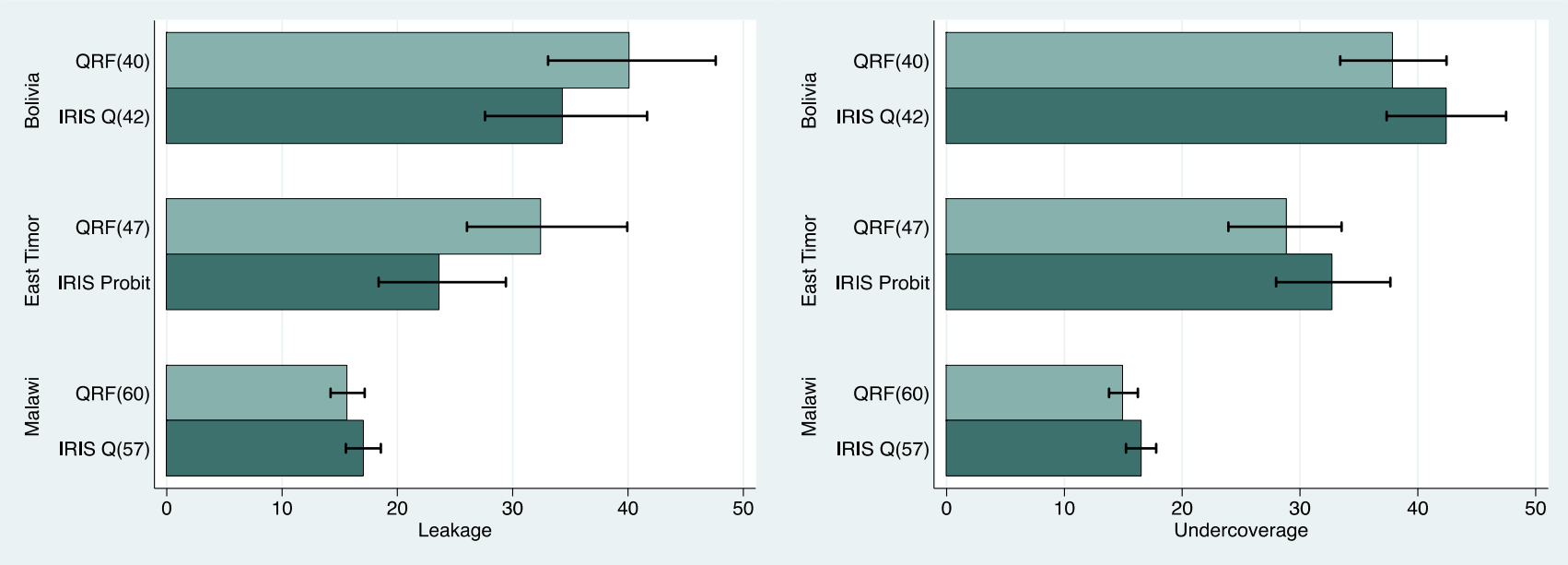
Note: *indicates bootstrap mean. AE indicates authors' estimates. IRIS indicates authors' replication of IRIS's estimates. PA indicates Poverty Accuracy; BPAC indicates Balanced Poverty Accuracy Criteria. p-values in parentheses.

Figure 1. Total and Poverty Accuracy by country and estimation procedure



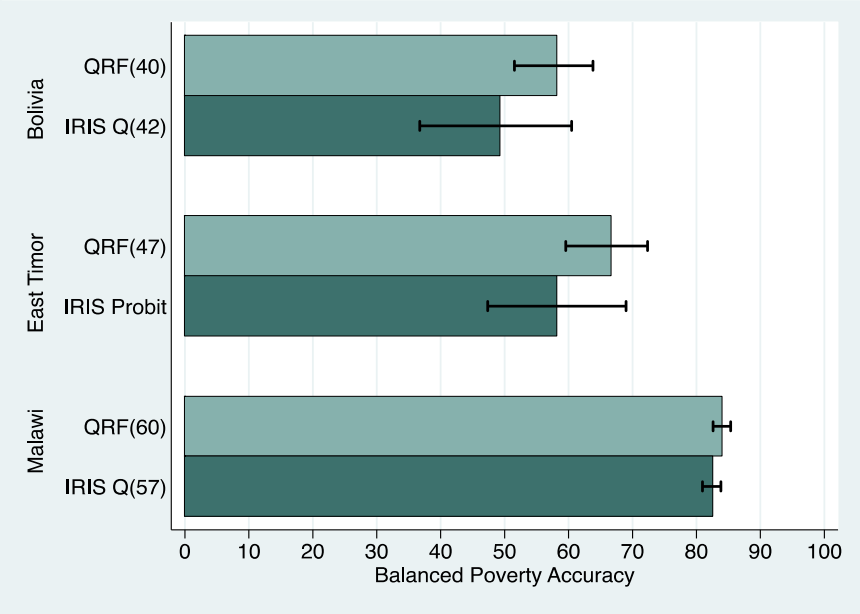
Note: 'QRF(#)' indicates quantile regression forest estimated by the authors at the #th quantile. 'IRIS Q(#)' indicates quantile regression estimated by IRIS at the #th quantile. 'IRIS probit' indicates probit regression estimated by IRIS. Error bars reflect the non-parametric confidence intervals.

