

Paper presentation at ACML 2022

DALE: Differential Accumulated Local Effects for efficient and accurate global explanations

Vasilis Gkolemis^{1,2} Theodore Dalamagas¹ Christos Diou²

¹ATHENA Research and Innovation Center

²Harokopio University of Athens

December 2022

eXplainable AI (XAI)

- Black-box model $f(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$, trained on \mathcal{D}
- XAI extracts interpretable properties:
 - Which features are important (in general)?
 - Which features favor a prediction?
- Categories:
 - Global vs local
 - Model-agnostic vs Model-specific
 - Output? number, plot, instance etc.

Feature Effect: global, model-agnostic, outputs plot

Feature Effect

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

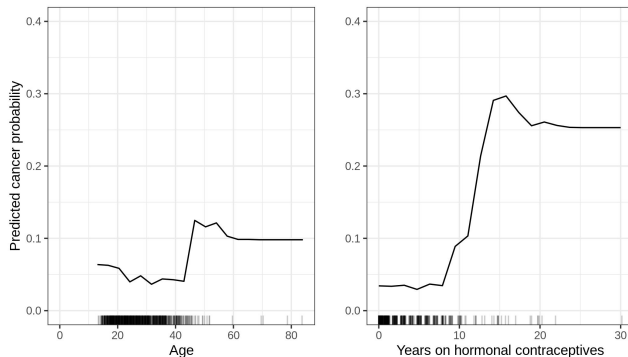


Figure: Image taken from Interpretable ML book [4]

Feature Effect is simple and intuitive.

Feature Effect Methods

- $x_s \rightarrow$ feature of interest, $\mathbf{x}_c \rightarrow$ other features
- FE methods take (f, \mathcal{D}, s) and return $y = f_{\langle \text{name} \rangle}(x_s)$
- PDP[3]
 - ▶ Expected outcome over \mathbf{x}_c : $f(x_s) = \mathbb{E}_{\mathbf{x}_c}[f(x_s, \mathbf{x}_c)]$
 - ▶ **Unrealistic instances**

PDP vs MPlot vs ALE

Feature Effect Methods

- $x_s \rightarrow$ feature of interest, $\mathbf{x}_c \rightarrow$ other features
- FE methods take (f, \mathcal{D}, s) and return $y = f_{\langle \text{name} \rangle}(x_s)$
- PDP[3]
 - ▶ Expected outcome over \mathbf{x}_c : $f(x_s) = \mathbb{E}_{\mathbf{x}_c}[f(x_s, \mathbf{x}_c)]$
 - ▶ **Unrealistic instances**
- MPlot[1]
 - ▶ Expected outcome over $\mathbf{x}_c | x_s$: $f(x_s) = \mathbb{E}_{\mathbf{x}_c | x_s}[f(x_s, \mathbf{x}_c)]$
 - ▶ **Aggregated effects**

PDP vs MPlot vs ALE

Feature Effect Methods

- $x_s \rightarrow$ feature of interest, $\mathbf{x}_c \rightarrow$ other features
- FE methods take (f, \mathcal{D}, s) and return $y = f_{\langle \text{name} \rangle}(x_s)$
- PDP[3]
 - ▶ Expected outcome over \mathbf{x}_c : $f(x_s) = \mathbb{E}_{\mathbf{x}_c}[f(x_s, \mathbf{x}_c)]$
 - ▶ **Unrealistic instances**
- MPlot[1]
 - ▶ Expected outcome over $\mathbf{x}_c | x_s$: $f(x_s) = \mathbb{E}_{\mathbf{x}_c | x_s}[f(x_s, \mathbf{x}_c)]$
 - ▶ **Aggregated effects**
- ALE[1]
 - ▶ $f(x_s) = \int_{x_{\min}}^{x_s} \mathbb{E}_{\mathbf{x}_c | z}[\frac{\partial f}{\partial x_s}(z, \mathbf{x}_c)] \partial z$
 - ▶ **Resolves both failure modes**

