CaDET: Interpretable Parametric Conditional Density Estimation with Decision Trees

Cyrus Cousins — Brown University

Matteo Riondato — Amherst College

Web: cyrus_cousins@brown.edu Mail: cs.brown.edu/~ccousins/cadet/

ECMLPKDD Würzburg 16.–20.09.2019

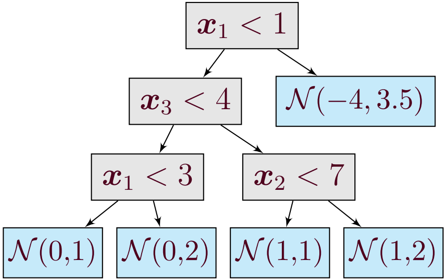
CaDET at a Glance

“Interpretable Parametric Conditional-Density-Estimation”

(1) *Conditional Density Estimation*: predict *distributions* (not point-estimates)

(2) CaDET predicts *parametric densities*, e.g. $\textsc{Gaussian}(1,1)$ or $\textsc{Beta}(3,2)$

(3) CaDET *trees and predictions* are *interpretable*



CaDET at a Glance

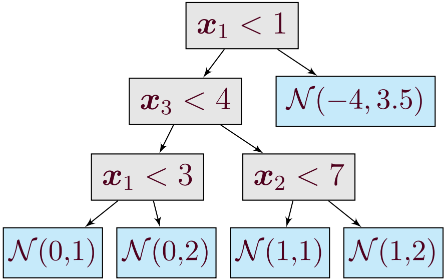
"Interpretable Parametric Conditional-Density-Estimation"

(1) *Conditional Density Estimation*: predict *distributions* (not point-estimates)

(2) CaDET predicts *parametric densities*, e.g. Gaussian$(1, 1)$ or Beta$(3, 2)$

(3) CaDET *trees and predictions* are *interpretable*

- Existing CDE tree methods
  - High training, query, and storage costs
  - Uninterpretable (non-parametric) estimates
  - High sample complexity



CADET at a Glance

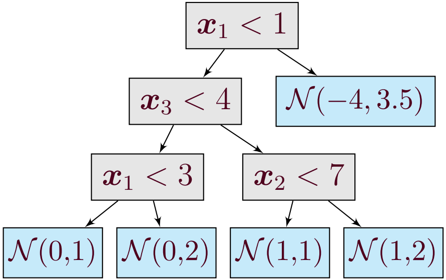
"Interpretable Parametric Conditional-Density-Estimation"

(1) *Conditional Density Estimation*: predict *distributions* (not point-estimates)

(2) CADET predicts *parametric densities*, e.g. GAUSSIAN$(1, 1)$ or BETA$(3, 2)$

(3) CADET *trees and predictions* are *interpretable*

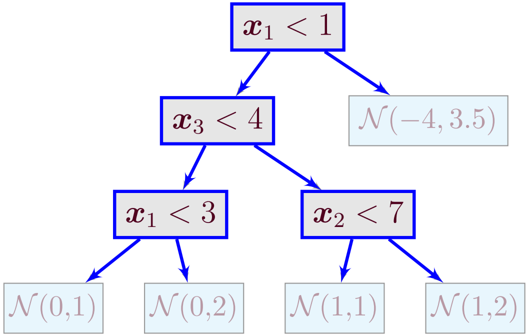
- Existing CDE tree methods
  - High training, query, and storage costs
  - Uninterpretable (non-parametric) estimates
  - High sample complexity
- CADET sacrifices *representativeness* for
  - Efficient training, storage, and querying
  - Easily understood parametric estimates
  - Generalizability



Interpretability in CDE Trees

Interpretability applies to:

(1) *Model:*
Tree structure easy to *visualize* & *understand*

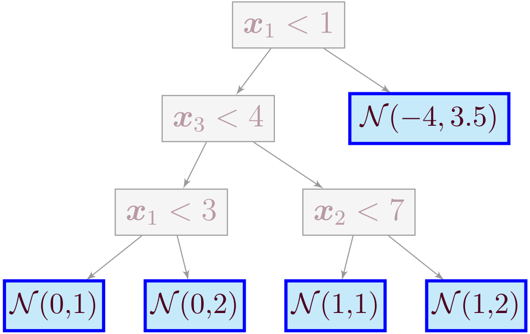


Interpretability in CDE Trees

Interpretability applies to:

(1) *Model:*
Tree structure easy to *visualize* & *understand*

(2) *Predictions:*
Model *output* must be simple



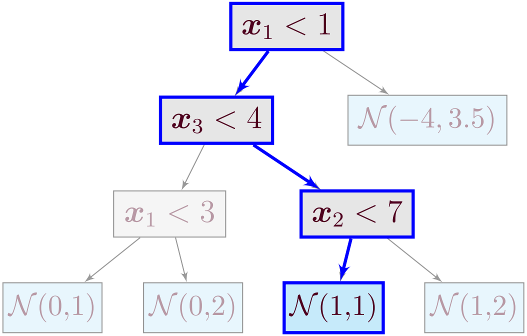
Interpretability in CDE Trees

Interpretability applies to:

(1) *Model:*
Tree structure easy to *visualize* & *understand*

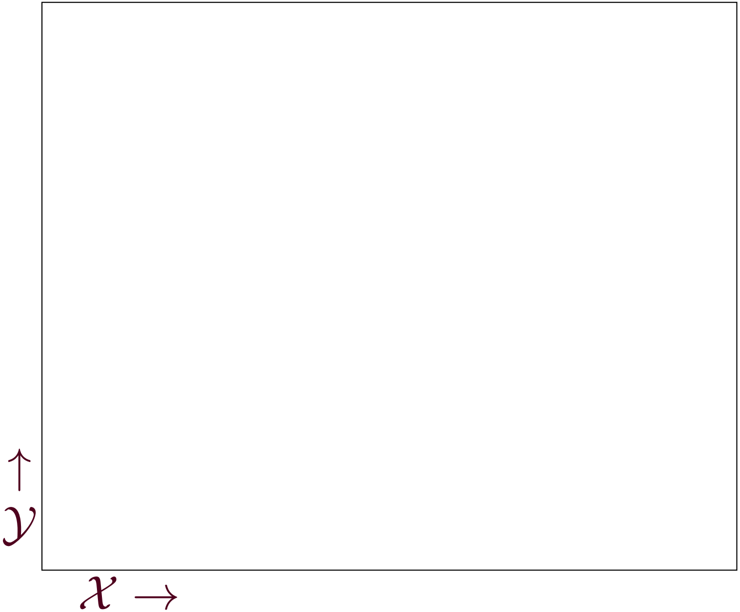
(2) *Predictions:*
Model *output* must be simple

(3) *Decision process:*
Easily audit decision making process



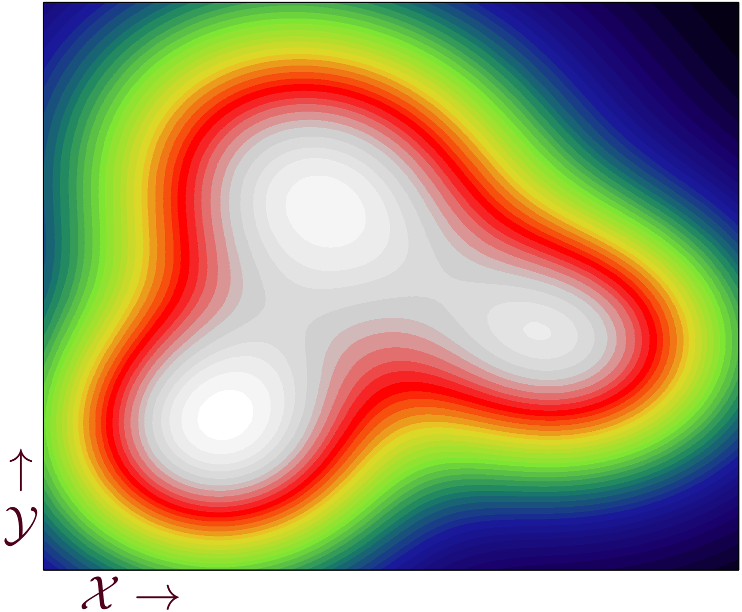
What is Conditional Density Estimation?

- Domain $\mathcal{X}$, codomain $\mathcal{Y}$



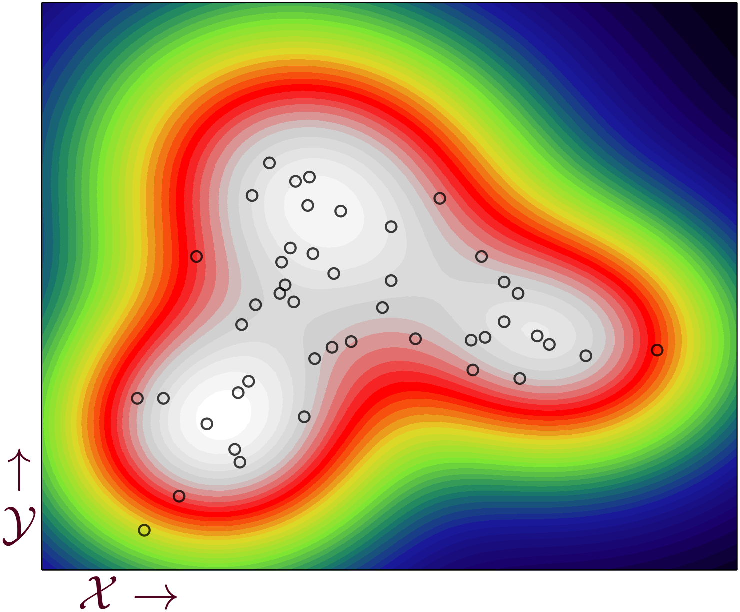
What is Conditional Density Estimation?