Figure 2. Leakage and Undercoverage rates by country and estimation procedure



Note: 'QRF(#)' indicates quantile regression forest estimated by the authors at the #th quantile. 'IRIS Q(#)' indicates quantile regression estimated by IRIS at the #th quantile. 'IRIS probit' indicates probit regression estimated by IRIS. Error bars reflect the non-parametric confidence intervals.

Figure 3. Balanced Poverty Accuracy Criteria by country and estimation procedure



Note: 'QRF(#)' indicates quantile regression forest estimated by the authors at the #th quantile. 'IRIS Q(#)' indicates quantile regression estimated by IRIS at the #th quantile. 'IRIS probit' indicates probit regression estimated by IRIS. Error bars reflect the non-parametric confidence intervals.

Conclusion

We have proposed methods for the improvement of a particular type of poverty targeting tool: proxy means test targeting. In the country level case studies analyzed here, application of stochastic ensemble methods to the problem of developing a poverty targeting tool produces a significant gain in poverty accuracy, a significant reduction in undercoverage, and an overall improvement in BPAC in comparison to current methods. Our analysis takes as given the PAT selected variables so as to demonstrate the power of machine learning methods in this setting; however, beginning with a larger set of variables over which the algorithm may build the model may produce even greater gains in targeting accuracy.[12] Therefore, the gains in accuracy we report are likely conservative; further analysis that involves augmenting the set of possible variables is planned.

While we do not advocate uncritical use of random forest algorithms or other stochastic ensemble methods for the improvement of poverty targeting accuracy, we do suggest further exploration of machine learning methods—particularly those that make use of cross validation to minimize prediction error—for tool development.

[12] Note, however, that the algorithm cannot be given completely free range in variable selection as the selected variables must be easily observable household characteristics (that can be quickly verified with a visit to the household) for them to contribute meaningfully to a PMT test.

References

Alatas, V., A. Banerjee, R. Hanna, B. Olken, and J. Tobias. 2012. Targeting the Poor: Evidence from a Field Experiment in Indonesia. *American Economic Review*, 102(4): 1206-1240.

Alatas, V., A. Banerjee, R. Hanna, B. Olken, R. Punamasar, and M. Wai-Poi. 2013. Self-Targeting: Evidence from a Field Experiment in Indonesia. Working paper. Accessed May 2014 at http://economics.mit.edu/files/8449

Alderman, H. 2002. Do local officials know something we don't? Decentralization of targeted transfers in Albania. *Journal of Public Economics*, 83(2002): 375-404.

Barrett, C.B. and E. Lentz. 2013. Hunger and Food Insecurity. *The Oxford Handbook of Poverty and Society.* Eds. D. Brady and L.M. Burton. Oxford: OUP.

Breiman, L. 2001a. Random Forests. *Machine Learning*, 45: 5-32.

----. 2001b. Statistical Modeling: The Two Cultures. *Statistical Science*, 16(3): 199-231.

----. 2004. Consistency for a simple model of Random Forests. Technical Report. UC-Berkeley.

Coady, D., M. Grosh, and J. Hoddinott. 2004. *Targeting of Transfers in Developing Countries: Review of Lessons and Experience*. Washington, DC: The International Bank for Reconstruction and Development.

Filmer, D. and L. H. Pritchett. 2001. Estimating Wealth Effects without Expenditure Data or Tears: An Applicaton to Educational Enrollments in States of India. *Demography*, 38(1): 115-132.

