CEF pre-conference workshop
Meta modelling and budgeted search

Francesco Lamperti

July 03, 2023

Sant'Anna School of Advanced Studies - SSSA
and
RFF-CMCC European Institute on Economics and the Environment - EIEE

The Calibration/Validation Problem

Machine Learning Surrogates

Application to two ABMs

Conclusions

The Calibration/Validation Problem

Agent-Based Models

Given the correct parameters,
Macroeconomic Agent-Based Models (ABMs)
can reproduce realistic economic behaviour (e.g. stylized facts)
and used as test-beds for counterfactual policy exercise.

Two issues and one problem

- **Issue 1: how to say a model produces a realistic output?**
  - Sty facts
  - Method of Simulated Moments (Franke and Westerhoff 2009, 2012; Gilli and Winker 2003)
  - MIC (Barde, 2016), GSL-div (Lamperti, 2017)
  - Causal-relation matching (Guerini and Moneta, forthcoming)
  - Simulated Likelihood (Barunik and Kukacka, 2016)

Two issues and one problem

- **Issue 1: how to say a model produces a realistic output?**
  - Sty facts
  - Method of Simulated Moments (Franke and Westerhoff 2009, 2012; Gilli and Winker 2003)
  - MIC (Barde, 2016), GSL-div (Lamperti, 2017)
  - Causal-relation matching (Guerini and Moneta, forthcoming)
  - Simulated Likelihood (Barunik and Kukacka, 2016)

- **Issue 2: how to find the correct parameters?**
  - Gradient-based methods (Recchioni et al, 2015)
  - Evolutionary algorithms (Platt and Gebbie, 2017)
  - Grid-search
  - MCMC/Rejection sampling (Grazzini et al, 2017)

Two issues and one problem

- **Issue 1: how to say a model produces a realistic output?**
    - Sty facts
    - Method of Simulated Moments (Franke and Westerhoff 2009, 2012; Gilli and Winker 2003)
    - MIC (Barde, 2016), GSL-div (Lamperti, 2017)
    - Causal-relation matching (Guerini and Moneta, forthcoming)
    - Simulated Likelihood (Barunik and Kukacka, 2016)

- **Issue 2: how to find the correct parameters?**
    - Gradient-based methods (Recchioni et al, 2015)
    - Evolutionary algorithms (Platt and Gebbie, 2017)
    - Grid-search
    - MCMC/Rejection sampling (Grazzini et al, 2017)

- ... **and one problem**

Realistic ABMs are computationally prohibitive to simulate extensively...

...and parameter spaces are often really large!

Inferential procedure	Sampling	Simulations (# of periods)	Total time (days)	Qualitative assessment
Non-parametric KDE	Grid exploration	10,000	37	Very good precision, small bias
Non-parametric KDE	MCMC	400,000	1,911	Poor precision
Parametric Gaussian likelihood	Grid exploration	400,000	800	Very good precision
Parametric Gaussian likelihood	MCMC	400,000	800	Poor precision
ABC	Rejection sampling	400,000	800	Good precision

Table 1: Performance of different Bayesian techniques. Running one simulation of 1,500 periods (trading day) requires 7.2 secs on our reference machine. Performing KDE requires 6.2 secs per simulation. Gaussian density estimation and ABC require practically no additional costs.

Source: Grazzini et al , 2017.

Agent-Based Model Calibration

Parameters $\rightarrow$ ABM Evaluation

Agent-Based Model Calibration

Parameters $\rightarrow$ ABM Evaluation $\rightarrow$ Calibration Test(s)

Agent-Based Model Calibration

Parameters $\rightarrow$ ABM Evaluation $\rightarrow$ Calibration Test(s) $\rightarrow$ Pass/Fail

Agent-Based Model Calibration: Costs

$$\underbrace{\text{Parameters}}_{\text{Cheap}} \rightarrow \underbrace{\text{ABM Evaluation}}_{\text{Expensive}} \rightarrow \underbrace{\text{Calibration Test(s)}}_{\text{Relatively Cheap}} \rightarrow \underbrace{\text{Pass/Fail}}_{\text{Cheap}}$$

ABM Evaluations are Prohibitive

Use a cheaper approximation
(with acceptable accuracy).

