2019
IARIW-World Bank

Special IARIW-World Bank Conference "New Approaches to Defining and Measuring Poverty in a Growing World" Washington, DC, November 7-8, 2019

Machine Learning for Monitoring Twin Goals: Pitfalls and Possible Solutions

Kazusa Yoshimura
Nobuo Yoshida

Paper Prepared for the IARIW-World Bank Conference

Washington, DC, November 7-8, 2019

Machine learning for monitoring twin goals: pitfalls and possible solutions

[1]Kazusa Yoshimura, [2]Nobuo Yoshida
[1,2] Poverty and equity global practice, the World Bank

Abstract
Traditional poverty measurement is costly and time-consuming. It costs multi-million dollars and requires 2 to 3 years from the preparation to dissemination of final estimates. This paper explores whether Machine Learning (ML) can reduce the cost and time for producing poverty estimates so that anyone can monitor poverty and inequality easily and frequently. However as we show in this paper, a simple application of ML can cause large biases in the estimates of poverty and shared prosperity indicators. But, by combining it with other statistical techniques, like multiple imputation (MI), it can produce precise estimates of poverty and shared prosperity from a set of asset ownership and non-monetary conditions of living in a cost-effective and timely manner. Our study proposes several methodologies, where ML and MI are combined in a different way and compares their performances on poverty measurement. Then, we discuss the pros and cons of these methodologies and suggest the way to select the most appropriate methodology for measuring poverty in each specific context.
Keyword*: Machine Learning, poverty measurement, SWIFT, survey to survey implementation*

1. Introduction

Ending extreme poverty and promoting shared prosperity are raised as twin goals of the World Bank Group. The former pursues to reduce the percentage of the world's population whose household expenditures per capita are below $1.90 per day to 3 percent by 2030, while the latter will be measured by the growth rates of the poorest 40 percent of the population in each country. Both measures require household expenditure or income data collected by household sample surveys.

However, estimating these statistics is no simple undertaking, especially for developing countries that are a center of attention for monitoring of the twin goals. It requires collection of household consumption data and estimation of poverty and shared prosperity indices, which need time, money, and highly skilled manpower. For example, a typical household survey data collection takes two to three years from its preparation to completion. It also costs multi-million dollars. Furthermore, to estimate the indicators, a country needs to deal with the following technical questions:

- What items should be included in consumption aggregates?
- How are price differences adjusted?
- How are housing rents imputed?
- How is a service flow of consumer durables estimated?
- How are poverty lines estimated?

Answering these questions requires additional training after completion of standard training in statistics and economics. For example, poverty economists who provide technical assistance on poverty measurement to developing countries need to take a week-long course even though most of them have master or Ph.D. in Economics or Statistics. These requirements make it nearly impossible for poor countries to monitor poverty and shared prosperity indices frequently.

These requirements also make inclusion of the twin goals in Monitoring and Evaluation (M&E) of policy interventions and investment projects almost impossible. A typical cycle of these interventions and projects is three to five years. If the estimation of the twin goal indicators takes more than two years, it is often too late for the evaluation. Also, finding experts of poverty measurement for every single policy intervention or investment project is nearly impossible while spending multi-million dollars only for estimating the twin goal indicators is likely unjustifiable. As a result, inclusion of monitoring of the twin goal indicators as part of M&E explicitly is almost non-existent in projects in the World Bank.

Survey-to-survey imputation offers a potentially cost-effective solution for improving the frequency and comparability of poverty data. The idea is as follows; suppose consumption or income data are not collected in a particular year, but non-consumption data from other surveys are available, then consumption or income data can be imputed into the non-consumption/income data set using imputation models calibrated on consumption data collected previously, and the twin goal indicators are estimated

from the imputed data. Such an approach can be used to estimate the twin goals using households surveys without consumption or income data like Development Health Surveys (DHS). However, this approach cannot be applied for years when no household survey is carried out or for M&E of most projects, which are carried out in different years and areas from official household surveys.

The World Bank's ongoing initiative, SWIFT (Survey of Well-being via Instant, Frequent Tracking), fills the data gaps (Yoshida, et al., 2015). After developing imputation models from the latest household surveys with consumption or income data, like household budget surveys, SWIFT conducts a survey to collect variables in the models and then estimates the twin goal indicators from the variables collected by the SWIFT survey using the models. This approach is cost-effective and time-saving because (i) the number of variables selected by models is usually 15 to 20 and they can be collected in a 3 to 5 minutes interview; (ii) developing models takes less than one week and estimating the twin goals indicators using imputation models from 10,000 observations, a typical sample size of a national household survey, takes only 1 to 2 minutes. Programs for developing models are readily available and technical support for the implementation of SWIFT is available with a minimum cost. Beyond its cost-effectiveness and timeliness, SWIFT's estimations are found to be accurate as well. Evaluation using testing data show the estimates of poverty headcount rates are usually within a few percentage points from the true rates.

