ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
HAROKOPIO UNIVERSITY

.dit/*
Informatics & Telematics

Regional effect plots for the interpretation of black box machine learning models

Χρήστος Δίου / Christos Diou

Department of Informatics and Telematics
Harokopio University of Athens

Statistics Seminars 2023-2024
Department of Statistics, Athens University of Economics and Business
19/04/2024

Contents

Hypothetical (?) scenarios

- The computer vision subsystem of an autonomous vehicle leads the vehicle to take a left turn, in front of a car moving in the opposite direction[1]

[1] https://www.theguardian.com/technology/2022/dec/22/tesla-crash-full-self-driving-mode-san-francisco

[2] https://www.technologyreview.com/2021/06/17/1026519/racial-bias-noisy-data-credit-scores-mortgage-loans-fairness-machine-learning/

[3] https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Hypothetical (?) scenarios

- The computer vision subsystem of an autonomous vehicle leads the vehicle to take a left turn, in front of a car moving in the opposite direction[1]
- The credit assessment system leads to the rejection of an application for a loan - the client suspects racial bias[2]

[1] https://www.theguardian.com/technology/2022/dec/22/tesla-crash-full-self-driving-mode-san-francisco

[2] https://www.technologyreview.com/2021/06/17/1026519/racial-bias-noisy-data-credit-scores-mortgage-loans-fairness-machine-learning/

[3] https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Hypothetical (?) scenarios

- The computer vision subsystem of an autonomous vehicle leads the vehicle to take a left turn, in front of a car moving in the opposite direction[1]
- The credit assessment system leads to the rejection of an application for a loan - the client suspects racial bias[2]
- A model that assesses the risk of future criminal offenses (and used for decisions on parole sentences) is biased against black prisoners[3]

[1] https://www.theguardian.com/technology/2022/dec/22/tesla-crash-full-self-driving-mode-san-francisco

[2] https://www.technologyreview.com/2021/06/17/1026519/racial-bias-noisy-data-credit-scores-mortgage-loans-fairness-machine-learning/

[3] https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Questions

- Why did a model make a specific decision?
- What could we change so that the model will make a different decision?
- Can we summarize and predict the model's behavior?

Today we focus on the last question

Taxonomy of interpretability methods

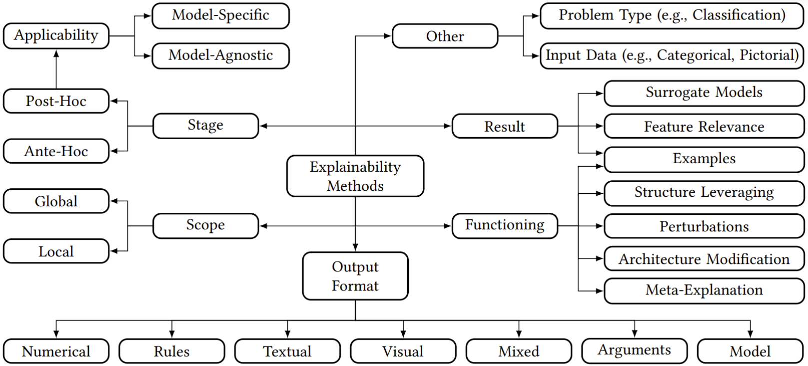


Figure: Timo Speith, “A Review of Taxonomies of Explainable Artificial Intelligence (XAI) Methods”. In 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT ’22), 2022 (Speith, 2022)

Contents

Interpretable models (ante-hoc)

- Some models afford explanations
  - interpretable-by-design
- Examples, (generalized) linear models, decision trees, k-NN
- Example: Linear regression

$$\hat{y} = w_1 x_1 + \ldots + w_p x_p + b$$

Interpretable models (ante-hoc)

- Result in the bike sharing dataset (model weights)

$$\hat{y} = w_1 x_1 + \ldots + w_p x_p + b$$

	Weight	SE	|t|
(Intercept)	2399.4	238.3	10.1
seasonSPRING	899.3	122.3	7.4
seasonSUMMER	138.2	161.7	0.9
seasonFALL	425.6	110.8	3.8
holidayHOLIDAY	-686.1	203.3	3.4
workingdayWORKING DAY	124.9	73.3	1.7
weathersitMISTY	-379.4	87.6	4.3
weathersitRAIN/SNOW/STORM	-1901.5	223.6	8.5
temp	110.7	7.0	15.7
hum	-17.4	3.2	5.5
windspeed	-42.5	6.9	6.2
days_since_2011	4.9	0.2	28.5

Figure: C. Molnar, IML book, 2022 (Molnar, 2022)

Interpretable models (ante-hoc)

- Feature effects (visualization)

$$effect_j^{(i)} = w_j x_j^{(i)}$$

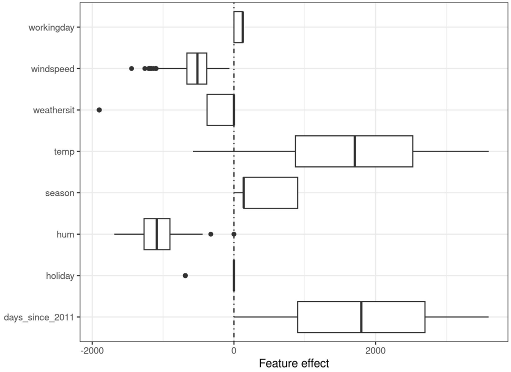


Figure: C. Molnar, IML book, 2022 Molnar, 2022

Feature effect methods (1)

- Black-box model $f(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$, trained on $\mathcal{D}$
- Goal:
  - For single variable: Plot illustrating the effect of a feature x_s on f for all values of x_s
  - For pairs of variables: Plot illustrating the effect of pair (x_s, x_l) on f for all values of x_s and x_l

Feature Effect: global, model-agnostic, outputs plot

Feature Effect methods (2)