Machine Learning Surrogates

Surrogate Model

$$\underbrace{\text{Parameters}}_{\text{Unlabelled Data}} \rightarrow \underbrace{\text{ABM Evaluation} \rightarrow \text{Calibration Test} \rightarrow \text{Pass/Fail}}_{\text{Surrogate Model}}$$

Surrogates in Economic/Financial ABMs

- **Kriging**
  - Salle and Yildizoglu, 2013; Dosi et al, 2017; Barglígli et al, 2017; Barde 2022.

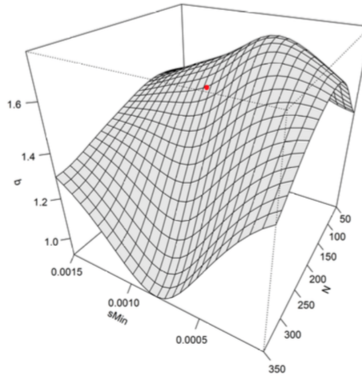
Optimal interpolation based on Gaussian regression against observed values from surrounding data points, weighted according to spatial covariance values.

Kriging estimator: $Z^*(u) - m(u) = \sum_{j=1}^{J} \lambda_j [Z(u_j) - m(u_j)]$ where

- u, u_j are the location of estimation and surrounding locations
- J is the total number of surrounding points
- m is the mean surface value

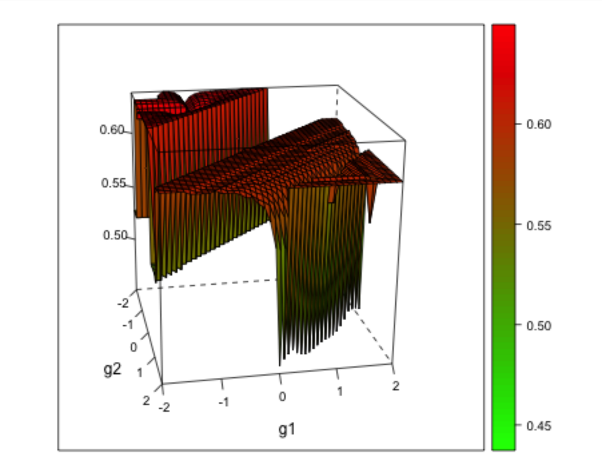
Goal is to find λ_j,
which ends up to depend of the covariance function among data points, which usually depends on their distance!

Kriging



(b) 3D response surface (Laplace)

Example of real response surface



From the analysis of the Brock Hommes asset pricing model in Lamperti (2018, JEIC)

Issues with Kriging

1. Response surfaces are very smooth, difficult to identify abrupt changes
2. Difficulties in handling large parameter spaces
3. Usually learned over few points
4. Usually tested (out of sample) over few points

Issues with Kriging

1. Response surfaces are very smooth, difficult to identify abrupt changes
2. Difficulties in handling large parameter spaces
3. Usually learned over few points
4. Usually tested (out of sample) over few points

Our proposal (Lamperti et al. 2018, JEDC)

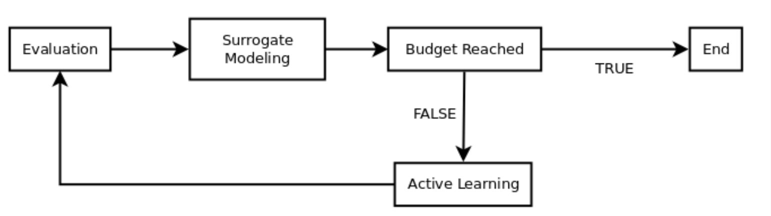
- Build a surrogate model
- Intelligently selecting learning points
- Allowing to capture abrupt "regime shifts"
- Efficiently and accurately approximating the true ABM

Our ML Surrogate

It comes from a semi-supervised active learning approach,

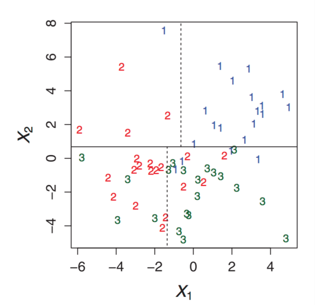
where the surrogate function is obtained through

