EDWARD P. FITTS DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING
NC STATE UNIVERSITY

# Improved Feature Selection with Simulation Optimization

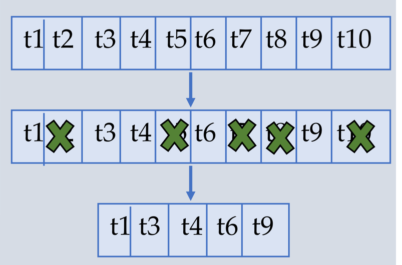
Kimia Vahdat (kvahdat@ncsu.edu) and Sara Shashaani, PhD (sshasha2@ncsu.edu)

Searching in a dataset for the "best" among many features is worthwhile for better and more timely predictions, and interpretability of the underlying system. Current automated methods depend on learning algorithm or produce weak and highly variable feature subsets.

A simulation optimization method with bootstrapping captures uncertainty in the dataset and, for any learning algorithm, generates reliable feature subsets that outperforms the benchmark at the cost of sampling. Number of bootstraps plays a crucial role and smart adaptive sample size selection significantly improves the optimal results.

## Why Feature Selection?

| Best $x$ (method) | # features | IS performance est. | OOS performance est. |
|---|---|---|---|
| $x_1$ (without FS) | 20 | **128** | 367 |
| $x_2$ (Benchmark) | 6 | 173 | **176** |




*What can go wrong with current feature selection methods?*

| Best $x$ (method) | # contributing feat. | # redundant feat. | # uninform. feat. | (CV) Performance |
|---|---|---|---|---|
| $x_2$ (Benchmark) | 0 | 3 | 3 | **176** |
| $x_3$ | 10 | 4 | 0 | 185 |

$\boldsymbol{x}$ often chosen with *greedy search* and *cross-validation* but it can poorly evaluate the performance

## Problem Statement

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) := \mathbb{E}_{F_1}\left[\mathbb{E}_{F_0}[Q(r(\boldsymbol{x}'\boldsymbol{t}_0|\boldsymbol{z}_1),\boldsymbol{y}_0)]\right]$$

Out-of-sample $\boldsymbol{z}_0 \sim F_0$

In-sample $\boldsymbol{z}_1 \sim F_1$

Estimate $f(\boldsymbol{x})$ with

$$\hat{f}_n(\boldsymbol{x}) = n^{-1}\sum_i Q\left(r\left(\boldsymbol{x}'\boldsymbol{t}_{\boldsymbol{z}(P_i)}\middle|\boldsymbol{z}(M_i)\right),\boldsymbol{y}_{\boldsymbol{z}(P_i)}\right)$$

CRN

100

| Bootstraps | 1 | 2 | ... | $N$ |
|---|---|---|---|---|
| training | $M_1$ | | ... | $M_N$ |
| testing | $P_1$ | | ... | $P_N$ |