With SWIFT, it is no longer impossible to monitor the twin goals frequently, cost-effectively and accurately. Even for a M&E of one project, all the project team needs to do is to add a 3-5 minutes interview to a regular monitoring system, with which we can monitor the twin goal indicators of project beneficiaries.

However, SWIFT is not flawless. Chen et al. (forthcoming) show for some countries, the estimates of poverty rates are not accurate for some datasets. SWIFT adopts linear regression models and uses a stepwise selection process to identify imputation models and variables in the models. To deal with overfitting issues, SWIFT uses a 10-fold cross validation technique to decide the p-values of the stepwise selection. The cross validation is certainly effective in reducing the risk of overfitting, but it increases the time to complete the development of imputation models. To develop one model, SWIFT needs 5 to 6 hours.

The objective of this paper is to explore the latest Machine Learning technique to improve the accuracy of imputation models and reduce the time needed to develop models.

Recently, along with the unprecedented progress of the data science and increasing availability of ICT tools for data collection, machine learning (ML) has attracted enthusiastic attention in a field of development (see, for example, Athey [2018] or Athey and W.Imbens [2019]). It is strongly believed that ML could be a game-changing solution for the accurate and timely measurement of the poverty data, and the SWIFT algorithm could also be improved greatly using the power of ML. In this paper, we demonstrate how ML, integrated into the framework of SWIFT, could produce an accurate poverty data.

ML is a generic term for a wide variety of algorithms which detect some kind of patterns from a data set, which is generally quite huge, and conduct prediction or classification for the new data set. The idea of ML itself appeared in many academic fields since a long times ago, but the application of its algorithm became popular and influential in a last decade especially due to the significant improvement of the computing power. Although ML includes so many other powerful approaches, in this paper, we focus on applying two very popular algorithm of ML; regularized regression approaches and random forest.

1.1 Regularized regression approaches

Regularized regression approaches are those which impose the regularization on the magnitude of coefficients when minimizing the loss function. In the normal linear regression with n observations and p predictors, the dependent variable **y** could be written as;

$$\boldsymbol{y} = \boldsymbol{X\beta} + \epsilon \qquad (1)$$

where $\mathbf{y}=(y_1,y_2,\ldots,y_n)^T$, $X=(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_p)$ is a $n \times p$ matrix of independent variables, and ϵ is a residual term. In the case of p < n, the coefficients $\boldsymbol{\beta}=(\beta_1,\beta_2,\ldots,\beta_p)$ can be estimated by minimizing the loss function,

$$L(\boldsymbol{\beta}) = (\boldsymbol{y} - \boldsymbol{X\beta})^T(\boldsymbol{y} - \boldsymbol{X\beta})$$

which leads to the ordinary least squares (OLS) estimator. However, in the cases of p > n, the above approach does not work. Also, even if p<n, if n is small or independent variables are highly correlated, predictions based on OLS estimators are vulnerable to small sample biases, such as overfitting. To overcome these issues, the regularized regression approach solves the following minimization problem:

$$\mathrm{Min}_{\beta}(\boldsymbol{y} - X\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X\beta}) + p_\lambda(\boldsymbol{\beta})$$

where $p_\lambda(\beta)$ is a regularization function or simply a

penalty against OLS estimators. Several useful forms of $p_\lambda(\beta)$ are known, leading to different regularized estimators. Some popular forms include ridge, $p_\lambda(\beta) = \lambda \sum_{j=1}^{p} \beta_j^2$, lasso, $p_\lambda(\beta) = \lambda||\beta||$, elastic net, $p_\lambda(\beta) = \lambda_1\big||\beta|\big| + \lambda_2||\beta||^2$. All these regularization functions add penalties against OLS estimators, such as λ and λ_i and as the penalty becomes bigger, the regularized regression coefficients shrink toward 0. This is a reason why the regularized regression estimators are sometimes called shrinkage estimators. In addition to shrinkage, each regularization function is known to have different characteristics. The ridge penalty adds the robustness to the multicollinearity and the lasso penalty is popular when there are too many predictors, as lasso enables the reduction of the predictors and estimation of the coefficients at the same time. Elastic net has mixed characteristics of ridge and lasso.

1.2 Random forest

Random forest is known as one of ensemble learning methods among machine learning techniques and probably one of the most popular algorithms used in the application of the ML. As stated above, ML makes a prediction on the big data set, based on computational algorithms which by themselves construct the model, learning from the data. When the prediction is about discrete data, the problem becomes classification, while when it is about continuous variables, it is called as a regression. The models constructed by algorithms in ML are often called as classifier or learner. In ML, the data is typically split into a training set from which the algorithm learn the model and a test set to which the model constructed by the algorithm is applied. ML contains numerous methods, and among them, the ensemble learning is characterized by an approach to predict explanatory variables at higher accuracy, combining the results drawn from weak learners which themselves do not have strong prediction accuracy.