$y = f(x_s) \rightarrow$ plot showing the effect of x_s on the output y

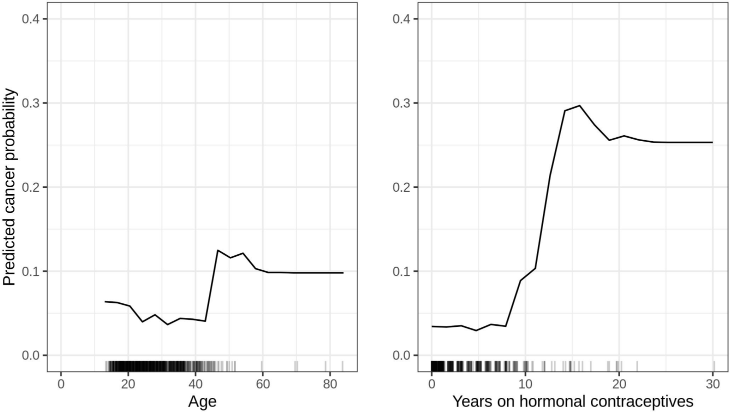


Figure: C. Molnar, IML book, 2022 (Molnar, 2022)

Feature Effect is simple and intuitive.

Feature Effect Methods (3)

- $x_s \rightarrow$ feature of interest, $\boldsymbol{x_c} \rightarrow$ other features
- How can we isolate x_s?
- Difficult task:
  - features are correlated
  - f has learned complex interactions

PDP, MPlot and ALE

- PDP (Friedman, 2001)
  - $f(x_s) = \mathbb{E}_{\boldsymbol{x_c}}[f(x_s, \boldsymbol{x_c})]$
  - **Unrealistic instances**
  - e.g. $f(x_{\texttt{age}} = 20, x_{\texttt{years\_contraceptives}} = 20) = ??$

PDP, MPlot and ALE

- PDP (Friedman, 2001)
  - $f(x_s) = \mathbb{E}_{\boldsymbol{x_c}}[f(x_s, \boldsymbol{x_c})]$
  - **Unrealistic instances**
  - e.g. $f(x_{\texttt{age}} = 20, x_{\texttt{years\_contraceptives}} = 20) = ??$

- MPlot (Apley and Zhu, 2020)
  - $\boldsymbol{x_c}|x_s$: $f(x_s) = \mathbb{E}_{\boldsymbol{x_c}|x_s}[f(x_s, \boldsymbol{x_c})]$
  - **Aggregated effects**
  - Real effect: $x_{\texttt{age}} = 50 \rightarrow 10$, $x_{\texttt{years\_contraceptives}} = 20 \rightarrow 10$
  - MPlot may assign 17 to both

PDP, MPlot and ALE

- PDP (Friedman, 2001)
  - $f(x_s) = \mathbb{E}_{\boldsymbol{x_c}}[f(x_s, \boldsymbol{x_c})]$
  - **Unrealistic instances**
  - e.g. $f(x_{\texttt{age}} = 20, x_{\texttt{years\_contraceptives}} = 20) = ??$

- MPlot (Apley and Zhu, 2020)
  - $\boldsymbol{x_c}|x_s$: $f(x_s) = \mathbb{E}_{\boldsymbol{x_c}|x_s}[f(x_s, \boldsymbol{x_c})]$
  - **Aggregated effects**
  - Real effect: $x_{\texttt{age}} = 50 \rightarrow 10$, $x_{\texttt{years\_contraceptives}} = 20 \rightarrow 10$
  - MPlot may assign 17 to both

- ALE (Apley and Zhu, 2020)
  - $f(x_s) = \int_{x_{min}}^{x_s} \mathbb{E}_{\boldsymbol{x_c}|z}[\frac{\partial f}{\partial x_s}(z, \boldsymbol{x_c})]\partial z$
  - **Resolves both failure modes**

ALE approximation

ALE definition: $f(x_s) = \int_{x_{s,min}}^{x_s} \mathbb{E}_{\boldsymbol{x_c}|z}[\frac{\partial f}{\partial x_s}(z, \boldsymbol{x_c})]\partial z$

ALE approximation: $f(x_s) = \sum_k^{k_x} \underbrace{\frac{1}{|\mathcal{S}_k|} \sum_{i:\boldsymbol{x}^i \in \mathcal{S}_k} \underbrace{[f(z_k, \boldsymbol{x_c^i}) - f(z_{k-1}, \boldsymbol{x_c^i})]}_{\text{point effect}}}_{\text{bin effect}}$

ALE approximation

ALE approximation: $f(x_s) = \sum_k^{k_x} \underbrace{\frac{1}{|\mathcal{S}_k|} \sum_{i:\boldsymbol{x}^i \in \mathcal{S}_k} \underbrace{\left[f(z_k, \boldsymbol{x}_{\boldsymbol{c}}^{\boldsymbol{i}}) - f(z_{k-1}, \boldsymbol{x}_{\boldsymbol{c}}^{\boldsymbol{i}})\right]}_{\text{point effect}}}_{\text{bin effect}}$

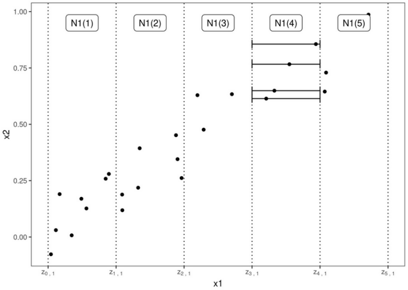


Figure: Image taken from Interpretable ML book (Molnar, 2022)

Bin splitting (parameter K) is crucial!