- Extremely Gradient Boosted Regression and Classification Trees (XGBoost), allowing to obtain our
- **Budgeted Online Active Surrogate Modeling approach (BOAM).**



CARTS and Boosting

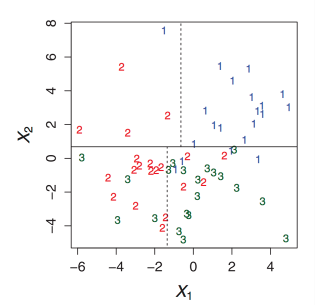
In a classification problem, we have a training sample of n observations on a class variable Y that takes values 1, 2, ... , k, and p predictor variables, X1,...,Xp. Our goal is to find a model for predicting the values of Y from new X values



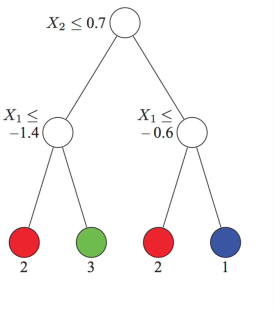
State space partitions.

CARTS and Boosting

In a classification problem, we have a training sample of n observations on a class variable Y that takes values 1, 2, ... , k, and p predictor variables, X1,...,Xp. Our goal is to find a model for predicting the values of Y from new X values



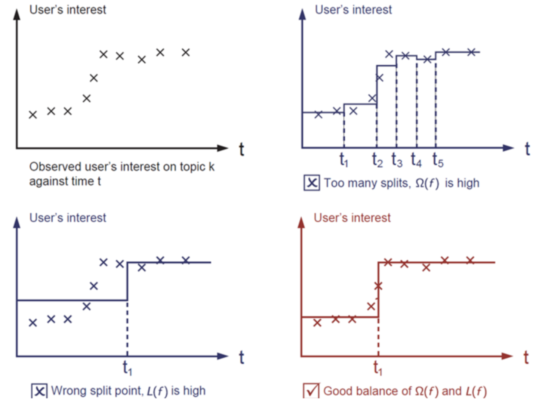
State space partitions.



Corresponding decision tree.

CARTS and Boosting

- **CARTS can be optimized**
  - Objective: $Obj(\Theta) = L(\theta) + \Omega(\Theta)$
  - L is the loss function: e.g. $L(\theta) = \sum(y_i - \hat{y}_i)^2$
  - Ω is the regularization term, which controls for model complexity



Carts and Boosting

- **CARTS can be boosted**
  - Usually, a single tree is not strong enough to be used in practice. What is actually used is the so-called **tree ensemble model**, which sums the prediction of multiple trees together.

Carts and Boosting

- **CARTS can be boosted**
  - Usually, a single tree is not strong enough to be used in practice. What is actually used is the so-called **tree ensemble model**, which sums the prediction of multiple trees together.
  - what we need to learn are functions (f_t), each containing the structure of the tree and the leaf scores

Carts and Boosting

- **CARTS can be boosted**
  - Usually, a single tree is not strong enough to be used in practice. What is actually used is the so-called **tree ensemble model**, which sums the prediction of multiple trees together.
  - what we need to learn are functions (f_t), each containing the structure of the tree and the leaf scores
- **Boosting CARTS**

$$
\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\dots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned}
$$

$$
\begin{aligned}
\text{obj}^{(t)} &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \Omega(f_i) \\
&= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + constant
\end{aligned}
$$

Carts and Boosting

- f_t can be written as: $f_t(x) = w_{q(x)},\ s.t.\ \ w \in R^T,\ \ q : R^d \rightarrow \{1, 2, ..., T\}$
  - w is the score assigned to each leaf
  - q a function mapping each data point to the corresponding leaf

Carts and Boosting

- f_t can be written as: $f_t(x) = w_{q(x)},\ s.t.\ \ w \in R^T,\ \ q : R^d \rightarrow \{1, 2, ..., T\}$
  - w is the score assigned to each leaf
  - q a function mapping each data point to the corresponding leaf
- in XGBoost: $\Omega(f) = \gamma T + 0.5\lambda \sum_{j=1}^{T} w_j^2$
- Therefore...