Typical example of ensemble learning is bagging. Bagging algorithm consists of mainly three steps. Given a vector of observations $Y=(y_1,y_2\ldots,y_n)^T$ and set of explanatory variables as a matrix $X=(x_1,x_2\ldots,x_p)$, where $x_i=(x_{1i},x_{2i}\ldots,x_{ni})^T$ for i = 1,2…p, bagging works as follows;

1) Randomly sample n observations from X and Y, allowing the replacement and call the new sample a training dataset. Due to the replacement, some portion of the original dataset (X, Y) are not included in the training dataset. This remaining dataset is called an "Out of Bag" (OOB) dataset and is used as a testing data to evaluate the fitness of predictions of Random Forest. Using the training dataset, construct the model of classification or regression, which is referred to as a (weak) learner ϕ
2) Repeat 1) for b times and get a weak learner ϕ_j, where j=1,2,…b
3) These ϕ_j are applied to a test set constructed from OOB data. Then the final prediction is made by a strong learner f, which averages the predictions from all the weak learners ϕ_j;

$$f(X) = \frac{1}{b}\sum_{j=1}^{b} \phi_j(x)$$

Averaging the weak learners is implemented by taking the mean of values predicted in the case of regression or by taking the majority vote when it is a problem of classification. The difference between the predicted and actual value in OOB data is referred to as an OOB error and is used for the model selection with different parameters.

One of popular methods used for developing a weak learner is a classification and regression tree (CART). CART is an approach for predicting or classifying variables Y, by repeatedly partitioning the data according to the certain value of explanatory variables X. It is called as a "tree" because the partitioning of the data is often shown as a tree-like structure, as exemplified in Figure 1. The figure illustrates the result of CART using the famous data set *Iris*, contained in R software[1]. *Iris* includes the length and width of the sepal and petal of three different species of iris, *setosa*, *versicolor* and *virginica*. In this example, the objective of the model is to classify the species of iris according to the information on its sepal and petal. The result of CART consists of nodes which determine the threshold value of X to split the data, and branches growing out of the nodes. The algorithm tries to detect these X values at the branch point so that the subset of y, grouped at the terminal of the node becomes homogeneous. In this case, 1.9 of the Petal Length, 1.7 of the Petal Width and 4.8 of the Petal Length are the thresholds detected by the algorithm. According to Figure 1, one can see that the node 1 managed to distinguish the group of *setosa*, node 3 with *Virginica* and node 4 with *Versicolor*. Although this is the example of classification, it can also be used to predict the continuous variables using explanatory variables, which is implemented by a regression tree. Unlike the normal linear regression analysis, the regression tree has a merit in capturing the non-linear relationship between X and Y, as it

[1] https://www.r-project.org/

does not assume any linear relationship between variables when searching for the branch point of the tree.

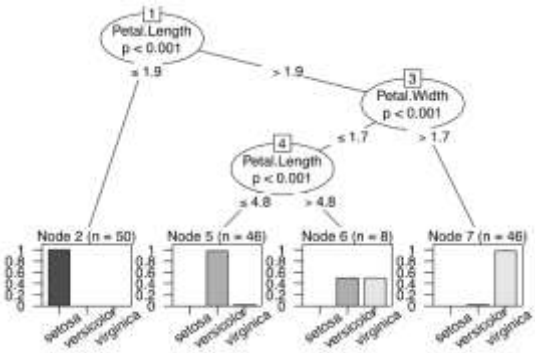

Figure 1: Example of CART

Random forest is an extended approach of bagging, using CART for developing a weak learner (or a tree). and makes a prediction from a strong learner - an ensemble of the results from weak learners (or trees), for improving the accuracy of the prediction. This is why it is called as a "forest". Another feature of random forest is double randomization. As described above, as part of bagging, b samples are created by random selection of n observations with replacement. Random forest has another randomization – random selection of a subset of explanatory variables instead of using all $X=(x_1,x_2\ldots x_p)$ for each sample. Each weak learner is thus constructed from a different subset of explanatory variables, which reduces correlations between weak learners and the variance of the final predictor – a strong learner (see more details in Hastie et al., 2007). Besides, as random forest is an ensemble approach which try to predict y variable combining the results from hundreds of regression/classification trees, it is especially good at detecting the non-linear relationships between Y and X variables.