ALE approximation - weaknesses

$$f(x_s) = \sum_{k}^{k_x} \underbrace{\frac{1}{|\mathcal{S}_k|} \sum_{i:\boldsymbol{x}^i \in \mathcal{S}_k} \underbrace{[f(z_k, \boldsymbol{x_c^i}) - f(z_{k-1}, \boldsymbol{x_c^i})]}_{\text{point effect}}}_{\text{bin effect}}$$

- Point Effect $\Rightarrow$ evaluation at bin limits
  - 2 evaluations of f per point $\rightarrow$ slow
  - change bin limits, pay again $2 * N$ evaluations of f $\rightarrow$ restrictive
  - broad bins may create out of distribution (OOD) samples $\rightarrow$ not-robust in wide bins

Contents

V. Gkolemis, T. Dalamagas and C. Diou, “DALE: Differential Accumulated Local Effects for efficient and accurate global explanations”, ACML 2022 (Gkolemis, Dalamagas, and Diou, 2023)

Work in collaboration with Vasilis Gkolemis (PhD student @ HUA) and Theodoros Dalamagas (Researcher, ATHENA RC)

Our proposal: Differential ALE

$$f(x_s) = \Delta x \sum_{k}^{k_x} \underbrace{\frac{1}{|\mathcal{S}_k|} \sum_{i:\boldsymbol{x}^i \in \mathcal{S}_k} \underbrace{[\frac{\partial f}{\partial x_s}(x_s^i, \boldsymbol{x_c^i})]}_{\text{point effect}}}_{\text{bin effect}}$$

- Point Effect $\Rightarrow$ evaluation on instances
  - Fast $\rightarrow$ use of auto-differentiation, all derivatives in a single pass
  - Versatile $\rightarrow$ point effects computed once, change bins without cost
  - Secure $\rightarrow$ does not create artificial instances
  - Unbiased estimator of ALE (bias / variance proofs in the paper and supporting material)

For differentiable models, DALE resolves ALE weaknesses

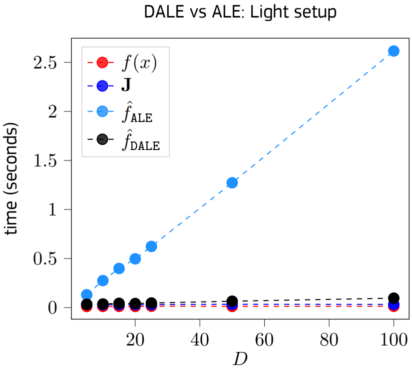
DALE is faster and more versatile - theory

$$f(x_s) = \Delta x \sum_{k}^{k_x} \underbrace{\frac{1}{|\mathcal{S}_k|} \sum_{i:\boldsymbol{x}^i \in \mathcal{S}_k} \underbrace{[\frac{\partial f}{\partial x_s}(x_s^i, \boldsymbol{x_c^i})]}_{\text{point effect}}}_{\text{bin effect}}$$

- Faster
  - gradients wrt all features $\nabla_{\boldsymbol{x}} f(\boldsymbol{x^i})$ in a single pass (via the Jacobian)
  - auto-differentiation must be available (deep learning)
- Versatile
  - Change bin limits, with near zero computational cost

DALE is faster and allows redefintion of the bin limits

DALE is faster and versatile - Experiments



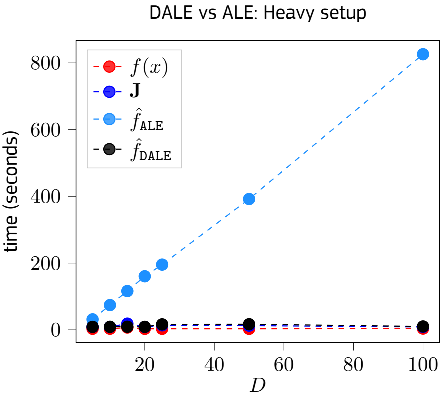


Figure: Light setup; small dataset $(N = 10^2$ instances), computationally light f. Heavy setup; big dataset ($N = 10^5$ instances), computationally heavy f. D is the number of dimensions.

DALE considerably accelerates the estimation

DALE uses on-distribution samples - Theory

$$f(x_s) = \underbrace{\sum_k^{k_x} \frac{1}{|\mathcal{S}_k|} \sum_{i:\boldsymbol{x}^i \in \mathcal{S}_k} \underbrace{[\frac{\partial f}{\partial x_s}(x_s^i, \boldsymbol{x_c^i})]}_{\text{point effect}}}_{\text{bin effect}}$$

- point effect independent of bin limits
  - $\frac{\partial f}{\partial x_s}(x_s^i, \boldsymbol{x_c^i})$ computed on real instances $\boldsymbol{x}^i = (x_s^i, \boldsymbol{x_c^i})$
- bin limits affect only the resolution of the plot
  - wide bins $\rightarrow$ low resolution plot, bin estimation from more points
  - narrow bins $\rightarrow$ high resolution plot, bin estimation from less points

DALE enables wide bins without creating out of distribution instances

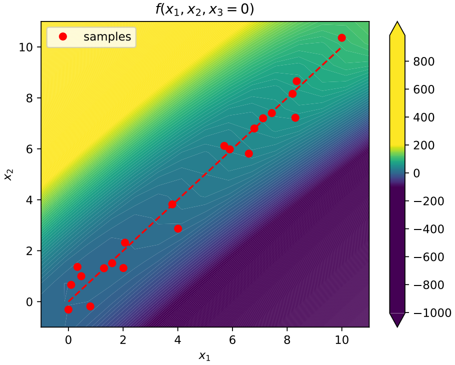
DALE uses on-distribution samples - Experiments

$$f(x_1, x_2, x_3) = x_1 x_2 + x_1 x_3 \pm g(x)$$

$$x_1 \in [0, 10], x_2 \sim x_1 + \epsilon, x_3 \sim \mathcal{N}(0, \sigma^2)$$

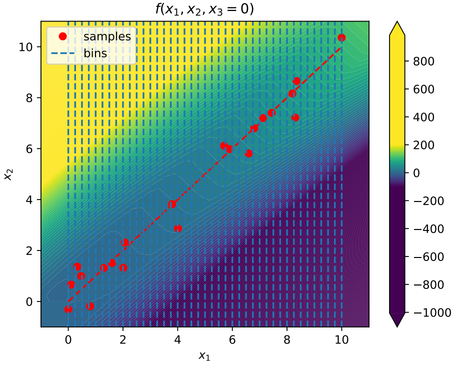
$$f_{\texttt{ALE}}(x_1) = \frac{x_1^2}{2}$$

