Paper presentation at ACML 2022

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

Vasilis Gkolemis[1,2] Theodore Dalamagas[1] Christos Diou[2]

[1]ATHENA Research and Innovation Center

[2]Harokopio University of Athens

December 2022

Interpretable ML

- Black-box model $f(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$, trained on $\mathcal{D}$
- Reveal interpretable properties:

  Which features are important (in general)?
  Which features have positive impact for a class?
- Categories:
  - $\rightarrow$ Global vs local
  - $\rightarrow$ Model-agnostic vs Model-specific
  - $\rightarrow$ Output? number, plot, instance etc.

Feature Effect: Global, Model-agnostic, outputs plot

Feature Effect

$y = f(x_s) \rightarrow$ Effect of a single feature x_s on the output y

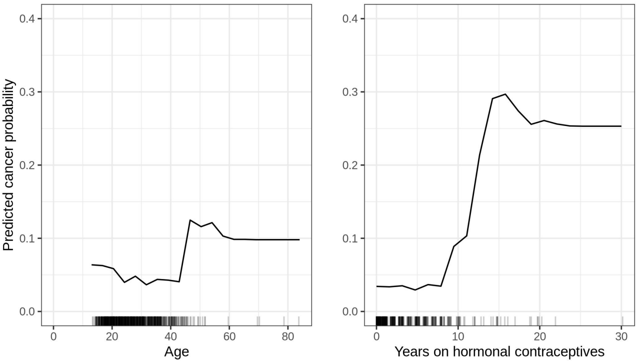


Figure: Image taken from Interpretable ML book [1]

Feature Effect is simple and intuitive.

Feature Effect Methods

- PDP
    - Expected outcome over x_c: $f(x_s) = \mathbb{E}_{\boldsymbol{x_c}}[f(x_s, \boldsymbol{x_c})]$
    - **Unrealistic instances**

PDP vs MPlot vs ALE

Feature Effect Methods

- PDP
  - Expected outcome over x_c: $f(x_s) = \mathbb{E}_{\mathbf{x_c}}[f(x_s, \mathbf{x_c})]$
  - **Unrealistic instances**
- MPlot
  - Expected outcome over $x_c|x_s$: $f(x_s) = \mathbb{E}_{\mathbf{x_c}|x_s}[f(x_s, \mathbf{x_c})]$
  - **Aggregated effects**

PDP vs MPlot vs ALE

Feature Effect Methods

- PDP
  - Expected outcome over x_c: $f(x_s) = \mathbb{E}_{\mathbf{x_c}}[f(x_s, \mathbf{x_c})]$
  - **Unrealistic instances**
- MPlot
  - Expected outcome over $x_c|x_s$: $f(x_s) = \mathbb{E}_{\mathbf{x_c}|x_s}[f(x_s, \mathbf{x_c})]$
  - **Aggregated effects**
- ALE
  - $f(x_s) = \int_{x_{min}}^{x_s} \mathbb{E}_{\mathbf{x_c}|x_s=z}[\frac{\partial f}{\partial x_s}(x_s, \mathbf{x_c})]\partial z$
  - **Resolves both failure modes**

PDP vs MPlot vs ALE

ALE approximation

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

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

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

- Slow $\rightarrow$ 2 evaluations of f per point
- Restrictive $\rightarrow$ change bin limits, pay again $2 * N$ evaluations of f
- Insecure $\rightarrow$ broad bins may create out of distribution samples instances

ALE approximation has some disadvantages

DALE - Differential ALE

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

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

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

- Fast → use of auto-differentiation, all derivatives in a single pass
- Versatile → point effects computed once, change bins without cost
- Secure → does not create artificial instances

DALE is a better approximation of ALE, for differentiable models

DALE is faster and versatile - Theory

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

- 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

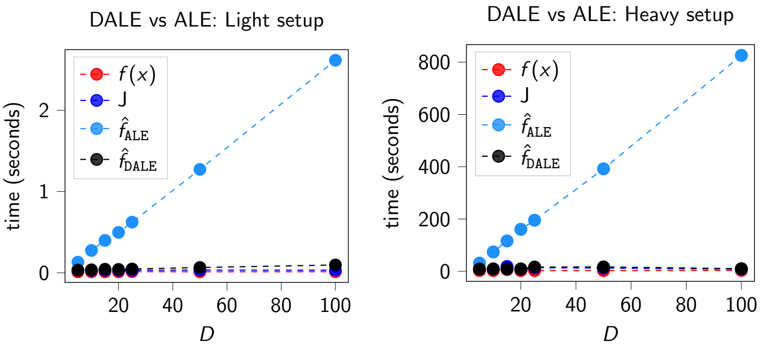


Figure: Light setup; small dataset, light f. Heavy setup; big dataset ($N = 10^5$), heavy f