Carts and Boosting

- f_t can be written as: $f_t(x) = w_{q(x)},\ s.t.\ w \in R^T,\ \ q : R^d \rightarrow \{1, 2, ..., T\}$
  - w is the score assigned to each leaf
  - q a function mapping each data point to the corresponding leaf

- in XGBoost: $\Omega(f) = \gamma T + 0.5\lambda \sum_{j=1}^{T} w_j^2$
- Therefore...

- $Obj \approx \sum_i [g_i w_{q_i} + 0.5 h_i w_{q_i}^2] + \gamma T + 0.5\lambda \sum_j w_j^2$
  - g_i is the gradient of the loss function
  - h_i is an element of the hessian of the loss function
- **this is the score of a given tree structure, which can be optimized (recursively)**

Our algorithm

Set:

- Agent Based Model: $ABM(\cdot)$
- Sampling distribution $\nu \in \mathcal{R}^J$
- Calibration function $C(\cdot)$
- Learning algorithm $\mathcal{A}$, with parameters Θ
- Evaluation budget B
- Initial training set size $N \ll B$
- $X^{Training} \in \mathbb{R}^{N \times J}$
- Calibration labels $Y^{Training} \in \mathbb{N}^N$ binary outcome case (at least 1 positive calibration)
- Calibration labels $Y^{Training} \in \mathbb{R}^N$ real-valued outcome case (at least 1 positive calibration)
- Hyper-parameter optimization algorithm (HPO)

Initialize:

- Per-round sampling size $S \ll B$
- Per-round out-of-sample size $K \gg B$

While $|Y| < B$, **repeat**

1. $\mathit{Surrogate} = \mathrm{HPO}(\mathcal{A}(\Theta, X^{\mathit{Training}}, Y^{\mathit{Training}}))$
2. Draw out-of-sample points $X^{OOS} \in \mathbb{R}^{K \times J} \sim \nu$
3. Select $X^{sample} \in \mathbb{R}^{S \times J}$ from X^{OOS}
4. Evaluate X^{sample} through the Surrogate
5. Compare to $Y^{sample} = \{C(\mathrm{ABM}(X_i^{sample}))\}_{i=1\ldots S}$
6. Identify True Positives (X_{TP}^{sample})
7. Set $X^{\mathit{Training}} = X^{\mathit{Training}} \cup X_{TP}^{sample}$)

end while

Application to two ABMs

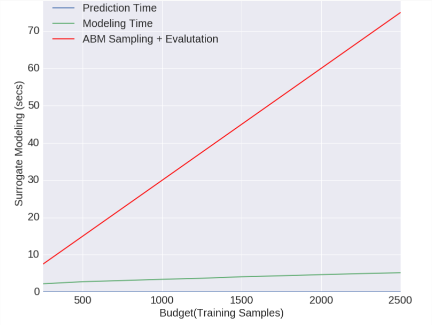
Example Calibration Results

Brock and Hommes Model

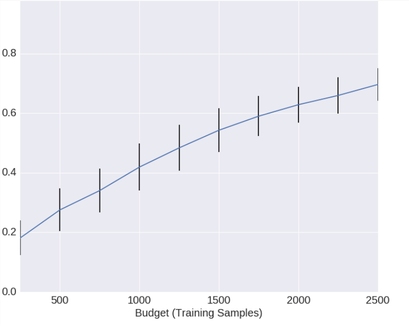
Parameter	Description	Support	Range
β	intensity of choice	$[0; +\infty)$	$[0.0; 10.0]$
n_1	initial share of type 1 traders	$[0; 1]$	0.5
b_1	bias of type 1 traders	$(-\infty; +\infty)$	$[-2.0; 2.0]$
b_2	bias of type 2 traders	$(-\infty; +\infty)$	$[-2.0; 2.0]$
g_1	trend component of type 1 traders	$(-\infty; +\infty)$	$[-2.0; 2.0]$
g_2	trend component of type 2 traders	$(-\infty; +\infty)$	$[-2.0; 2.0]$
C	cost of obtaining type 1 forecasts	$[0; +\infty)$	$[0.0; 5.0]$
ω	weight to past profits	$[0.0, 1.0]$	$[0.0; 1.0]$
σ	asset volatility	$(0; +\infty)$	$(0.0; 1.0]$
ν	attitude towards risk	$[0; +\infty]$	$[0; 100]$
r	risk-free return	$(1; +\infty)$	$[1.01, 1.1]$
T_{BH}	number of periods	$\mathcal{N}$	500