- point effects affected by $(x_1 x_3)$ (σ is large)
- bin estimation is noisy (samples are few)



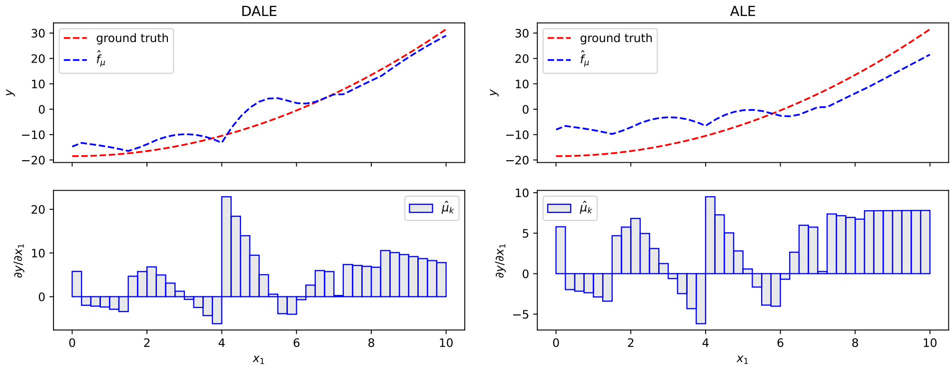
Intuition: we need wider bins (more samples per bin)

DALE vs ALE - 40 Bins



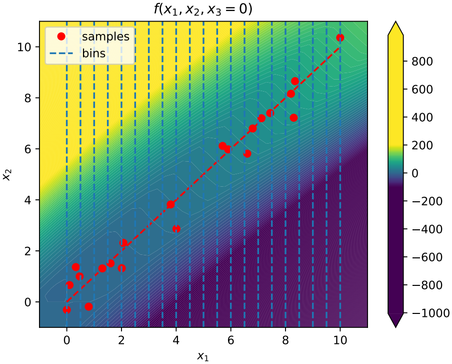
- DALE: on-distribution, noisy bin effect → poor estimation
- ALE: on-distribution, noisy bin effect → poor estimation

DALE vs ALE - 40 Bins



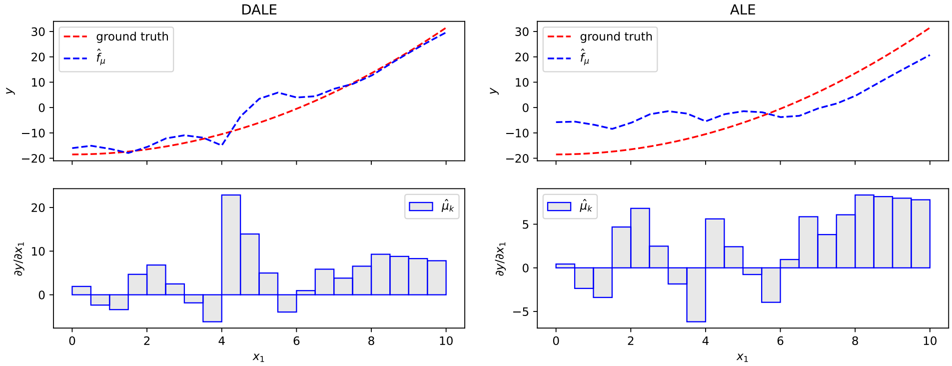
- DALE: on-distribution, noisy bin effect → poor estimation
- ALE: on-distribution, noisy bin effect → poor estimation

DALE vs ALE - 20 Bins



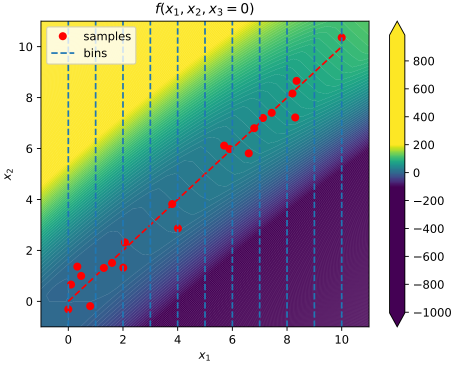
- DALE: on-distribution, noisy bin effect $\rightarrow$ poor estimation
- ALE: on-distribution, noisy bin effect $\rightarrow$ poor estimation

DALE vs ALE - 20 Bins



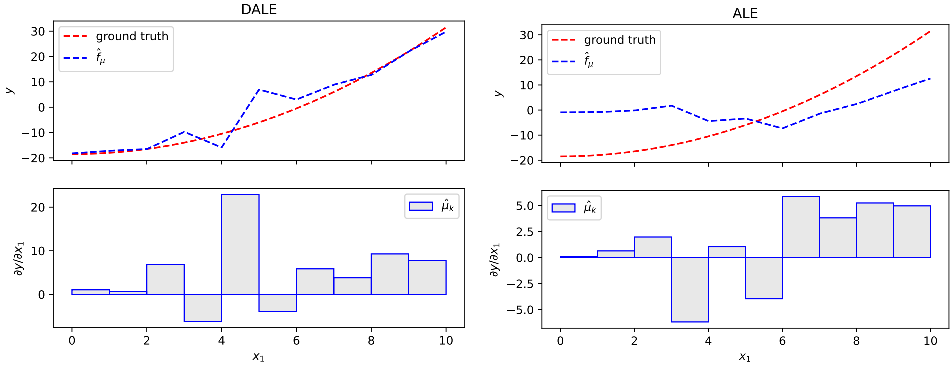
- DALE: on-distribution, noisy bin effect $\rightarrow$ poor estimation
- ALE: on-distribution, noisy bin effect $\rightarrow$ poor estimation