Grosh, M. and J. Baker. 1995. Proxy Means Tests for Targeting Social Programs. *LSMS Working Paper* No. 118. Washington, DC: The World Bank.

Hastie, T., R. J. Tibshirani, and J. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* 2nd ed. New York: Springer.

IRIS Center. 2005. "Note on Assessment and Improvement of Tool Accuracy." Poverty Assessment Tools. USAID. Accessed http://www.povertytools.org/training_documents /Introduction%20to%20PA/Accuracy_Note.pdf

Koenker, R. 2005. *Quantile Regression*. Cambridge: Cambridge University Press.

Liaw, A. and M. Wiener. 2002. Classification and regression by randomForest. *R News*. 2:18-22.

Lee, D. 2014. Measuring Poverty Using Asset Ownership: Developing a Theory-Driven Asset Index Incorporating Utility and Prices. Unpublished Job Market Paper. Accessed January 2014 at http://areweb.berkeley.edu/candidate/Diana_Lee

Lin, Y., and Y. Jeon. 2006. Random Forest and Adaptive Nearest Neighbors. *Journal of the American Statistical Association*, 101(474): 578-590.

Meinshausen, N. 2006. Quantile Regression Forests. *Journal of Machine Learning Research*, 7: 983-999.

PAT (Poverty Assessment Tool). 2014. Quantifying the Very Poor. Poverty Assessment Tools Website. Accessed Feb 2014 at http://www.povertytools.org

R Development Core Team. 2005. *R: A language and environment for statistical computing. R Foundation for Statistical Computing.* Vienna, Austria.

SAS Institute Inc. 2009. SAS/STAT 9.2 User's Guide, Second Edition. Cary, NC: SAS Institute Inc. Accessed 13 May 2012 at support.sas.com/documentation/cdl/en/statug/63033/PDF/default/statug.pdf

USAID MRR. USAID Microenterprise Results Reporting Portal. Data accessed 17 Dec 2014 at eads.usaid.gov/mrr/

Varian, H. 2014. Big Data: New Tricks for Econometrics. *Journal of Economic Perspectives*, 28(2): 3-28.

WBG (World Bank Group). 2011. Targeting: Safety Nets and Transfers: Proxy Means Testing. Accessed May 2014 at web.worldbank.org

Appendix

Random forest algorithm (Hastie *et al.* 2009, Breiman 2001)

1) Grow B trees, $T(\theta_b), b = 1, \ldots, B$ by recursively repeating steps (a)-(c):
    a. Select *m* variables at random from the total J variables, (j=1,…J).
    b. Select variable x_j and split point $x_{ij} = s$ to solve the minimization problem as shown in EQ2-EQ4.
    c. Split data into the resulting regions.
2) Output ensemble of trees $\{T_b\}_1^B$.
3) To make prediction at new point, *x*, drop observation down all trees and calculate $\hat{f}_{rf}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$.

Quantile regression forest algorithm (Meinshausen 2006)

1) Grow B trees, $T(\theta_b), b = 1, \ldots, B$ as in the random forests algorithm. However, retain the value of all observation in a given region, not just their average.
2) For a given x_{ij}, drop observation down all trees and compute the weight, $w_i(x_i; \Theta_b)$ of observation i for every tree, b, as $w_i(x_i; \Theta_b) = \frac{1\{x_i \in R_l(j,s)\}}{\sum_{i=1}^{n} 1\{x_i \in R_l(j,s)\}}$. Then compute the weight for every observation as an average over all trees as $\frac{\sum_{b=1}^{B} w_i(x_i;\Theta_b)}{\beta}$.
3) Compute the estimate of the distribution function as $\sum_{i=1}^{N} \frac{\sum_{b=1}^{B} w_i(x_i;\Theta_b)}{\beta} 1\{y_i \leq y\}$ for all y.