Brock and Hommes model Performance (Single RNG Seed)

Time

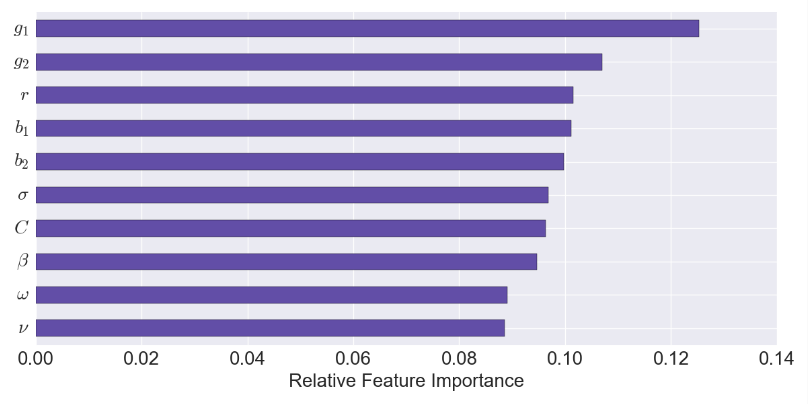


True positive rate



XGBoost Surrogate with unlabelled pool of 10K pts and 100 runs.

Feature importance

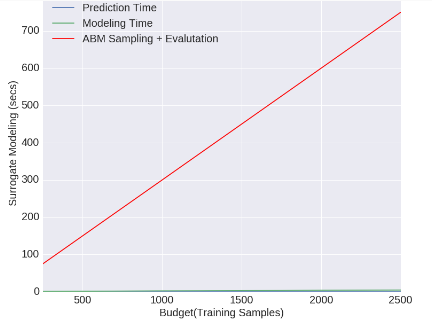


Islands ABM Model

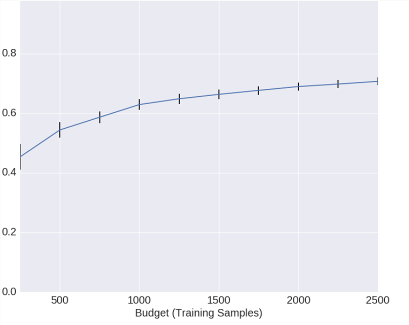
Parameter	Description	Support	Range
ρ	degree of locality in the diffusion of knowledge	$[0, +\infty)$	$[0; 10]$
λ	mean of Poisson r.v. - jumps in technology	$[0; +\infty)$	1
α	productivity of labour in extraction	$[0, +\infty)$	$[0.8; 2]$
φ	cumulative learning effect	$[0, 1]$	$[0.0; 1.0]$
π	probability of finding a new island	$[0.0, 1.0]$	$[0.0; 1.0]$
ϵ	willingness to explore	$[0, 1]$	$[0.0; 1.0]$
m_0	initial number of agents in each island	$[2, +\infty)$	50
T_{IS}	number of periods	$\mathcal{N}$	1000

Island model Performance (Single RNG Seed)

Time

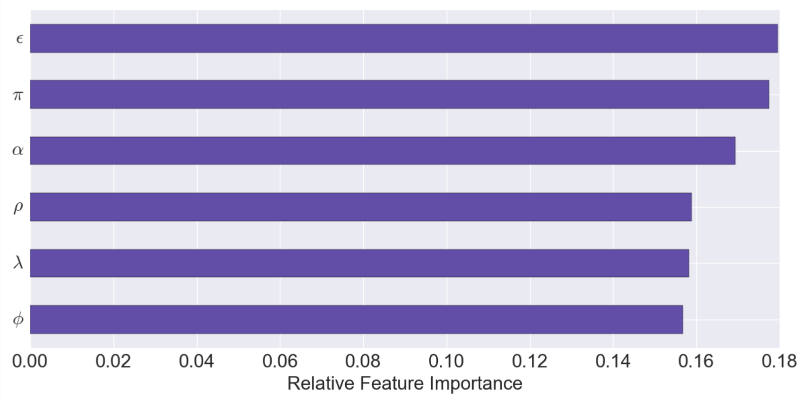


True positive rate



XGBoost Surrogate with unlabelled pool of 10K pts and 100 runs.