DALE vs ALE - 10 Bins



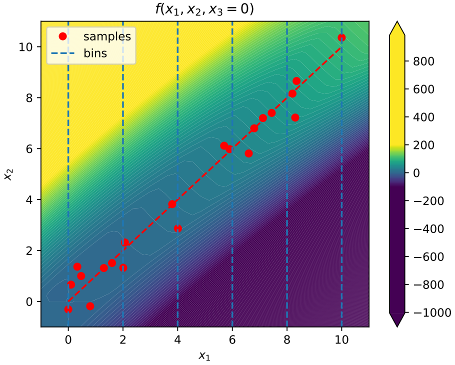
- DALE: on-distribution, noisy bin effect $\rightarrow$ poor estimation
- ALE: starts being OOD, noisy bin effect $\rightarrow$ poor estimation

DALE vs ALE - 10 Bins



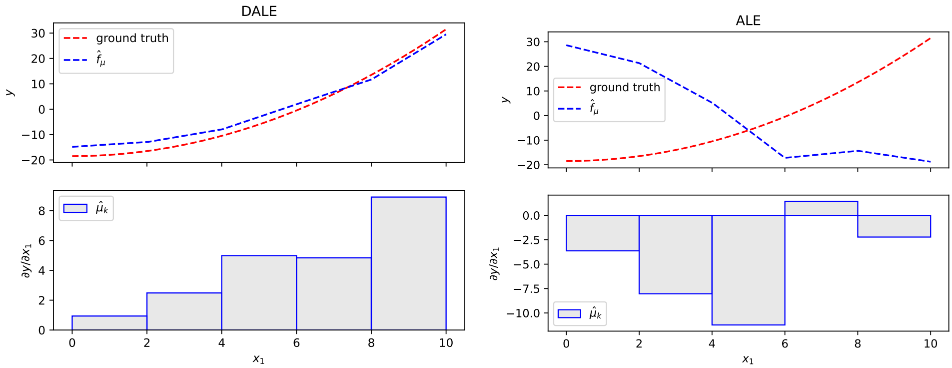
- DALE: on-distribution, noisy bin effect $\rightarrow$ poor estimation
- ALE: starts being OOD, noisy bin effect $\rightarrow$ poor estimation

DALE vs ALE - 5 Bins



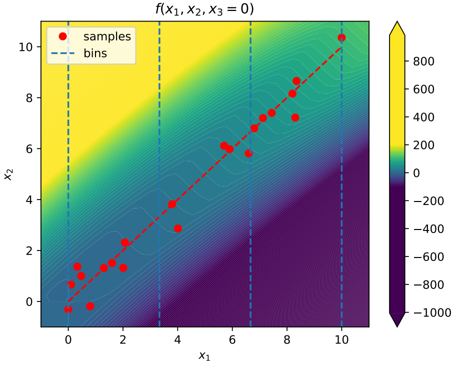
- DALE: on-distribution, robust bin effect $\rightarrow$ good estimation
- ALE: completely OOD, robust bin effect $\rightarrow$ poor estimation

DALE vs ALE - 5 Bins



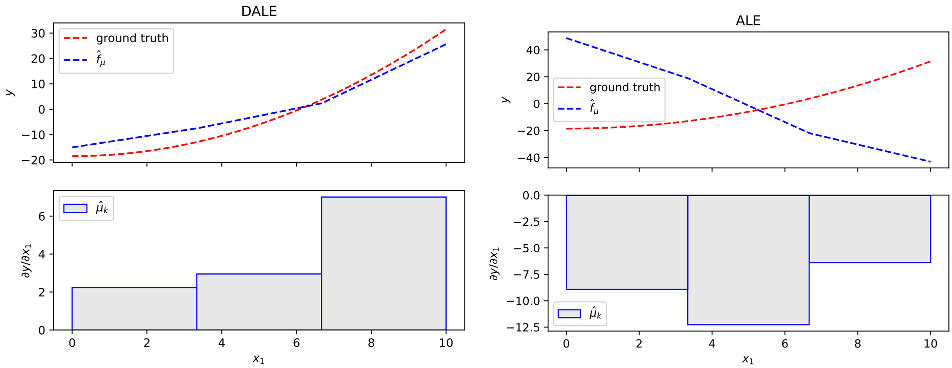
- DALE: on-distribution, robust bin effect $\rightarrow$ good estimation
- ALE: completely OOD, robust bin effect $\rightarrow$ poor estimation

DALE vs ALE - 3 Bins



- DALE: on-distribution, robust bin effect $\rightarrow$ good estimation
- ALE: completely OOD, robust bin effect $\rightarrow$ poor estimation

DALE vs ALE - 3 Bins



- DALE: on-distribution, robust bin effect → good estimation
- ALE: completely OOD, robust bin effect → poor estimation

Real Dataset Experiments - Efficiency

- Bike-sharing dataset (Fanaee-T and Gama, 2013)
- $y \rightarrow$ daily bike rentals
- $\mathbf{x}$: 10 features, most of them characteristics of the weather

Efficiency on Bike-Sharing Dataset (Execution Times in seconds)

	Number of Features										
	1	2	3	4	5	6	7	8	9	10	11
DALE	1.17	**1.19**	**1.22**	**1.24**	**1.27**	**1.30**	**1.36**	**1.32**	**1.33**	**1.37**	**1.39**
ALE	**0.85**	1.78	2.69	3.66	4.64	5.64	6.85	7.73	8.86	9.9	10.9

DALE requires almost same time for all features

Real Dataset Experiments - Accuracy

- Difficult to compare in real world datasets
- We do not know the ground-truth effect
- In most features, DALE and ALE agree.
- Only $X_{\texttt{hour}}$ is an interesting feature