DALE considerably accelerates the estimation

DALE uses only on-distribution samples - Theory

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

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

---

Wide bins without creating out of distribution instances

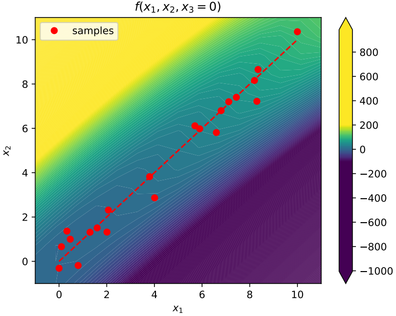
DALE uses only on-distribution samples - Experiments

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

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

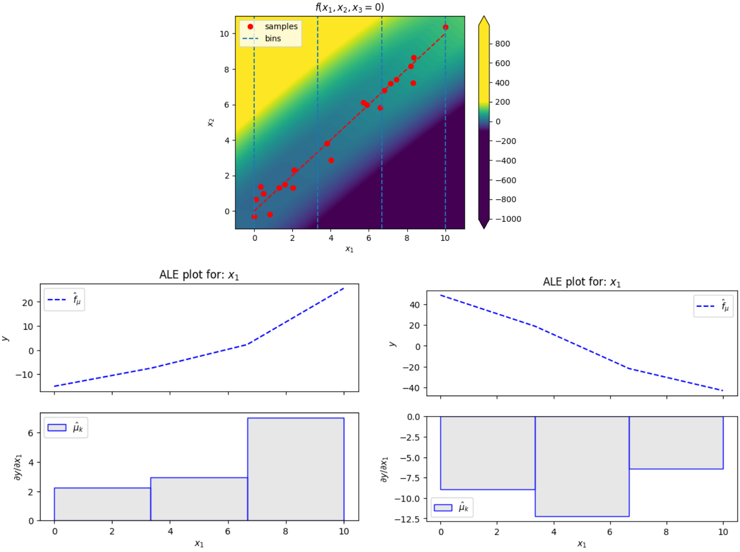
$$f_{\texttt{ALE}}(x_1) = \frac{x_1^2}{2}$$

- but the estimation is noisy $(x_1 x_3)$
- the samples are few

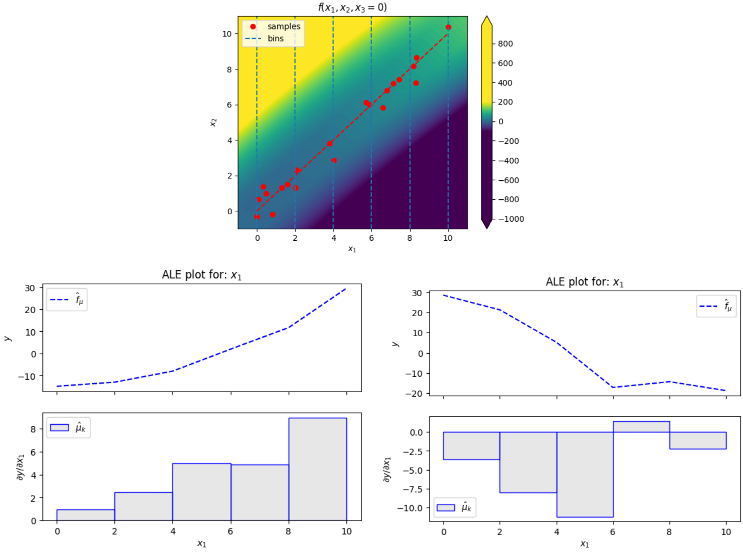


Intution: we need wider bins (more samples per bin)