PDP vs MPlot vs ALE

ALE approximation

ALE definition: $f(x_s) = \int_{x_{s,min}}^{x_s} \mathbb{E}_{\mathbf{x}_c|z} \left[\frac{\partial f}{\partial x_s}(z, \mathbf{x}_c) \right] \partial z$

ALE approximation from $\mathcal{D} = \{\mathbf{x}^i, y^i\}_{i=1}^N$

$$f(x_s) = \underbrace{\sum_k^{k_x} \frac{1}{|S_k|} \sum_{i:\mathbf{x}^i \in S_k} \underbrace{[f(z_k, \mathbf{x}_c^i) - f(z_{k-1}, \mathbf{x}_c^i)]}_{\text{point effect}}}_{\text{bin effect}}$$

- Partitions interval $[x_{s,min}, x_{s,max}]$ in K equisized bins
- S_k instances lying at k -th bin
- z_{k-1}, z_k limits of k -th bin
- Point effect $\rightarrow [f(z_k, \mathbf{x}_c^i) - f(z_{k-1}, \mathbf{x}_c^i)] \rightarrow$ evaluating at bin limits
- Bin effect \rightarrow mean of point effects
- ALE \rightarrow sum of bin effects until k_x -th bin

Bin splitting (parameter K) is crucial!

ALE approximation

ALE approximation from $\mathcal{D} = \{\mathbf{x}^i, y^i\}_{i=1}^N$

$$f(x_s) = \underbrace{\sum_k \frac{1}{|S_k|} \sum_{i: \mathbf{x}^i \in S_k} \underbrace{[f(z_k, \mathbf{x}_c^i) - f(z_{k-1}, \mathbf{x}_c^i)]}_{\text{point effect}}}_{\text{bin effect}}$$

- 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

ALE approximation has some weaknesses

DALE - Differential ALE

DALE, from the dataset $\mathcal{D} = \{\mathbf{x}^i, y^i\}_{i=1}^N$

$$f(x_s) = \Delta x \underbrace{\sum_k \frac{1}{|\mathcal{S}_k|} \sum_{i: \mathbf{x}^i \in \mathcal{S}_k} \underbrace{\left[\frac{\partial f}{\partial x_s}(\mathbf{x}_s^i, \mathbf{x}_c^i) \right]}_{\text{point effect}}}_{\text{bin effect}}$$

- only change point effect computation
- 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

For **differentiable** models, DALE resolves ALE weaknesses

DALE is faster and versatile - theory

$$f(x_s) = \underbrace{\Delta x \sum_k \frac{1}{|\mathcal{S}_k|} \sum_{i: \mathbf{x}^i \in \mathcal{S}_k}}_{\text{bin effect}} \underbrace{\left[\frac{\partial f}{\partial x_s}(\mathbf{x}_s^i, \mathbf{x}_c^i) \right]}_{\text{point effect}}$$

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

DALE is faster and allows redefining bin-limits

DALE is faster and versatile - Experiments

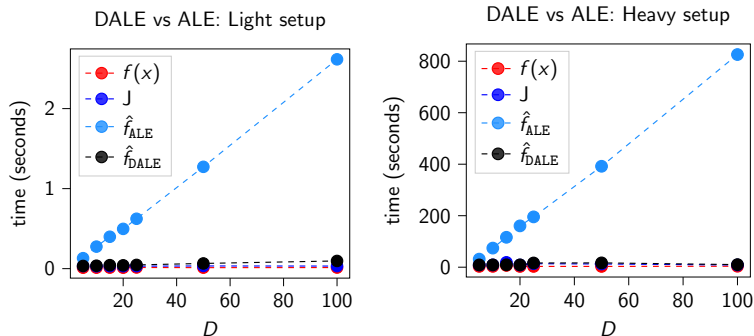


Figure: Light setup; small dataset ($N = 10^2$ instances), light f . Heavy setup; big dataset ($N = 10^5$ instances), heavy f