2. Data and Evaluation Methodology

We used Uganda National Household Survey (UNHS) data of two different years (2009, 2012), where both consumption and non-consumption data are included. First, the models for imputations between household expenditures in natural log and covariates including the assets of the household, educational level of the household head and number of household members, etc., are developed from the 2009 round of data by using the current poverty prediction approach (SWIFT) and ML algorithms including regularized regression approaches and random forest, and the modifications. Then applying these models into the 2012 round of data, poverty rates at the national level are estimated and compared with the actual ones calculated from the household consumption data.

3. Issues to be dealt with when applying ML for poverty measurement

The simple applications of the ML approaches for calculating the poverty measures such as the poverty rate or mean income for the bottom 40% face large biases. Figure 2 shows what will happen when we simply apply the regularized regression approach, which is elastic net in this case, and the random forest using the Ugandan consumption data.

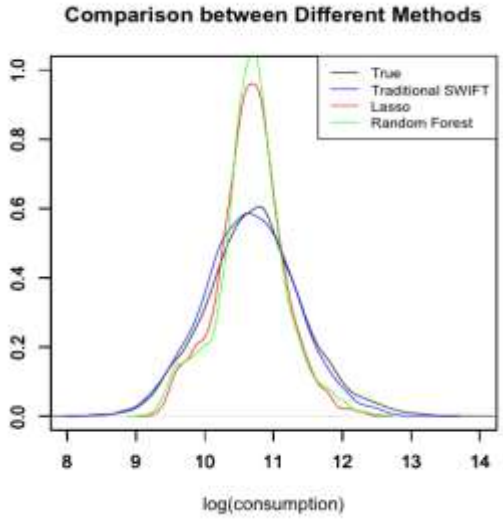

Figure 1: Comparison between SWIFT and simple

The black line shows the true distribution of the log of consumption expenditure, and the blue line depicts the distribution predicted by the traditional SWIFT model. Compared with these two distributions, the ones of elastic net and random forest are much narrower. As a result, if the ML estimators are used for estimating poverty rates or the mean household expenditure of the poorest 40 percent of population, both statistics face large biases.

This is due to the fact that ML is designed to produce only the prediction of the log of household consumption expenditure. It is however clear from equation (1) that the variance of log of household expenditure is a sum of variances of predicted expenditures and of residuals. But since ML approaches, both regularized regression approaches and random forest, produce only the predictions, the variance of the predictors is significantly smaller than the actual one. On the other hand, SWIFT and a typical survey to survey imputation technique take into account the variances of predictions and errors, the distribution of imputed household expenditures

(in log), as discussed below, is much closer to the actual one than ML predictors (see Yoshida et al., 2015).

Ignoring the variance of the residuals causes large biases in estimation of poverty rates and the mean household expenditures of the poorest 40 percent of population. This implies that in order to estimate unbiased poverty rates and shared prosperity indices, we need to modify ML approaches so that the impact of residuals needs to be properly considered.

4. Possible solution

Actually, the above issue occurs even if we just use the normal OLS regression. However, in the SWIFT, we solved this issue by combining the OLS with multiple imputation (MI). MI was originally developed to handle the missing data[2]. The key idea of MI is to replace each missing value with a set of plausible values drawn from the predictive distribution conditional on the observed data and then generate the multiple imputed data sets to account for uncertainty of imputing missing values. In case of the normal linear regression, the procedure of the MI can be described as follows.

Considering a univariate variable $x=(x_1,x_2,\ldots,x_n)$ with p variables $Z=(z_1,z_2,\ldots,z_p)$ that follows a normal linear regression model;

$$x_i|z_i \sim N(z_i'\beta, \sigma^2) \quad (2)$$

Let Z_o denote the observed components of Z and Z_m denote the missing components.

1. Fit a regression model (2) to the observed data (x_0, Z_0) to obtain estimates β and σ^2 of the model parameters.
2. Simulate new parameters β_* and σ_*^2 from their joint posterior distribution;
$$\sigma_*^2 \sim \hat{\sigma}^2(n_0 - q)/X_{n0-q}^2$$
$$\beta_*|\sigma_*^2 \sim N\{\hat{\beta}, \sigma_*^2(Z_o'Z_o)^{-1}\}$$
3. Obtain one set of imputed values, X_m^1 by simulating from $N(Z_m\beta_*, \sigma_*^2 I_{n1\times n1})$
4. Repeat steps 2 and 3 to obtain M sets of imputed values, $X_m^1, X_m^2, \ldots, X_m^M$

In the following sections, we propose a way to combine the MI with our regularized regression approaches and random forest to integrate the error term.

4.1 Combining regularized regression approaches with MI

The first step of MI described above can be done easily with the regularized regression approaches. However, the issue comes under the step 2. Let θ be the unknown model parameters, which in this case $\theta = (\beta, \sigma)$. When the regularized regression approaches are used, it is not easy to derive the distribution $f(\theta|Z_{obs})$ mathematically as done in the above step 2. Therefore, it is required to obtain the distribution of θ somehow empirically, which would be enabled by using a bootstrap data as follows (Deng, et al., 2016).

