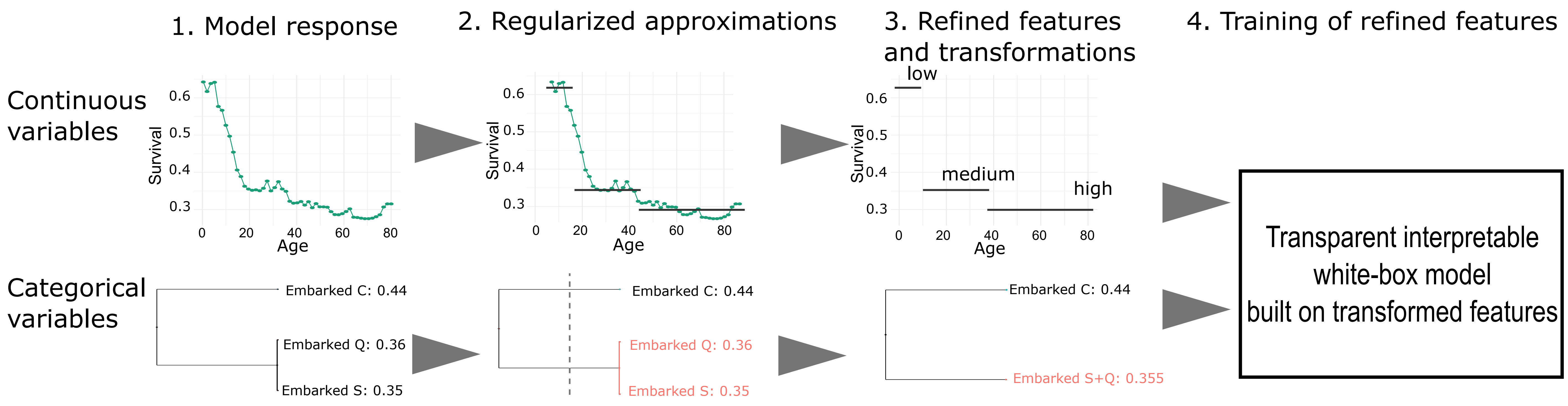
SAFE ML: Surrogate Assisted Feature Extraction for Automated Model Training

Alicja Gosiewska[1], Przemyslaw Biecek[1,2]
[1]Faculty of Matchematics and Information Science, Warsaw University of Technology
[2]Faculty of Mathematics, Informatics, and Mechanics, University of Warsaw


1. Model response
2. Regularized approximations
3. Refined features and transformations
4. Training of refined features

Continuous variables

Categorical variables

Transparent interpretable white-box model built on transformed features

Figure 1: The SAFE ML algorithm in four steps, 1. train an elastic surrogate model, 2. approximate model response, 3. extract transformations and new features, 4. train refined model.

Introduction

Complex black-box predictive models may have high accuracy, but opacity causes problems such as lack of trust, lack of stability, or sensitivity to concept drift. On the other hand, interpretable models require more work related to feature engineering, which is time consuming. **Can we train interpretable and accurate models without timeless feature engineering?**

SAFE ML is a method that extracts knowledge from elastic black-boxes (surrogate models) to perform feature engineering and create simpler and interpretable glass-box models (refined models). This approach overcomes the trade-off between interpretability and accuracy of the model. Results of benchmarks shows that with SAFE ML we can often create interpretable, yet still accurate models.

Formal formulation of the problem

Let us consider a data generating process that creates the data (X, Y), where X is a matrix of n rows (observations) and p columns (independent variables) and Y is a potentially stochastic vector of n response values.

We consider X as a subspace $X \subseteq \mathbb{R}^p$ and X_i for $i = 1, 2, ..., p$ as variables. We can think of sets of variables X_i as aspects that can be coded into new set of features. An aspect can code one variable but also may include multiple variables. As X'_j we denote coded aspect, where X'_j is a subset of $\mathbb{R}^{q_j}$ for some $q_j \in \mathbb{N}$. Let us denote $x \in X$ and $x'_j \in X'_j$ as vectors form the relevant spaces.

Also, let $f : X \to \mathbb{R}$ be **a black-box model**. As function f represents a potentially complex model, our goal is to obtain a simple model train on the basis of knowledge gained from f. To accomplish this, we use relationships between variables and model response to create transformations of these variables. Transformed variables may be then used to train new, simple model.

Now, we can define **transformer functions** $h^f_j(x)$. Let $h^f_j(x) = x'_{K_j}$, where $h^f_j : X \to X'_{K_j}$ be a transformer function from space X into space $X_{K'_j}$.