DALE considerably accelerates the estimation

DALE uses on-distribution samples - Theory

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

- point effect **independent** of bin limits
 - ▶ $\frac{\partial f}{\partial x_s}(\mathbf{x}_s^i, \mathbf{x}_c^i)$ computed on real instances $\mathbf{x}^i = (\mathbf{x}_s^i, \mathbf{x}_c^i)$
- 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

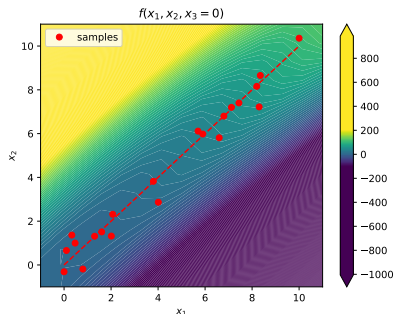
DALE uses on-distribution samples - Experiments

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

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

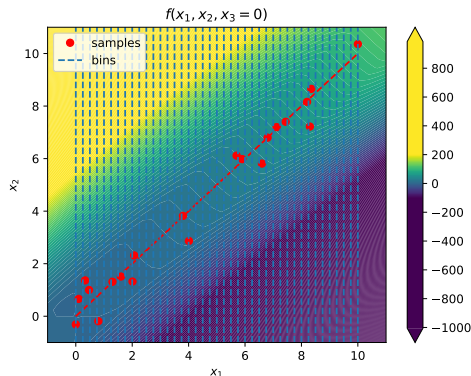
$$f_{\text{ALE}}(x_1) = \frac{x_1^2}{2}$$

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



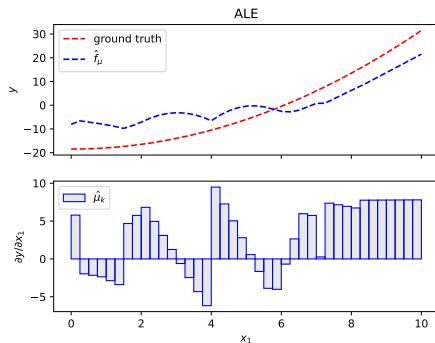
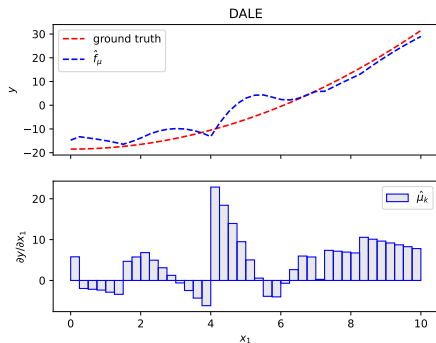
Intuition: we need wider bins (more samples per bin)

DALE vs ALE - 40 Bins



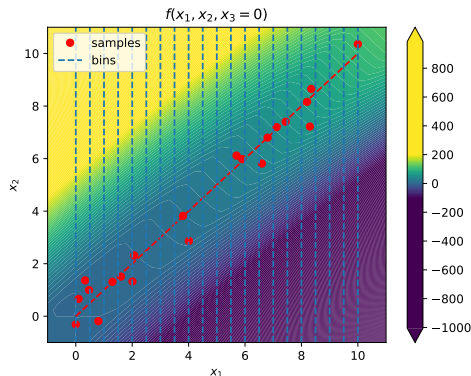
- DALE: on-distribution, noisy bin effect → poor estimation
- ALE: on-distribution, noisy bin effect → poor estimation