(1) Generate a bootstrap data set Z of size n by randomly drawing n observations from Z with replacement.
(2) Use a regularized regression method to fit the model $\boldsymbol{Z_{o,1} = Z_{o,-1}} + \epsilon$ based on the Z_o, and obtain parameter estimate $\hat{\theta}$, noting that $\hat{\theta}$ can be considered a random draw from $f(\theta|Z_o)$.
(3) Impute $Z_{m,1}$, with $Z_{m,-1}$ by drawing randomly from the predictive distribution $f(Z_{m,1}|Z_{m,-1}, \hat{\theta})$, which is in this case, $N(Z_m\hat{\beta}, \hat{\sigma}^2 I)$.

And we can repeat the above procedure for M times results to obtain M imputed data sets.

4.2 Combining random forest with MI

We can combine random forest with MI in a similar way with regularized regression approaches, but in random forest, we do not obtain β, but $\hat{E}(Y|X_{1,\ldots,}X_p)$[3]. So the procedure looks like as follows (Shah et al., 2013).

(1) Generate a bootstrap data set Z of size n by randomly drawing n observations from Z with replacement
(2) Standard random forest is applied to $(Z_{o,1}, Z_{o,-1})$, which gives $\hat{E}(Y|X_{1,\ldots,}X_p)$
(3) Missing Y values are imputed by taking a normal draw centered on $\hat{E}(Y|X_{1,\ldots,}X_p)$ and residual variance equal to the "out of bag" mean square error

[2] For details on different algorithms on MI, see, for example, Harel and Zhou (2006), Kropko et al. (2013), Bertsimas et al.(2018)

[3] In other words, we obtain the group of trees which return Y taking the inputs of $(X_{1,\ldots,}X_p)$

By using a bootstrap data set, it is possible to combine ML with MI. Figure 3 shows the distributions of the consumption using the traditional SWIFT, elastic net combined with MI, and random forest combined with MI. It can be seen that the performance of the regularized regression approach, which is elastic net in this case, and random forest improved greatly because they managed to take into account the error term properly.

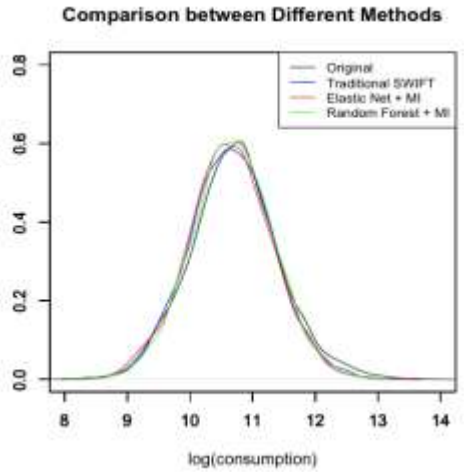

Figure 2: Comparison between SWIFT and ML+MI

But just from this graph visually comparing the multiple distributions, it is difficult to say which methodology fits the best, as some methods are good at capturing the lower tail, while others perform good at the higher tail of the distribution. To grasp the overall performance of the methodology, the absolute gap between the true poverty rate and the predicted one, when the poverty line is moved from 1% quantile of the distribution up to 100% with increase of 1%, which produces 100 numbers for each methodology, was calculated, as shown in Figure 4.

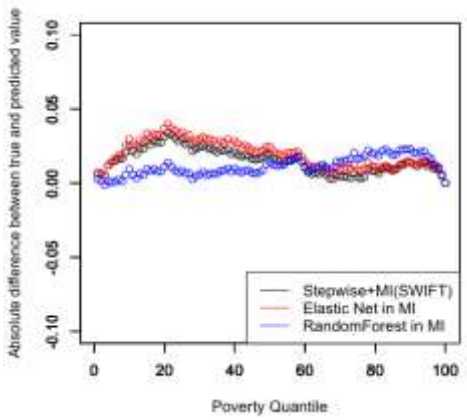

Figure 4: Predicted values VS True value

Figure 5 shows the mean and maximum absolute difference between true and predicted poverty rate. In this example, the random forest+MI appears to be the best methodology with the least error.