Feature importance



Full Monte-Carlo Results

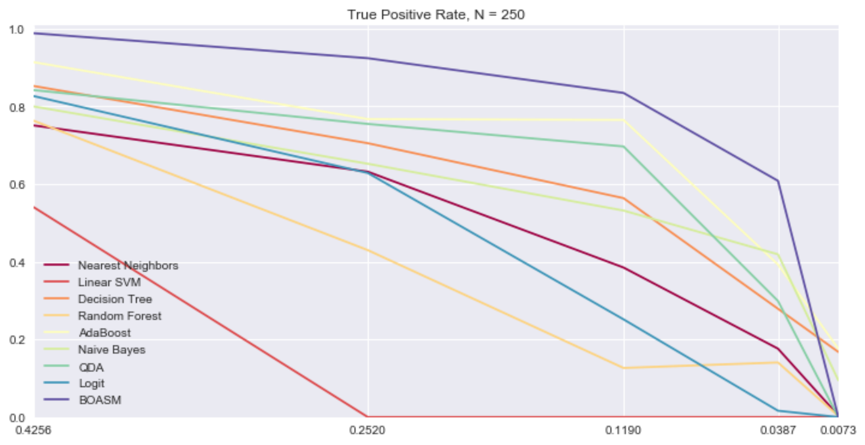
Surrogate Algorithm	False Positives	True Positives	Precision
Logit	22	355	94.17%
BOAMS	2	305	99.35%

Islands ABM Model with 100 MC evaluations on unlabelled pool of 100K points.

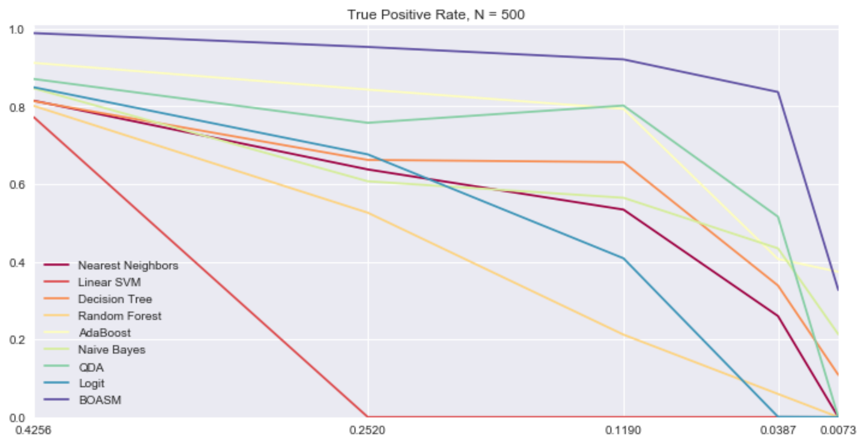
What about other surrogates?

- Gaussian Processes (Kriging)
- Logit
- Support Vector Machines
- Random Forest
- Deep Neural Networks (Universal Approximator)
- Ensembling
- ...