DALE vs ALE - 40 Bins



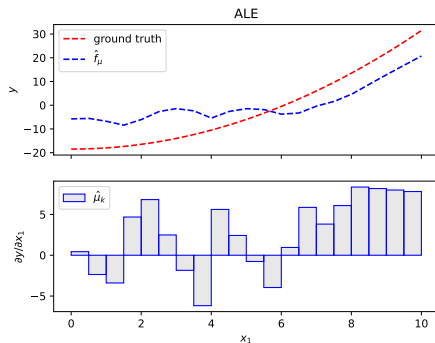
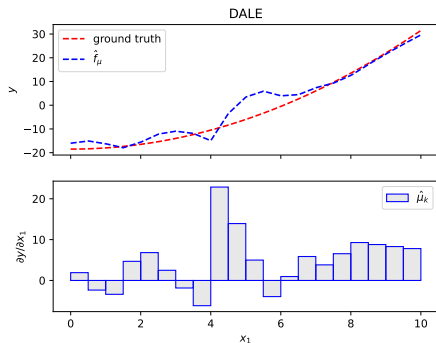
- DALE: on-distribution, noisy bin effect → poor estimation
- ALE: on-distribution, noisy bin effect → poor estimation

DALE vs ALE - 20 Bins



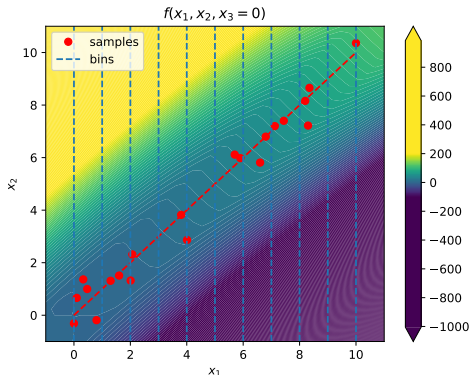
- DALE: on-distribution, noisy bin effect → poor estimation
- ALE: on-distribution, noisy bin effect → poor estimation

DALE vs ALE - 20 Bins



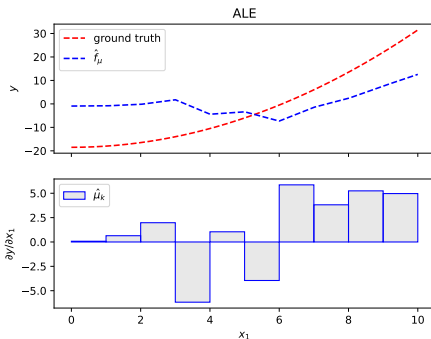
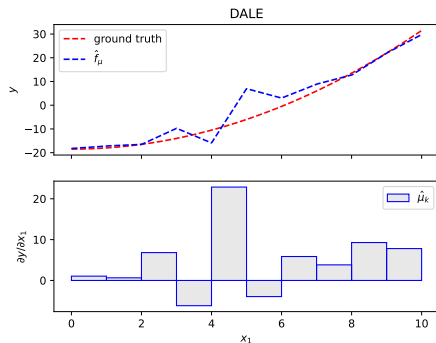
- DALE: on-distribution, noisy bin effect → poor estimation
- ALE: on-distribution, noisy bin effect → poor estimation

DALE vs ALE - 10 Bins



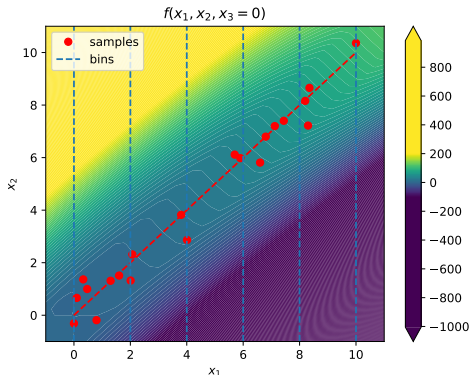
- DALE: on-distribution, noisy bin effect → poor estimation
- ALE: starts being OOD, noisy bin effect → poor estimation