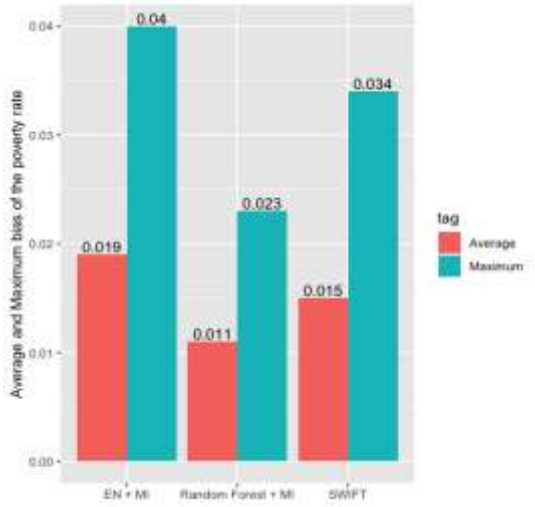

Figure 3: Summary of the overall performance

5. Need for variable selection

Now we confirm that ML could be very promising for the accurate poverty measurement. Nevertheless, there remains an issue for regularized regression approaches and random forest. In the typical SWIFT problem setting, the model is constructed using a data set of year 0 and in the model construction, the number of variables required to estimate the poverty rate normally drops drastically to around 15 to 20 due to the stepwise process. Then in year 1, we only have to collect the data for these 15 to 20 variables, which reduces the interview time and cost of the survey and this is one of the biggest advantages of the SWIFT approach.

However, in the new ML+MI approaches described above, there is no explicit process of variable selection. In regularized regression approaches, for instance, if we use LASSO, the number of variables selected will be reduced in each bootstrapped data, but the group of variables selected is slightly different across different bootstrapped data, and so the process as a whole does not reduce the number of variables required. In the algorithm of random forest+MI, there is not any variable reduction in the end, either[4]. Therefore, we need some process of variable selection in order to practically make use of it in actual SWIFT programs. A question is whether limiting the number of variables used for ML

[4] Although we use limited number of variables in each tree, overall, we are using all the available variables.

approaches can reduce the accuracy of the estimation of the twin goal indicators.

5.1 Variable selection for regularized regression

Regarding the regularized regression approaches, one possible solution is to conduct some statistical test based on the coefficients. After applying the regularized regression on the bootstrapped data, we obtain the distribution of coefficients [5] for each predictor. Then we can select only those variables with coefficients, of which 95% confidence interval does not contain 0. In principle, this procedure is the same as typical statistical tests for coefficients with a significance level of 5 percent.

Figure 6 compares the performance of the normal Elastic Net (EN) +MI and the one with variable selection conducted in a way described above. In this case, the normal EN +MI uses all the 55 variables, while the EN + MI with the variable selection uses only 22 variables. Interestingly, as shown in both figures, the performance slightly improved with the process of variable selection.

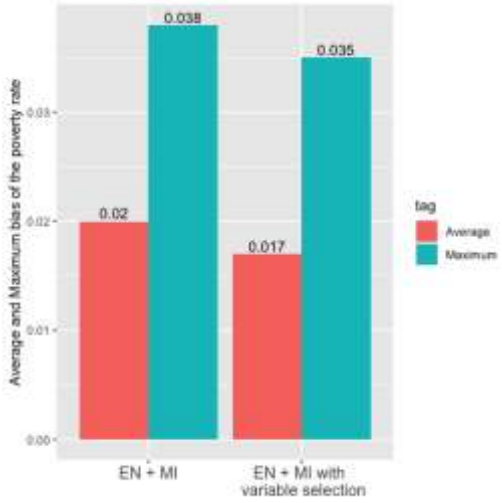

Figure 5: EN+MI with variable selection

5.2 Variable selection for random forest

For random forest, different algorithms are proposed for the variable selection, but probably the most straightforward way is to use the information of the variable importance, which can be calculated in random forest. Often used variable importance is a *permutation importance* (PI), which measures how much the accuracy of the prediction will get worse if the particular x variable is randomly permuted.

[5] For LASSO and Elastic Net, 0 will be assigned to those coefficients which were not selected.

Formally, PI of a specific predictor x_j calculated from tree t, with t=1,2…T, can be written as follows;

$$VI^t(X_j) = L(y^t, \hat{y}^t) - L(y^t, \hat{y}^t_\pi)$$

, where y^t, $\hat{y}^t$ and $\hat{y}^t_\pi$ are observed actual y value, predicted y value in tree t, and predicted y value for tree t after the permutation of X respectively. L() is a loss function, which calculates the error between actual y value and predicted y value. In terms of the classification, misclassification rate or gini impurity is often used as a loss function, while for regression, mean squared error is used. Then, the importance of variable x_j is finally computed by averaging $VI^t(x_j)$ across all trees;

$$VI(X_j) = \frac{1}{T}\sum_{t=1}^{T} VI^t(X_j)$$

Using the above variable importance, we can rank the variables in the order of importance of each variable in relation to the dependent variable. Several methodologies are proposed to select variables systematically based on this ranking[6], but here, we show the results of the simplest approach, in which we simply pick up the first top 10 or 20 variables from that ranking and compare their performances. Figure 7 illustrates the comparison between the normal RF+MI, RF+MI with top 10 variables and RF+MI with top 20 variables.