$M_i, P_i := M(\zeta_i), P(\zeta_i)$

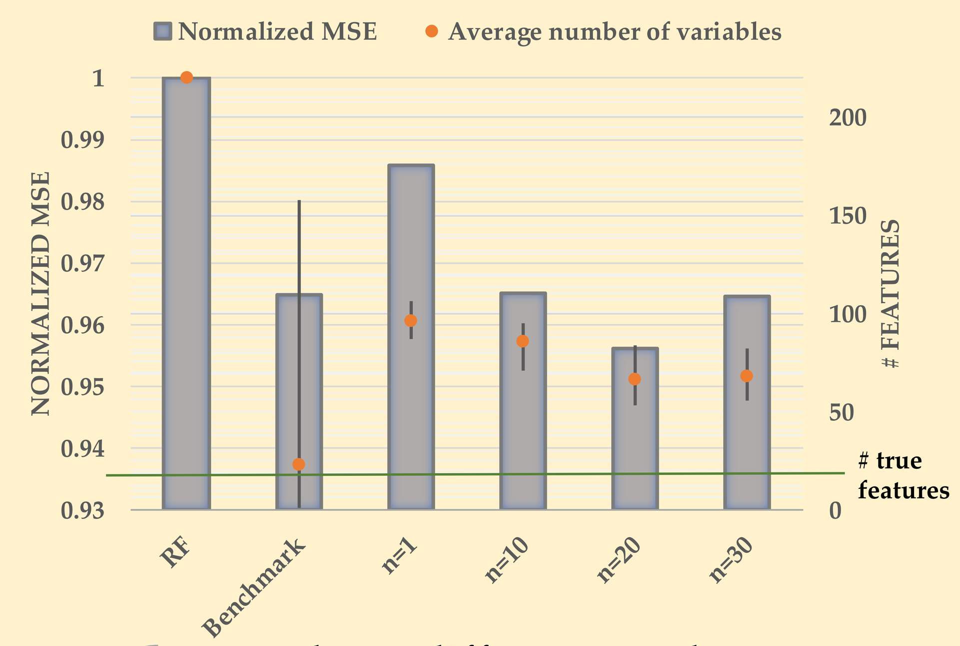
$\zeta_i \sim Unif[0,1]$

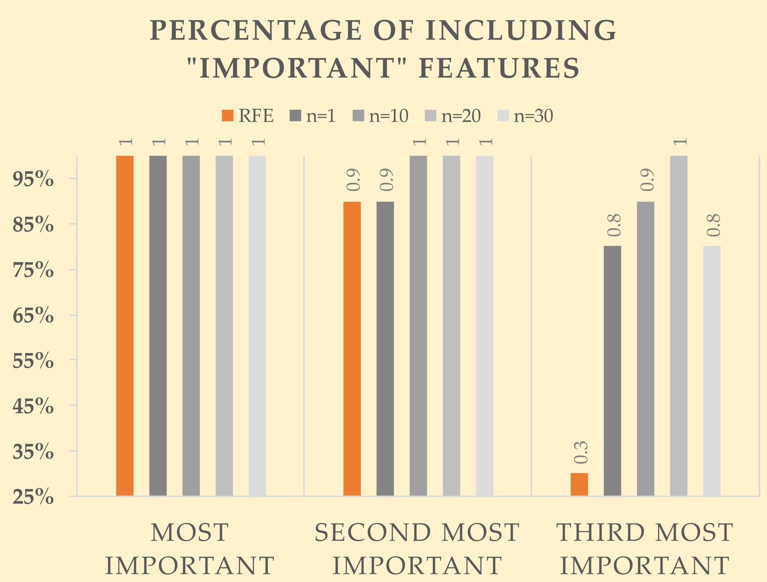
- $\boldsymbol{z}_i = (\boldsymbol{t}_i, y_i) \sim F_{\boldsymbol{z}}$ are given data points;
- $\boldsymbol{t}_i = (t_i^1, \dots, t_i^p)$ features,
- $\boldsymbol{x} = (x^1, \dots, x^p)$ $\boldsymbol{x}_i = \{0,1\}$,
- $M_i, P_i \subseteq \{1,2,\dots,|\boldsymbol{z}|\}$ index of points in bootstrap,
- $r(\boldsymbol{x}'\boldsymbol{t}|\boldsymbol{z})$ a prediction model,
- $Q(r(\boldsymbol{x}'\boldsymbol{t}|\boldsymbol{z}),\boldsymbol{y})$ deviation of predicted from observed.

$Q(.)$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $t_1^1$ | $t_1^2$ | ... | $t_1^p$ | $y_1$ | $r_{\boldsymbol{x}}(\boldsymbol{t}_1\|\{\boldsymbol{z}_i\}_1^m)$ | $(r_{\boldsymbol{x}}(\boldsymbol{t}_1\|\{\boldsymbol{z}_i\}_1^m) - y_1)^2$ | In Err |
| $t_0^1$ | $t_0^2$ | ... | $t_0^p$ | $y_0$ | $r_{\boldsymbol{x}}(\boldsymbol{t}_0\|\{\boldsymbol{z}_i\}_1^m)$ | $(r_{\boldsymbol{x}}(\boldsymbol{t}_0\|\{\boldsymbol{z}_i\}_1^m) - y_0)^2$ | Out Err |