DALE vs ALE - 10 Bins



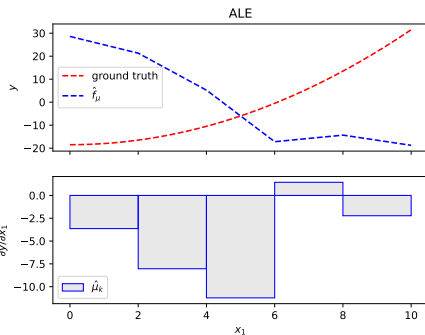
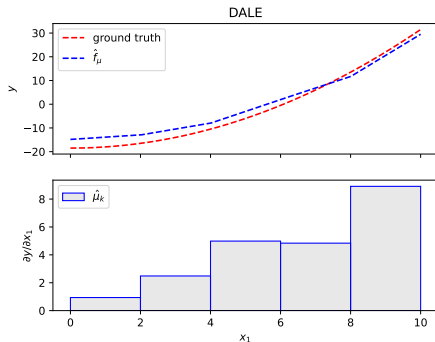
- DALE: on-distribution, noisy bin effect → poor estimation
- ALE: starts being OOD, noisy bin effect → poor estimation

DALE vs ALE - 5 Bins



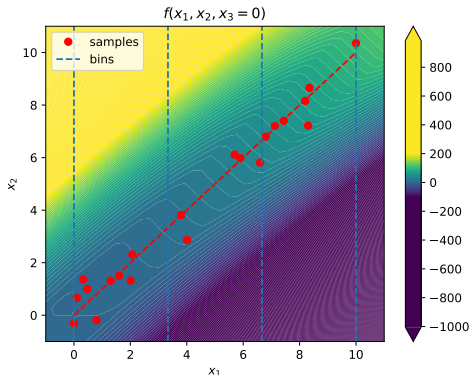
- DALE: on-distribution, robust bin effect → good estimation
- ALE: completely OOD, robust bin effect → poor estimation

DALE vs ALE - 5 Bins



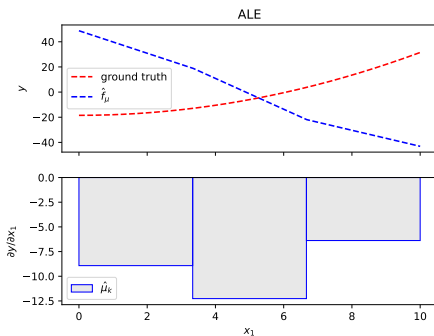
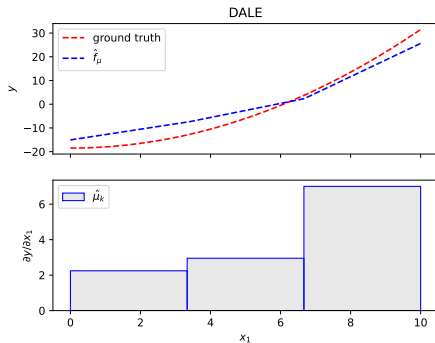
- DALE: on-distribution, robust bin effect → good estimation
- ALE: completely OOD, robust bin effect → poor estimation

DALE vs ALE - 3 Bins



- DALE: on-distribution, robust bin effect → good estimation
- ALE: completely OOD, robust bin effect → 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[2]
- $y \rightarrow$ 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_{hour} is an interesting feature

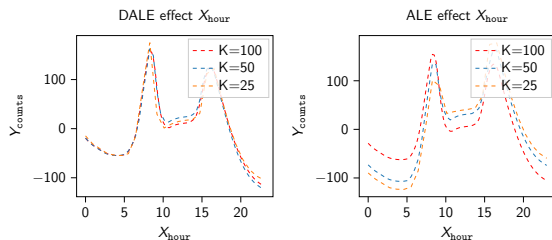


Figure: (Left) DALE (Left) and ALE (Right) plots for $K = \{25, 50, 100\}$

What next?

- How to (automatically) decide the optimal bin sizes?
 - ▶ Sometimes narrow bins are ok
 - ▶ Sometimes wide bins are needed
- Can we DALE are fast to decide optimal bin splitting?
- What about variable size bins?
- Model the uncertainty of the estimation?

DALE advantages can be a driver for future work

Thank you

- Questions?

References I

- [1] Daniel W. Apley and Jingyu Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 82(4):1059–1086, 2020.
- [2] Hadi Fanaee-T and Joao Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, pages 1–15, 2013.
- [3] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, oct 2001.
- [4] Christoph Molnar. *Interpretable Machine Learning*. 2 edition, 2022.