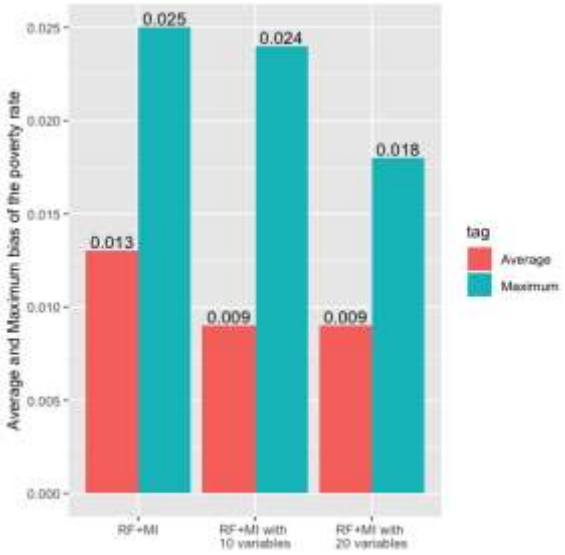

Figure 6: RF+MI with variable selection

It is interesting that the performance is better with

[6] One of the popular algorithms for selecting variables using the variable importance calculated by Random Forest is VSURF (Genuer, Poggi and Tuleau-Malot, 2015), which can be implemented in R software.

variable selection, which is encouraging when considering the application of this algorithm for the SWIFT program, as this result implies that we need only a part of the variables in order to get the better results. In this case, the RF+MI with 20 variables is the best among all other methodologies with the average gap between the true poverty rate of 0.009 and the maximum gap with 0.018, which is a significant improvement from the original SWIFT.

6. Robustness against the multicollinearity

One of the issues of SWIFT or other methodologies using stepwise for variable selections is multicollinearity between the explanatory variables. To see the robustness of the different modified ML approaches against the multicollinearity, we conducted the above exercise including the square, cube, the fourth and fifth power of the household size cumulatively and compared the performances by the mean error of the poverty rate, which is illustrated in Figure 8.

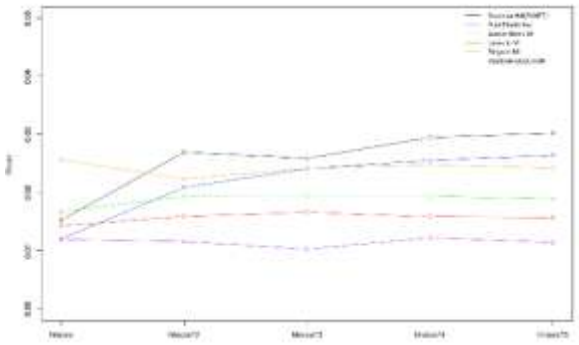

Figure 8: Robustness against the multicollinearity

It can be seen that the performance of traditional poverty approaches using OLS framework (SWIFT, Post Lasso where the explanatory variables are selected using lasso instead of stepwise) deteriorate suddenly after including the term of square and gets worse gradually along with the additional power term. On the other hand, the performance of regularized regression approaches, no matter what the penalty terms are, and random forest is stable irrespective of the addition of the power terms, which clearly indicates the robustness of these ML approaches against the multicollinearity.

7. How to select the best methodology

Now we know that different approaches of machine learning could improve the poverty measurement significantly. However, there still remains an issue on how to select the best methodology. For instance, the random forest is essentially non-linear, which is fundamentally different from other linear regression approaches such as original SWIFT or regularized regression approaches. Whether the linear or non-linear methods should be selected depends on the data structure. In the previous case study using Uganda data, we figured out that the best performer was random forest+MI if we use the difference between the predicted poverty rates and true rates. But in a real setting, nobody knows the true poverty rates. Therefore, we need some procedure to determine which methodologies to be taken only from the training data. Figure 9 illustrates the proposed solution.

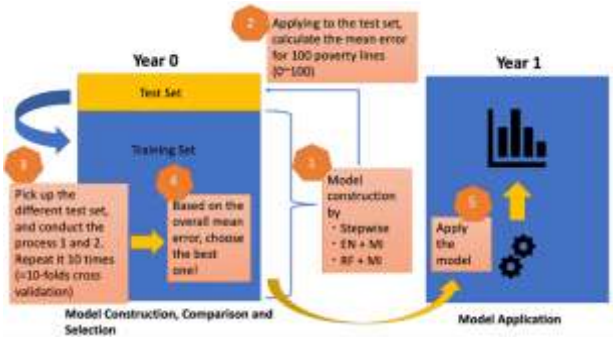

Figure 9: How to determine the best methodology?