Let X' be a cartesian product of sets X'_{K_j}: $X' = X'_{K_1} \times X'_{K_2} \times ... \times X'_{K_J}$. **Feature transformation function** $h^f : X \to X'$ is:

$$h^f(x) = (h^f_1(x), h^f_2(x), ..., h^f_J(x)).$$

Let us note that h^f_i could be defined on a subset of X since h^f_i do not have to include all p variables. However, we set the domain of functions h^f_i on X to keep the notation simpler.

We define a glass-box model $g : x' \in X' \to y \in \mathbb{R}$ and $g \in G$ where G is a class of interpretable models. Let H be a defined class of transformations.

The best glass-box model is obtained by the following formulation:

$$g = \arg\min_{g \in G} \min_{h^f \in H} \mathcal{L}(g(h^f(x)), y).$$

Where $\mathcal{L}$ is some loss function, for example, accuracy, cross-entropy, or root mean square error.

SAFE ML as data-driven feature transformation

Classic feature engineering is based on some class of variable transformations, such as logarithm, sinus, or root square. SAFE ML approach is data-driven, this means that feature transformation functions are heavily dependent on data and surrogate model. New binary features are created on the basis of surrogate predictions.

Let us now consider transformation functions $h^{f,SAFE}_i : X \to \{0, 1\}^{q_i}$, such as $h^{f,SAFE}_i$ transforms values of the i-th variable into binary vectors of length q_i.

- If X_i is a **categorical variable**, function $h^{f,SAFE}_i$ merges some of levels of x_i and code them as a single binary variable. In SAFE ML, we use hierarchical clustering to find new concatenated levels.
- If X_i is a **numerical variable**, function $h^{f,SAFE}_i$ binns x_i, for example, due to the changepoints [3] analysis of partial dependence profile [2]. See an example in Figure 2.

A result of $h^{f,SAFE}_i$ transformation is a space of new interpretable binary representations of variables from a space X.

Benchmark

We have benchmarked SAFE ML method on 30 classification data sets from OpenML100 [4] collection. In Table 1, we present results for 2 selected data sets. We compare performances of naïve logistic regressions, surrogate xgboosts, and refined logistic regressions. Here **naïve regression** means that we fit vanilla regression model without any feature engineering and we consider it as a baseline. **Refined logistic regressions** are models trained on featured engineered with SAFE ML and with different penalties for changepoint. All outcomes of benchmark are available as a web application (see Figure 3).

Data set	AUC						
	Naïve	Ref 2	Ref 4	Ref 6	Ref 8	Ref 10	Surrogate
pc3	0.450	0.772	0.779	0.775	0.781	0.783	0.830
kc2	0.710	0.817	0.818	0.808	0.802	0.802	0.818

Table 1: Performances of models trained on pc2 and kc2 data sets for classification. Headers indicate naïve model, refined models created with SAFE ML with penalties 2, 4, 6, 8, 10, and surrogate xgboost model. Values of AUC are averaged over results of 10-fold cross-validation. Refined models perform better than naïve logistic regression.