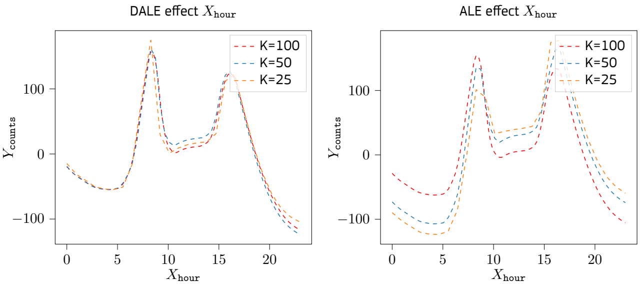


Figure: (Left) DALE (Left) and ALE (Right) plots for $K = \{25, 50, 100\}$

Contents

V. Gkolemis, T. Dalamagas, E. Ntoutsi and C. Diou, “RHALE: Robust and Heterogeneity-aware Accumulated Local Effects ”, ECAI 2023 (Gkolemis, Dalamagas, Ntoutsi, et al., 2023)

Work in collaboration with Vasilis Gkolemis (PhD student @ HUA), Theodoros Dalamagas (Researcher, ATHENA RC) and Eirini Ntoutsi (Prof, Universität der Bundeswehr, München)

Next step: Heterogeneity and optimal bin selection

Using DALE, one has the computational margin to worry about additional issues:

- Computation of heterogeneity of local effects (i.e., standard error of the mean)
- Optimal selection of bins such that the effect does not have a high variation within the bin

RHALE: Robust and Heterogeneity-aware Accumulated Local Effects

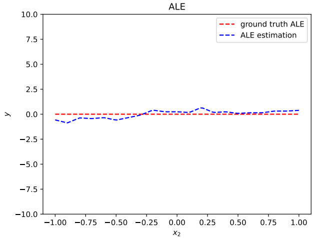
- Robust: Automatic bin splitting (result does not depend on arbitrary bin selection)
- Heterogeneity aware: $\pm$ from the average

Example (based on Goldstein et al., 2015)

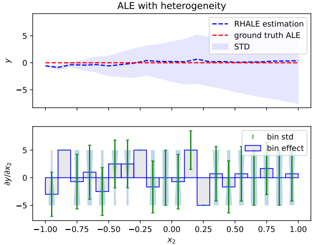
Aggregation bias

$$Y = 0.2X_1 - 5X_2 + 10X_2\mathbb{1}_{X_3>0} + \mathcal{E}$$

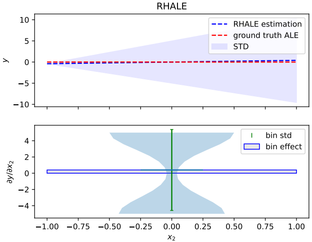
$$\mathcal{E} \overset{i.i.d.}{\sim} \mathcal{N}(0,1), \quad X_1, X_2, X_3 \overset{i.i.d.}{\sim} \mathcal{U}(-1,1)$$



x_2 ALE plot (20 bins)



x_2 ALE + heterogeneity (20 bins)



x_2 RHALE (auto-binning)

Definitions and Approximations - Main effect

ALE main effect definition

$$f^{\texttt{ALE}}(x_s) = \int_{x_{s,\min}}^{x_s} \underbrace{\mathbb{E}_{X_c|X_s=z}\left[f^s(z, X_c)\right]}_{\mu(z)} \partial z$$

ALE main effect approximation

$$\hat{f}^{\texttt{ALE}}(x_s) = \Delta x \sum_{k}^{k_x} \underbrace{\frac{1}{|\mathcal{S}_k|} \sum_{i:\boldsymbol{x}^i \in \mathcal{S}_k} [\frac{\partial f}{\partial x_s}(x_s^i, \boldsymbol{x_c^i})]}_{\text{bin effect:}\hat{\mu}(z)}$$

Simple but wrong: ALE + Heterogeneity

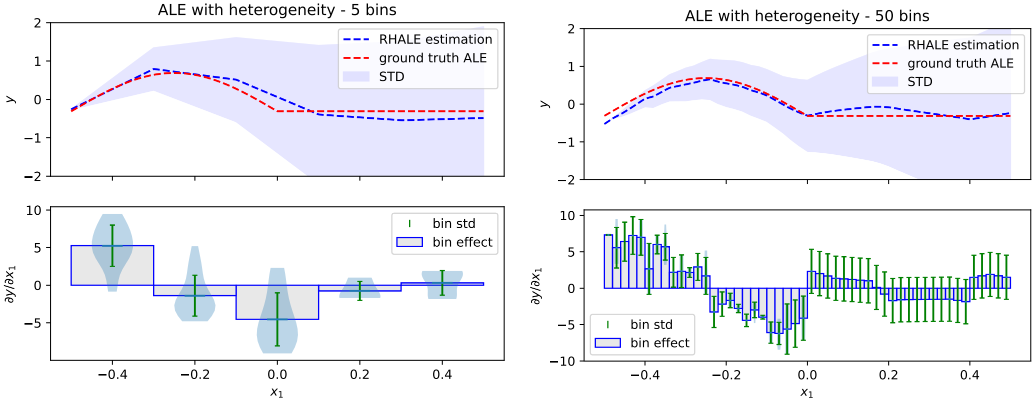


Figure: Left: approximation with narrow bin-splitting (5 bins) and (Right) with dense-bin splitting

- Fixed-size bin splitting can ruin the estimation of the heterogeneity

Definitions and Approximations - Heterogeneity

ALE heterogeneity definition

$$\sigma(x_s) = \sqrt{\int_{x_{s,\min}}^{x_s} \underbrace{\mathbb{E}_{X_c|X_s=z}\left[\left(f^s(z, X_c) - \mu(z)\right)^2\right] \partial z}_{\sigma^2(z)}}$$

ALE heterogeneity approximation

$$\mathtt{STD}(x_s) = \sqrt{\sum_{k=1}^{k_x} (z_k - z_{k-1})^2 \underbrace{\frac{1}{|\mathcal{S}_k| - 1} \sum_{i:\mathbf{x}^i \in \mathcal{S}_k} \left(f^s(\mathbf{x}^i) - \hat{\mu}(z_{k-1}, z_k)\right)^2}_{\hat{\sigma^2}(z)}}$$

Derivations

In the paper we formally prove

1. the conditions under which the above definition is an unbiase estimator of the heterogeneity
2. the conditions under which a bin splitting minimizes the estimator variance

Based on the above, we formulate bin-splitting as an optimization problem and propose an efficient solution using dynamic programming.