Let's say we have the consumption data in year 0 and tries to predict the poverty rate in year 1. Then the first step will be to divide the data set in year 0 into 10 folds, randomly select one fold as a test set, while putting other folds as a training set. Then the models are constructed from these training set using different methodologies such as stepwise, EN and RF. Then we apply these models to the test set and calculate the mean error of poverty rate for 100 poverty lines like we did in the previous Uganda data. As a third step, we pick out the different test set and training set, and repeat the step 1 and 2 for 10 times. Now we have the 10 different numbers for each three methods, so we take the overall mean error and choose the best one with the least mean error of the poverty rate. Then finally, we apply that model for the year 1 data set to predict the poverty rate.

8. Limitations and future research

Although these approaches described above seem to be very promising, there are still limitations and space for the improvement. First of all, in the above new methodologies, we focus on the prediction of the consumption data which is normally distributed. In the process of MI under the framework of OLS, we assume two normal distributions for β and Y, which are $\beta_*|\sigma_*^2 \sim N\{\hat{\beta}, \sigma_*^2(Z_o'Z_o)^{-1}\}$ and $Y \sim N(z_i'\beta, \sigma^2)$. On the other hand, in the EN+MI and RF+MI approach, we do not have the former assumption[7], but

[7] This is because for regularized regression approaches, we

still we assume that each observation comes from the normal distribution when imputing. The additional research is required to clarify the extent of the effect on the prediction accuracy of the new methodologies when the assumption of the normality is violated.

In the above examples, we use the regularized regression approaches and random forest, both of which are some of the most popular algorithms of machine learning. However, there are also other powerful approaches, such as support vector machine and neural network. There exists some research on using these tools for imputing missing data[8], and it would be interesting to try these novel methods for the poverty measurement using not only the traditional sets of explanatory variables, but also image data, such as satellite image or pictures of houses taken by the drone, which is difficult to handle with traditional approaches.

References

Athey S. The Impact of machine Learning on Economics. 2018

Athey S and Imbens GW. Machine Learning methods Economists should know. 2019

Bertsimas D, Pawlowski C, Zhuo YD. From Predictive methods to Missing Data Imputation: An Optimization Approach. Journal of Machine Learning Research 18(2018) 1-39

Dang HA, Jolliffe D, Carletto C. Data Gaps, Data Incomparability, and Data Imputation. A Review of poverty measurement methods for Data-Scarce Environments. WB 2017

Degenhardt F, Seifert S, Szymczak S, Evaluation of variable selection methods for random forests and omics data sets, Briefings in Bioinformatics 2019, 20(2); 492-503

Deng, Yi, et al. "Multiple imputation for general missing data patterns in the presence of high-dimensional data." *Scientific reports* 6 (2016): 21689.

Fujii T and van der Weide R. Is predicted data a viable alternative to real data?

Genuer R, Poggi JM, Tuleau-Malot C. Variable selection using Random Forests. Pattern Recognition Letters, Elsevier, 2010, 31(14). 2225-2236

Harel O and Zhou XH. Multiple imputation-Review of theory, implementation and software. UW Biostatistics Working Paper Series. 2006.

Hastie, T., R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning – Data Mining, Inference, and Prediction.* 2007. Springer. New York. US.

Kshirsagar V,Wieczorek J, Ramanathan S, Wells R. Household poverty classification in data-scarce environments: a machine learning approach. 31st Conference on Neural Information Processing Systems (NIPS 2017)

Kropko J, Goodrich B, Gelman A, Hill J. Multiple Imputation for Continuous and Categorical Data: Comparing Joint and Conditional Approaches. 2013

Liu Y and Gopalakrishnan V. An Overview and Evaluation of Recent Machine learning Imputation Methods Using Cardiac Imaging Data. 2017

Pape U, Fujii T, and Mistiaen J. Household Expenditure and Poverty measures in 60 minutes: A new approach with results from Somalia. 2018

Shah AD,Bartlett JW, Carpenter J, et al. Comparison of Random Forest and Parametric Imputation Models for Imputing Missing Data using MICE: A CALIBER Study. *American Journal of Epidemiology* 2014; 179(6):764-774

Yoshida, N., R. Munoz, A. Skinner, L.C. Kyung-Eun, M. Brataj, William, D.S., D. Sharma D. 2015. "Survey of Well-Being via Instant and Frequent Tracking (SWIFT) Data Collection Guidelines." Washington, D.C. World Bank Group. http://documents.worldbank.org/curated/en/591711545170814297/Survey-of-Well-Being-via-Instant-and-Frequent-Tracking-SWIFT-Data-Collection-Guidelines

Zhao Y and Long Q. Multiple imputation in the presence of high-dimensional data. *Statistical methods in Medical Research* 2015;0(0):1-15

obtain the empirical distribution of β using bootstrapped data. As for the random forest, we do not have β, but get $\bar{Y}$ directly from each tree.

[8] See for instance, (Bertsimas, Pawlowski and Zhuo, 2018)