Table A1. A comparison of stochastic ensemble and IRIS accuracy results

Data		Estimation	TA	PA	UC	LE	BPAC
Bolivia (2005 EH)	IRIS	1) Q(0.42)-In sample (half)	83.65	67.18	32.82	33.29	66.71
		2) Q(0.42)^	81.88	57.58	42.42	34.3	49.33
		3) Std. Err.	1.02	2.61	2.61	3.6	6.11
		4) Q(0.42)+	[79.78, 83.68]	[52.51, 62.65]	[37.35, 47.49]	[27.6, 41.66]	[36.73, 60.48]
	Authors' est.	5) Q(0.42) rep.-In sample (full)	82.45	60.69	39.3	33.71	55.1
		6) QRF(0.40)^	81.44	62.14	37.86	40.11	58.17
		7) Std. Err.	0.92	2.35	2.35	3.83	4.15
		8) QRF(0.40)+	[79.67,83.28]	[57.59,66.59]	[22.41,42.41]	[33.10,47.57]	[47.15,63.45]
East Timor (2001 TLSS)	IRIS	1) Probit-In sample (full)	77.14	75.08	24.92	26.20	73.79
		2) Probit^***	76.94	67.29	32.71	23.64	58.17
		3) Std. Err.	1.41	2.47	2.47	2.80	5.54
		4) Probit+,***	[74.28,79.82]	[62.33,72.03]	[27.97,37.67]	[18.37,29.41]	[47.35,69.00]
	Authors' est.	5) Probit rep.-In sample (full)	77.16	71.41	28.59	27.633	70.45
		6) QRF(0.47)^	74.8	71.16	28.84	32.42	66.67
		7) Std. Err.	1.52	2.46	2.46	3.66	3.27
		8) QRF(0.47)+	[71.75,77.75]	[66.47,76.08]	[23.92,33.53]	[26.05,39.94]	[59.59,72.35]
Malawi (2004/5 IHS2)	IRIS	1) Q(0.57)-In sample (half)	80.15	84.12	15.88	16.43	83.57
		2) Q(0.57)^	79.69	83.47	16.53	17.06	82.56
		3) Std. Err.	0.55	0.65	0.65	0.76	0.74
		4) Q(0.57)+	[78.6, 80.84]	[82.2, 84.77]	15.23, 17.79]	[15.53, 18.56]	[80.95, 83.82]
	Authors' est.	5) Q(0.57) rep.-In sample (full)	80.82	84.88	15.11	14.39	84.17
		6) QRF(0.60)^	80.04	85.06	14.94	15.64	84.09
		7) Std. Err.	0.57	0.63	0.63	0.76	0.71
		8) QRF(0.60)+	[78.94,81.13]	[83.77,86.21]	[13.79,16.23]	[14.21,17.16]	[82.59,85.36]

Note: 'QRF(#)' indicates quantile regression forest estimated at the #th quantile; 'Q(#)' indicates quantile regression estimated at the #th quantile.

^ Bootstrapped 1000 times, with replacement, mean reported

+ Bootstrapped 1000 times, with replacement; 95% bootstrap confidence interval reported where lower bound is 2.5% and upper bound is 97.5%

***Because these bootstrapped estimates were not available in materials made public by IRIS, the estimates reported here were calculated by the authors based on the replication sample and model.

Table A2. Stochastic ensemble estimates iterated over training and testing data (i.e., conservative estimate of performance)

Data		Estimation	TA	PA	UC	LE	BPAC
Bolivia (2005 EH)	Authors' est	6) QRF(0.40)^	80.99	60.36	39.64	39.6	54.99
		7) Std. Err.	0.72	2.59	2.59	4.68	4.59
		8) QRF(0.40)+	[79.54,82.38]	[55.51,65.53]	[34.47,44.49]	[31.10,49.35]	[43.88,61.05]
East Timor (2001 TLSS)	Authors' est	6) QRF(0.50)^	74.66	75.05	24.95	37.48	62.16
		7) Std. Err.	1.13	3.23	3.23	5.01	4.62
		8) QRF(0.50)+	[72.45,76.90]	[68.28,80.56]	[19.44,31.72]	[28.04,47.10]	[52.87,69.87]
Malawi (2004/5 IHS2)	Authors' est	6) QRF(0.57)^	80.31	86.64	13.36	16.94	83.03
		7) Std. Err.	0.37	0.88	0.87	0.87	0.84
		8) QRF(0.57)+	[79.58,81.02]	[84.88,88.24]	[11.76,15.12]	[15.25,18.60]	[81.38,84.61]

Note: 'QRF(#)' indicates quantile regression forest estimated at the #th quantile.

^ Bootstrapped 1000 times, with replacement, mean reported

+ Bootstrapped 1000 times, with replacement; 95% bootstrap confidence interval reported where lower bound is 2.5% and upper bound is 97.5%