Predicting Self-Response Rates

Via generalized additive models with interactions

Emanuel Ben-David

In collaboration with Shibal Ibrahim, Rahul Mazumder and Peter Radchenko

Center for Statistical Research and Methodology
US Census Bureau

April 5, 2022

FedCASIC
2022 Federal Computer Assisted Survey Information Collection Workshops

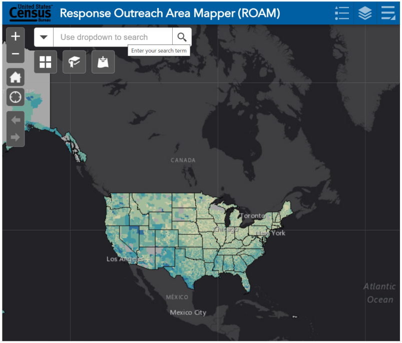
Shape your future START HERE >
United States Census 2020

Background

- Since the 1990s, the Census Bureau has been working on developing models for predicting the self-response rate for each census tract (or block) based on the characteristics of the tract (or block).
- These models are used to identify hard-to-count (HTC) areas in preparation for the next decennial census.
- The first model was developed for predicting a HTC score in planning the 2010 Census.
- The current model is a linear regression model that includes 25 covariates form the 2014 Planing Data Base (PDB).
- These are variables highly correlated with the self-response rate.

The Response Outreach Area Mapper

The Response Outreach Area Mapper (ROAM) released in 2019 is an application for predicting the Low Response Score (LRS), a metric for Hard-to-Survey Populations, based on a similar linear regression model.



The Kaggle competition

- In 2012, the U.S. Census Bureau carried out a crowd-sourcing competition through the Kaggle.com to explore the best machine learning methods for predicting the 2010 Census self-response rates.
- The challenge was to predict 2010 Census mail return rates using the 2012 Census Planning Database (PDB) and any other publicly available sources of data.
- Although models based on ensembles of regression trees won the challenge, they were not found interpretable and useful for the intended applications.

Using Generalized Additive Models (GAMs)

- We propose a new model based on Generalized Additive Models with interactions via ℓ_0 regularization.
- We show that these models are:
  - interpretable
  - effectively predictive, comparable with the state-of-the-art black-box machine learning methods such as XGBoost and deep learning for identify hard-to-survey populations
  - amenable to automatic variable selection in high-dimensional regression

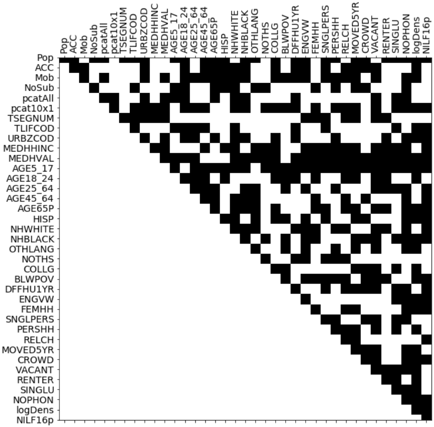
A simple description

- The standard two-way interaction GAM is given by

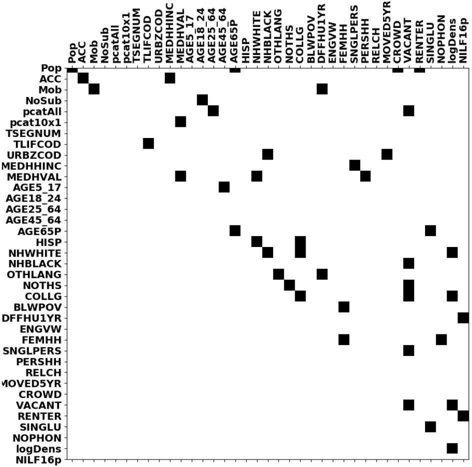
$$g\left(\mathbb{E}(y|\mathbf{x})\right) = \sum_{j\in[p]} f_j(x_j) + \sum_{j<k} f_{j,k}(x_j, x_k).$$

- As a generalized linear model, g is the link function.
- Functions f_j and $f_{j,k}$ are unknown and need to be estimated from the data.
- A key problem in the estimation of two-way interaction models is the explosion in the number of unknown parameters.
- To facilitate interpretation, we advocate a parsimonious model, hence a model with a small number of main and interaction effects.