RHALE: Robust and Heterogeneity-aware ALE

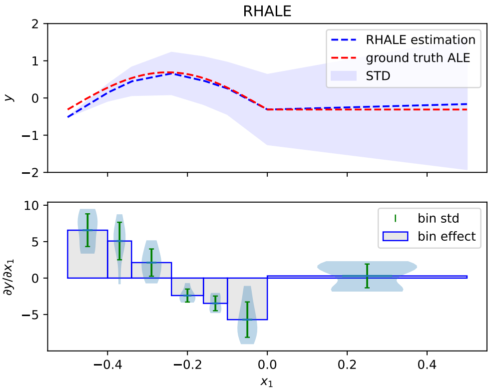


Figure: Variable bin size leads to improved estimation

Simple but correct:

- Automatically finds the **optimal** bin-splitting
- Optimal $\Rightarrow$ best approximation of the average (ALE) effect
- Optimal $\Rightarrow$ best approximation of the heterogeneity

Impact

In case you work with a differentiable model, as in Deep Learning, use the combination of DALE and RHALE to:

- compute ALE fast, for multiple bin sizes in one pass
- quantify the heterogeneity of the ALE plot, i.e., the deviation of the instance-level effects from the average effect
- get a robust approximation of (a) the main ALE effect and (b) the heterogeneity, using automatic bin-splitting

Contents

`Effector` - A python package for global and regional feature effects

V. Gkolemis, C. Diou, V. Gkolemis, C. Diou, E. Ntoutsi, T. Dalamagas, B. Bischl, J. Herbinger, G. Casalicchio, "Effector: A Python package for regional explanations", arXiv preprint arXiv:2404.02629, 2024

https://xai-effector.github.io

Installation, for python version 3.7+:

```
pip install effector
```

Regional effects

- Similar to the way one can select optimal bin splits to minimize heterogeneity, one can also identify optimal subregions of the features $\boldsymbol{x}_c$ where the effect is homogeneous

Regional effect plots - Process

- Combines two methods:
  - RHALE
  - Regional effects (Herbinger, Bischl, and Casalicchio, 2023)
- Idea:
  - Feature effect is the average effect of each feature x_s on the output y
  - It is computed by averaging the instance-level effects
  - Heterogeneity $\mathcal{H}$ measures the deviation of the instance-level effects from the average effect due to feature interactions
  - Split the dataset in subgroups in order to minimize the heterogeneity
- Concretely:

$$\underbrace{\mathcal{H}(f_i(x_i))}_{\mathcal{H}\text{ before split}} >> \underbrace{\mathcal{H}(f_i(x_i|x_j > \tau)) + \mathcal{H}(f_i(x_i|x_j \leq \tau))}_{\text{sum of }\mathcal{H}\text{ after split}}$$

Regional effect plots - Objective

$$
\begin{aligned}
\underset{\{\mathcal{R}_{st}\}_{t=1}^{T_s}}{\text{minimize}} \quad & \mathcal{L}_s = \sum_{t=1}^{T_s} \frac{|\mathcal{D}_{st}|}{|\mathcal{D}|} H_{st}^{\mathtt{m}} \\
\text{subject to} \quad & \bigcup_{t=1}^{T} \mathcal{R}_{st} = \mathcal{X}_c \\
& \mathcal{R}_{st} \cap \mathcal{R}_{s\tau} = \emptyset, \quad \forall t \neq \tau
\end{aligned} \tag{1}
$$

Algorithm

Algorithm 1: Detect subspaces

Input : Heterogeneity function H_s, Maximum depth L

Output: subspaces $\{\mathcal{R}_{st}\}_{t=1}^{T_s}$, where $T_s \in \{0, 2, \dots, 2^L\}$

H_{s0} ; `// Compute the level of interactions before any split`
$D = \{(\mathbf{x}^i, y^i)\}_{i=1}^N$; `// Initial dataset`
$T_s = 0$; `// Initialize the number of splits for feature` s
for $l = 1$ *to* L **do**
  if $H_s^{l-1} = 0$ **then**
    break ; `// Stop if the heterogeneity is zero`
  end
  `/* Iterate over all features` $\mathbf{x}_c$ `and candidate split positions` p `*/`
  `/* Find the optimal split with heterogeneity` $H_s^l = \sum_{t=1}^{2^l} \frac{|\mathcal{D}_{st}|}{|\mathcal{D}|} H_{st}$ `*/`
  `/* Define the subspaces` $\{\mathcal{R}_{st}\}_{t=1}^{2^l}$ `and the datasets` $\{\mathcal{D}_{st}\}_{t=1}^{2^l}$ `*/`
  if $\frac{H_s^l}{H_s^{l-1}} < \epsilon$ **then**
    break ; `// Stop, if heterogeneity drop is small` $(< \epsilon)$
  end
  $T_s = 2^l$; `// Update the number of splits for feature` s
end
return $\{\mathcal{R}_{st} | s \in \{1, \dots, D\}, t \in \{1, \dots, T_s\}\}$