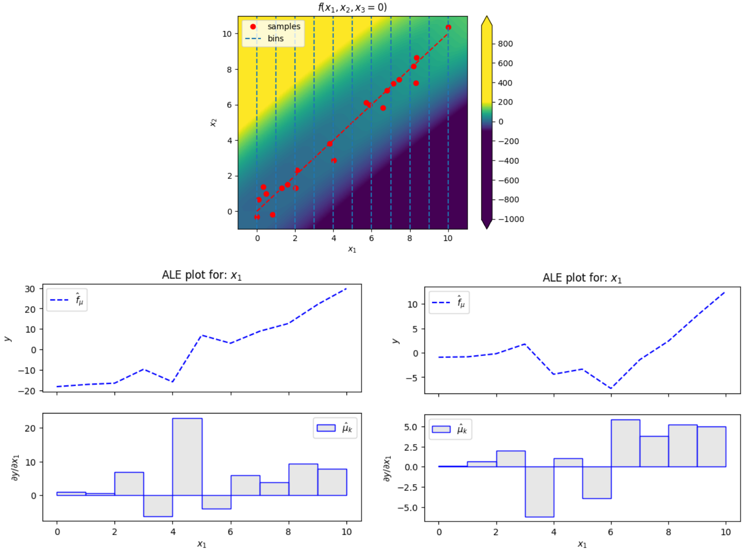
DALE vs ALE - 3 Bins



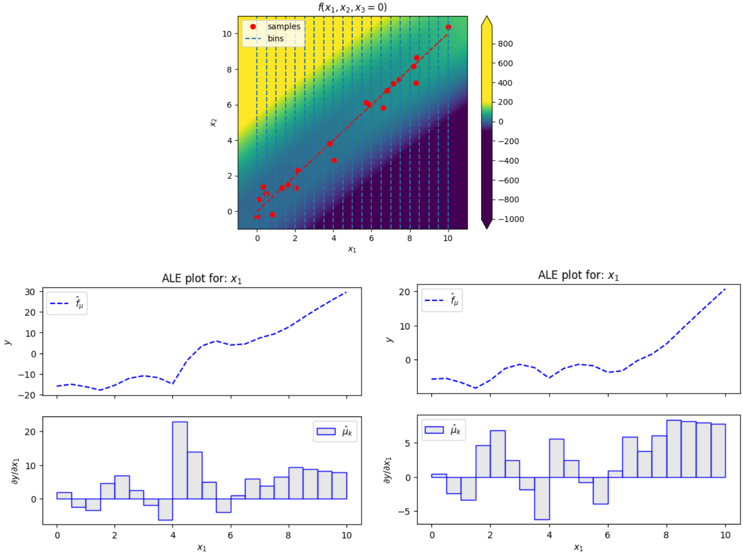
DALE vs ALE - 5 Bins



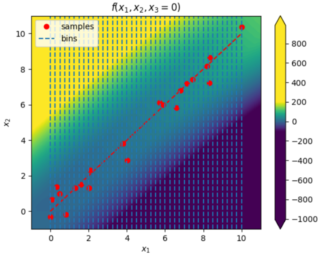
DALE vs ALE - 10 Bins

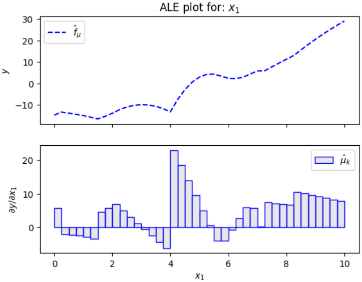


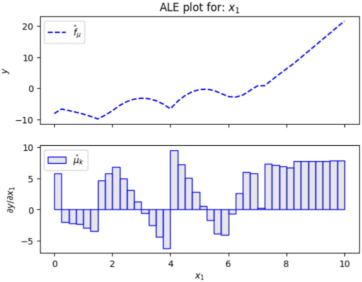
DALE vs ALE - 20 Bins



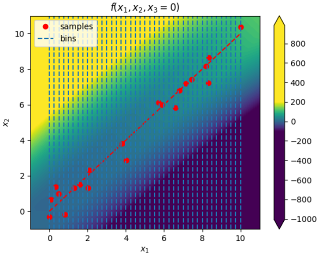
DALE vs ALE - 40 Bins



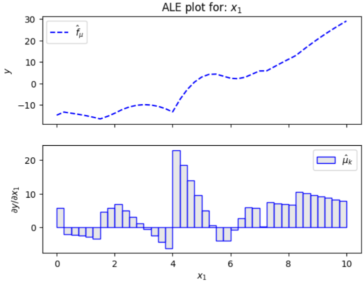




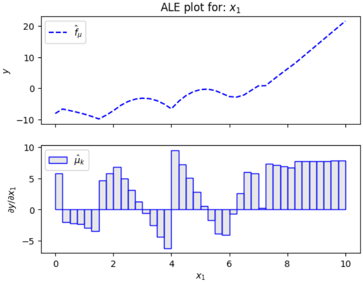
What next?



$f(x_1, x_2, x_3 = 0)$
samples
bins



ALE plot for: x_1
$\hat{f}_\mu$
$\hat{\mu}_k$
$\partial y/\partial x_1$



ALE plot for: x_1
$\hat{f}_\mu$
$\hat{\mu}_k$
$\partial y/\partial x_1$

References I

[1] Christoph Molnar. *Interpretable Machine Learning*. 2 edition, 2022.