The sparsity pattern of a linear model with main and interaction effects



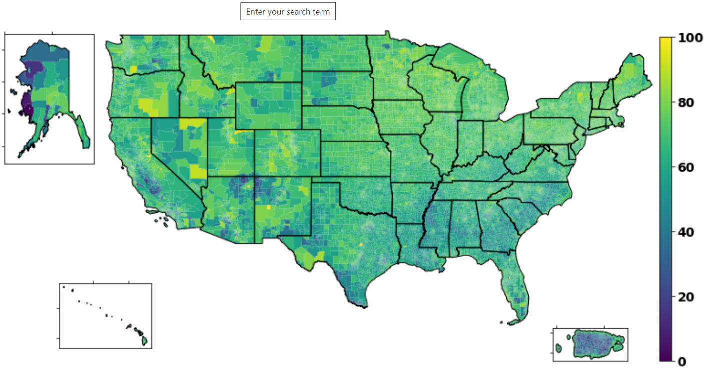
The sparsity pattern of a GAM with main and interaction effects



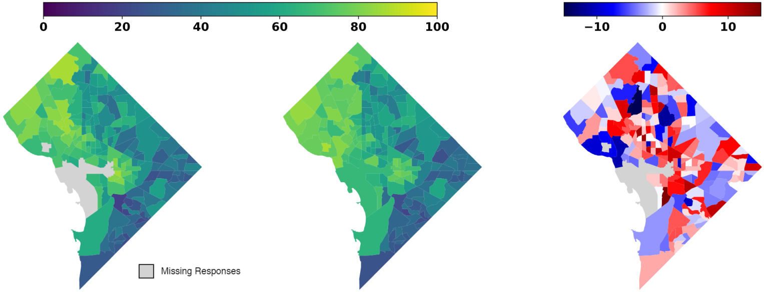
Performance compared with several benchmark models

Type	Model	RMSE	MAE	#Covariates
Linear Models (LMs)	Ridge	6.804 (0.080)	5.254 (0.051)	295
	Lasso	6.803 (0.080)	5.254 (0.051)	221
	L0Learn ($\ell_0 - \ell_2$)	6.813 (0.080)	5.268 (0.051)	136
	LM+Interactions (Lasso)	6.528 (0.077)	5.026 (0.049)	264 (76 Mn + 1598 Int)
	LM+Interactions with Strong Hierarchy (hierScale)	6.621 (0.078)	5.086 (0.049)	276 (276 Mn + 4885 Int)
Nonparametric Additive Models (AM)	AM under ℓ_0 (ours)	6.593 (0.078)	5.120 (0.049)	182
	AM+Interactions under ℓ_0 (ours)	**6.467** (0.077)	**4.973** (0.049)	**160** (16 Mn + 174 Int)
	AM+Interactions with Strong Hierarchy (ours)	**6.452** (0.076)	**4.995** (0.049)	**131** (131 Mn + 173 Int)
Nonparametric (Non-interpretable)	**XGBoost**	**6.440** (0.076)	**4.973** (0.049)	295
	Feedforward Neural Networks	6.501 (0.077)	4.996 (0.049)	295

Predicted ACS self-response rates for the US

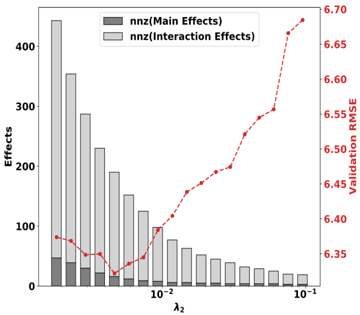


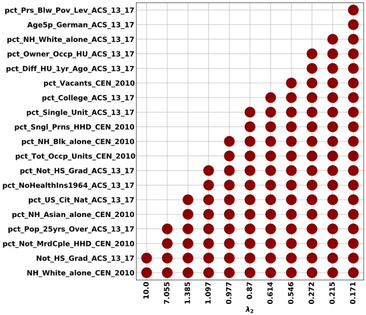
Predicted ACS self-response rates for Washington DC



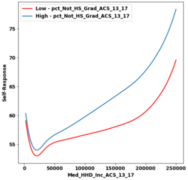
(a) Actual (b) Predicted (c) Difference

Regularization path of the GAM

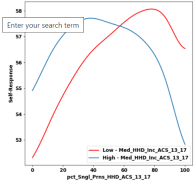




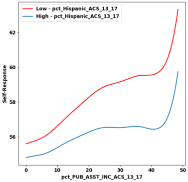
Interaction plots for the GAM



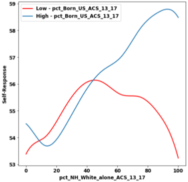
(a)



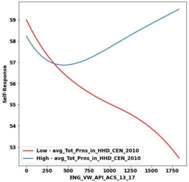
(b)



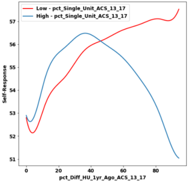
(c)



(d)



(e)



(f)

References

- **The main paper:** https://arxiv.org/abs/2108.11328
- **The source code in python:** https://github.com/ShibalIbrahim/Additive-Models-with-Structured-Interactions

Thank You!