## Does choice of $n$ make a difference?



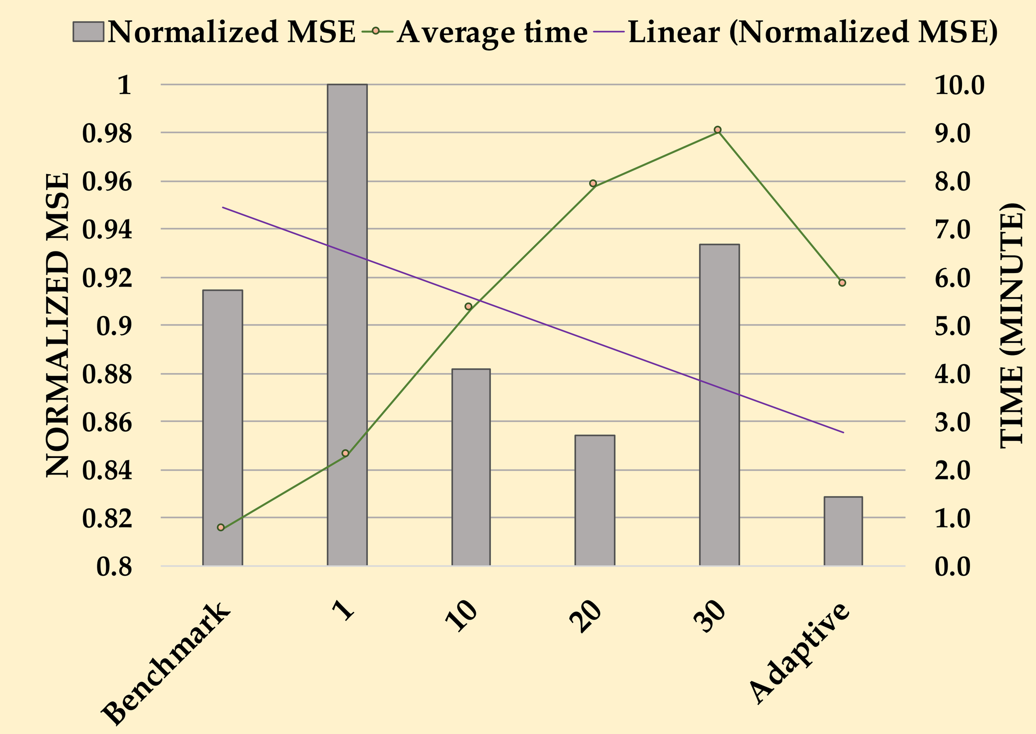

- $n$ makes a difference and
- beats benchmark in:
  - Sharpness
  - Accuracy

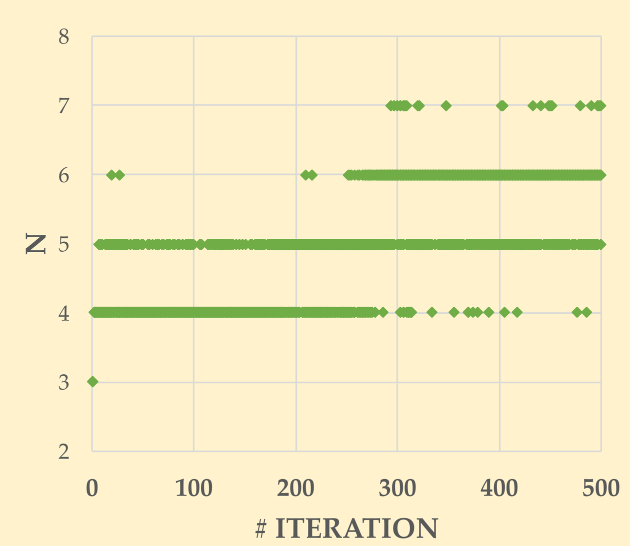
- No monotone relationship
- Very time consuming