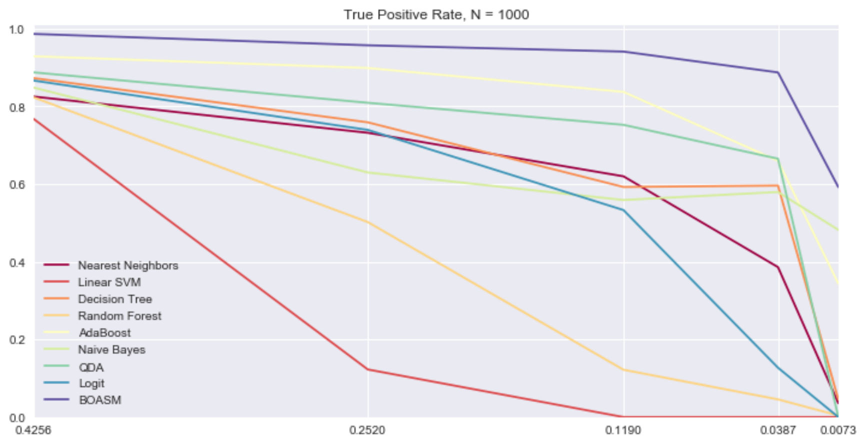
True Positive Rate vs. Positive Density



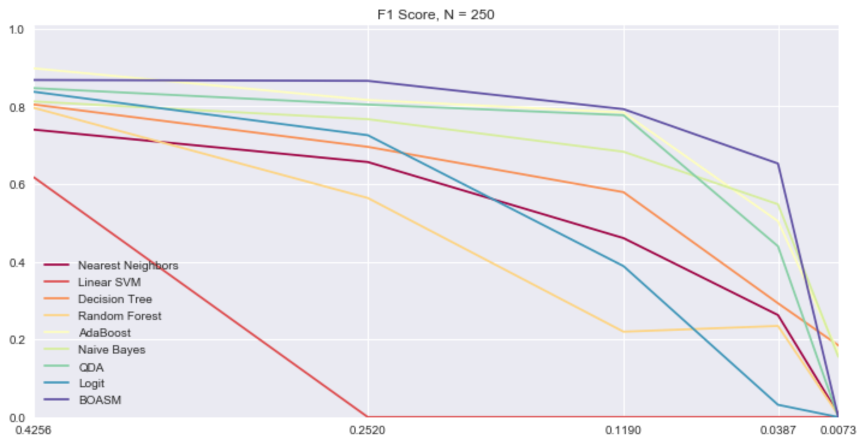
True Positive Rate vs. Positive Density



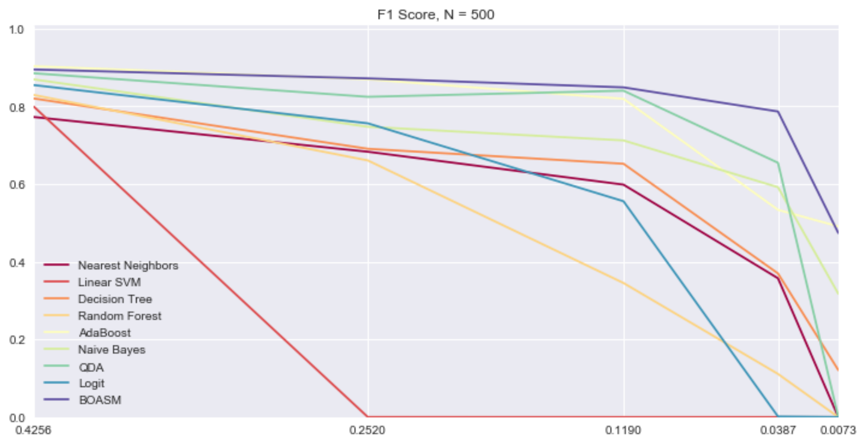
True Positive Rate vs. Positive Density



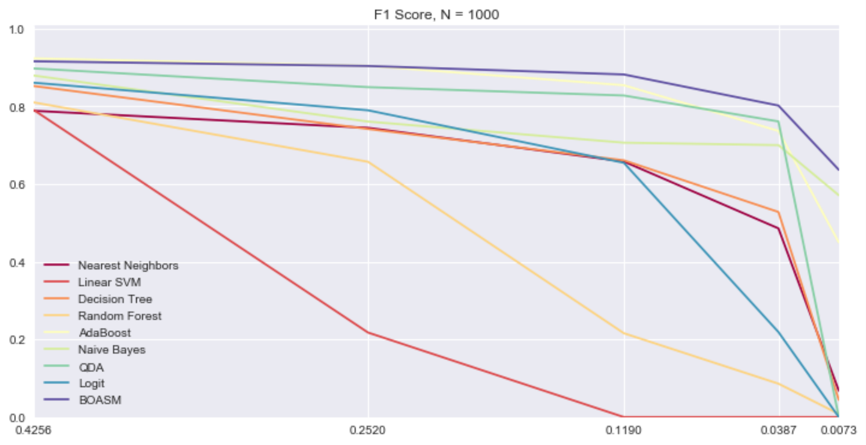
F1-Score vs. Positive Density



F1-Score vs. Positive Density



F1-Score vs. Positive Density



Conclusions

Thank you!