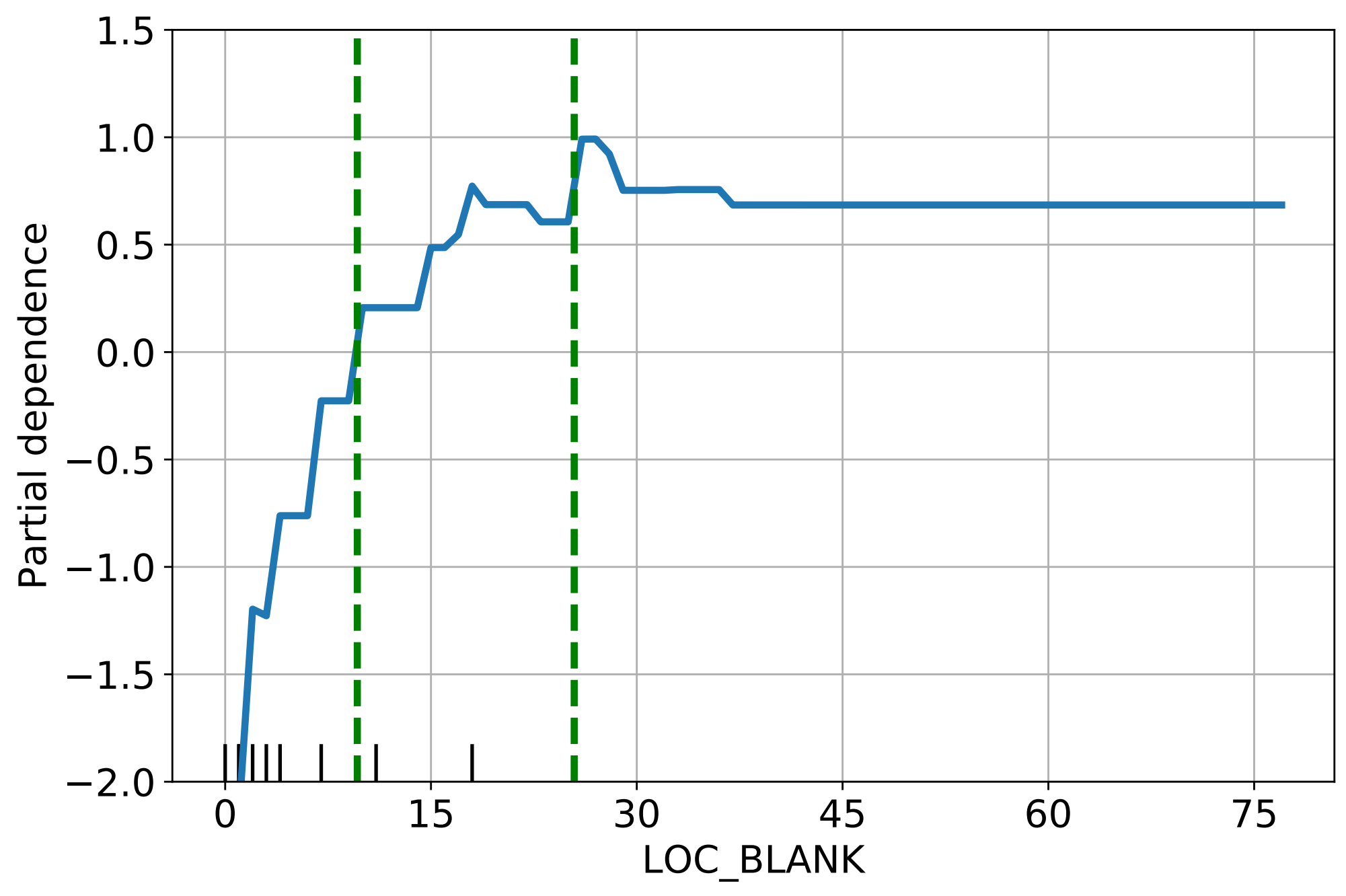


Figure 2: Example SAFE transformation of variable with penalty equals 2. Blue line is pratial dependence profile of LOC_BLANK variable from pc3 data set based on surrogated xgboost model. Dashed green lines mark bins indicated by SAFE ML.

Figure 3: The Shiny application with results of benchmark.

Conclusions

SAFE ML algorithm uses surrogate model to perform feature transformations which are then used to train refined glass-box model. Substitution of black-boxes by interpretable models increase transparency and trust in predictions. What is more, simple models, such as, linear regression and logistic regression are extensively described from a mathematical point of view and there are many tools to diagnose them.

The results of benchmark shows that with SAFE ML it is possible to build an accurate interpretable model in a semi-automatic manner.

Software and code

Benchmarks were generated with SafeTransformer Python library available at `https://github.com/ModelOriented/SAFE`. There is also an R implementation of SAFE method available at `https://github.com/MI2DataLab/SAFE`.

References

[1] Alicja Gosiewska, Aleksandra Gacek, Piotr Lubon, and Przemyslaw Biecek. SAFE ML: Surrogate Assisted Feature Extraction for Model Learning. 2019. URL `https://arxiv.org/abs/1902.11035`.

[2] Brandon M. Greenwell. pdp: An R Package for Constructing Partial Dependence Plots. *The R Journal*, 2017.

[3] Charles Truong, Laurent Oudre, and Nicolas Vayatis. A review of change point detection methods. 2018. URL `http://arxiv.org/abs/1801.00718`.

[4] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: Networked science in machine learning. *SIGKDD Explorations*, 2013.

Acknowledgements

Alicja Gosiewska was financially supported by the grant of Polish Centre for Research and Development POIR.01.01.01-00-0328/17. Przemyslaw Biecek was financially supported by the grant NCN Opus grant 2017/27/B/ST6/01307.

Contact

- `mi2.mini.pw.edu.pl`
- ✉ `alicjagosiewska@gmail.com` `gosiewska.com`