## How do we choose $n$?

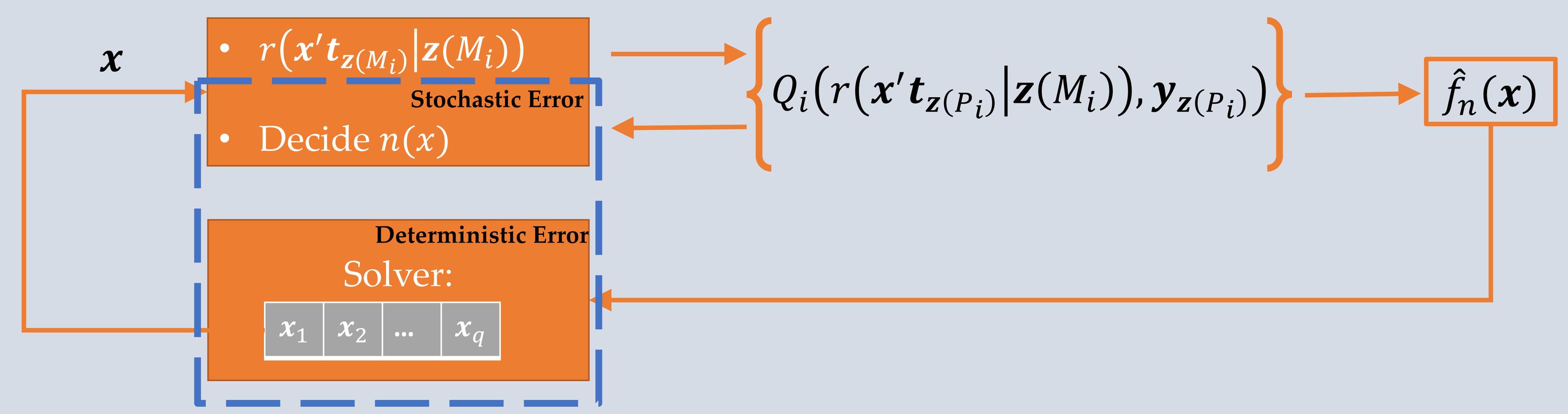
Increase $n$ from lower bound $\ell(k)$ until no further improvements in the upper confidence level

$$\min_n\left\{n \geq \ell(k):\ \hat{f}_n + \alpha_k\frac{\hat{\sigma}_n}{\sqrt{n}} < \hat{f}_{n+1} + \alpha_k\frac{\hat{\sigma}_{n+1}}{\sqrt{n+1}}\right\}$$




## Our Framework




We used these configurations for our numerical experiments:

- Solver : Genetic Algorithm → tuned parameters
- Learning Algorithm r(.) : LM, RF → tuned parameters for RF
- Sampling rule:
  - Lower bound $\ell(k) = \min_n\{3 + \lfloor k/100 \rfloor, 7\}$,
  - CI width $\alpha(k) = t_{n-1,.025}$
- Benchmark: Recursive Feature Elimination,
- Datasets: 2 real data, 1 simulated (#features range (56,221)),
- Performance Measure Q(.): mean squared error.

## Conclusion and Future Work

Novel framework: selecting features with adaptive sampling simulation optimization.

### What's promising:

- good solution for any class of learning algorithms;
- effective sampling rules that adapt to incumbent variability (with bootstraps)
- increased reliability:
  - ✓ more accurate solutions (accuracy),
  - ✓ less variable solutions (sharpness);
- better identifying important features.

### What needs attention:

- large $n$ possibly makes things worse;
- finding structure reduces dimension and makes integer SO solvers applicable;
- combining sampling and solver tailors search to dataset's characteristics, by using info from previously visited solutions.

Winter Simulation Conference 2019
*Simulation for Risk Management*
December 8-11
Gaylord National Resort & Conference Center
National Harbor, Maryland

**Acknowledgements:**

- Seth Guikema (University of Michigan) for initial discussions,
- ISE faculty, Dr. Mayorga and Ivy for their comments,
- ISE friends, Breanna Swan and Amira Hijazi for their insights,
- NC State Graduate Fellowship for funding support.