Effector - Implemented methods

Method	Equation	Formula
$\hat{f}^{\texttt{RHALE}}(x_s)$	Eq. (4)	$\sum_{k=1}^{k_{x_s}} \frac{z_k - z_{k-1}}{\|\mathcal{S}_k\|} \sum_{i:\mathbf{x}^i \in \mathcal{S}_k} \frac{\partial f}{\partial x_s}(x_s^i, \mathbf{x}_c^i)$
$\hat{H}_s^{\texttt{RHALE}}$	Eq. (6)	$\sum_{k=1}^{K_s} \frac{z_k - z_{k-1}}{\|\mathcal{S}_k\|} \sum_{i:\mathbf{x}^i \in \mathcal{S}_k} \left[\frac{\partial f}{\partial x_s}(x_s^i, \mathbf{x}_c^i) - \hat{\mu}_k^{\texttt{RHALE}}\right]^2$
$\hat{f}^{\texttt{ALE}}(x_s)$	Eq. (3)	$\sum_{k=1}^{k_{x_s}} \frac{1}{\|\mathcal{S}_k\|} \sum_{i:\mathbf{x}^i \in \mathcal{S}_k} \left[f(z_k, \mathbf{x}_c^i) - f(z_{k-1}, \mathbf{x}_c^i)\right]$
$\hat{H}_s^{\texttt{ALE}}$	Eq. (5)	$\sum_{k=1}^{K} \frac{1}{\|\mathcal{S}_k\|} \sum_{i:\mathbf{x}^i \in \mathcal{S}_k} \left[(f(z_k, \mathbf{x}_c^i) - f(z_{k-1}, \mathbf{x}_c^i)) - \hat{\mu}_k^{\texttt{ALE}}\right]^2$
$\hat{f}^{\texttt{PDP}}(x_s)$	Eq. (7)	$\frac{1}{N} \sum_{i=1}^{N} f(x_s, \mathbf{x}_c^i)$
$\hat{H}_s^{\texttt{PDP}}$	Eq. (8)	$\frac{1}{T} \sum_{t=1}^{T} \frac{1}{N} \sum_{i=1}^{N} \left[\hat{f}_{i,\text{centered}}^{\texttt{ICE}}(x_{s,t}) - \hat{f}_{\text{centered}}^{\texttt{PDP}}(x_{s,t})\right]^2$
$\hat{f}^{\texttt{d-PDP}}(x_s)$	Eq. (9)	$\frac{1}{N} \sum_{i=1}^{N} f(x_s, \mathbf{x}_c^i)$
$\hat{H}_s^{\texttt{d-PDP}}$	Eq. (10)	$\frac{1}{T} \sum_{t=1}^{T} \frac{1}{N} \sum_{i=1}^{N} \left[\hat{f}_{i,\text{centered}}^{\texttt{ICE}}(x_{s,t}) - \hat{f}_{\text{centered}}^{\texttt{PDP}}(x_{s,t})\right]^2$
$\hat{f}^{\texttt{SHAP-DP}}(x_s)$	Eq. (13)	$\kappa(x_s), \quad \kappa(x_s)$ is a univariate spline fit to $\{(x_s^i, \hat{\phi}_s^i)\}_{i=1}^N$
$\hat{H}_s^{\texttt{SHAP-DP}}$	Eq. (14)	$\frac{1}{N} \sum_{i=1}^{N} \left[\hat{\phi}_s^i - f^{\texttt{SHAP-DP}}(x_s^i)\right]^2$

Effector - Tutorial

Tutorial (Bike Sharing Dataset)
Colab notebook

Recap

- DALE can help with the computation of fast and accurate feature effect explanations for differentiable models
  - One can change the resolution of the explanation (i.e., number of bins K) for free
- RHALE can improve explanations by selecting variable bin splits, in an optimal way
  - Unbiased estimation of heterogeneity
  - Select optimal bin splits to minimize heterogeneity and improve the robustness of the explanation
- `Effector`
  - Implements all popular global effect plot methods
  - Extends these methods to regional effect plots
  - Has very fast implementation, especially for differentiable models (takes advantage of auto-differentiation)

Thank you!

References I

Apley, Daniel W. and Jingyu Zhu (2020). "Visualizing the effects of predictor variables in black box supervised learning models". In: *Journal of the Royal Statistical Society. Series B: Statistical Methodology* 82.4, pp. 1059–1086. ISSN: 14679868. DOI: 10.1111/rssb.12377. arXiv: 1612.08468.

Fanaee-T, Hadi and Joao Gama (2013). "Event labeling combining ensemble detectors and background knowledge". In: *Progress in Artificial Intelligence*, pp. 1–15. ISSN: 2192-6352. DOI: 10.1007/s13748-013-0040-3. URL: [Web%20Link].

Friedman, Jerome H. (2001). "Greedy function approximation: A gradient boosting machine". In: *Annals of Statistics* 29.5, pp. 1189–1232. ISSN: 00905364. DOI: 10.1214/aos/1013203451.

Gkolemis, Vasilis, Theodore Dalamagas, and Christos Diou (2023). "DALE: Differential Accumulated Local Effects for efficient and accurate global explanations". In: *Asian Conference on Machine Learning*. PMLR, pp. 375–390.

Gkolemis, Vasilis, Theodore Dalamagas, Eirini Ntoutsi, et al. (2023). "RHALE: Robust and heterogeneity-aware accumulated local effects". In.

References II

- Goldstein, Alex et al. (2015). "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation". In: *journal of Computational and Graphical Statistics* 24.1, pp. 44–65.
- Herbinger, Julia, Bernd Bischl, and Giuseppe Casalicchio (2023). "Decomposing global feature effects based on feature interactions". In: *arXiv preprint arXiv:2306.00541*.
- Molnar, Christoph (2022). *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2nd ed. URL: https://christophm.github.io/interpretable-ml-book.
- Speith, Timo (2022). "A review of taxonomies of explainable artificial intelligence (XAI) methods". In: *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 2239–2250.