- Domain $\mathcal{X}$, codomain $\mathcal{Y}$
- PDF ρ over $\mathcal{X} \times \mathcal{Y}$



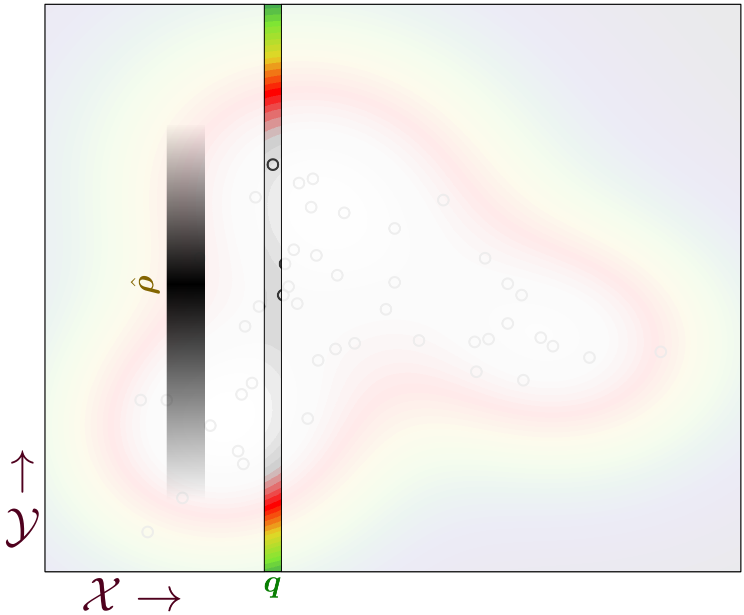
What is Conditional Density Estimation?

- Domain $\mathcal{X}$, codomain $\mathcal{Y}$
- PDF $\boldsymbol{\rho}$ over $\mathcal{X} \times \mathcal{Y}$
- Sample $(\boldsymbol{x}, \boldsymbol{y})$ of m points drawn with density $\boldsymbol{\rho}$



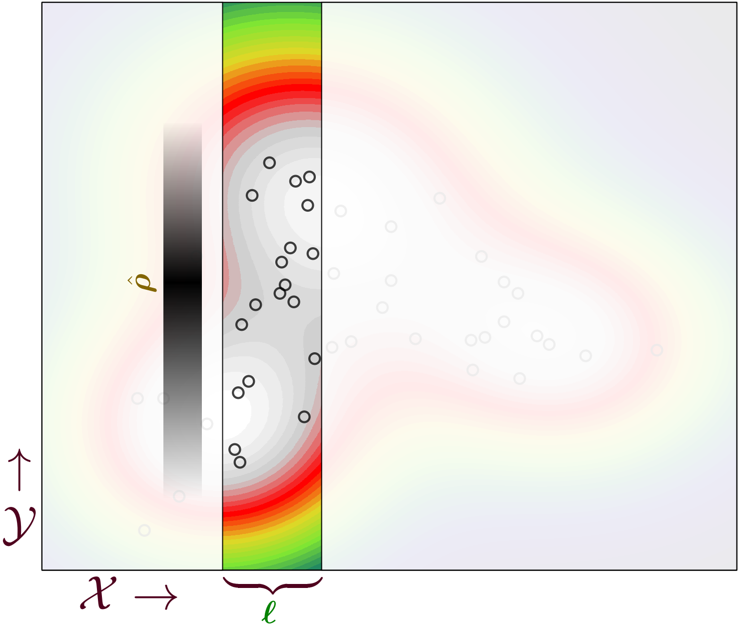
What is Conditional Density Estimation?

- Domain $\mathcal{X}$, codomain $\mathcal{Y}$
- PDF $\boldsymbol{\rho}$ over $\mathcal{X} \times \mathcal{Y}$
- Sample $(\boldsymbol{x}, \boldsymbol{y})$ of m points drawn with density $\boldsymbol{\rho}$
- Condition on query $\boldsymbol{q} \in \mathcal{X}$
- Estimate $\hat{\rho}(\cdot \mid \boldsymbol{q}) \approx \boldsymbol{\rho}(\cdot \mid \boldsymbol{q})$



What is Conditional Density Estimation?

- Domain $\mathcal{X}$, codomain $\mathcal{Y}$
- PDF $\boldsymbol{\rho}$ over $\mathcal{X} \times \mathcal{Y}$
- Sample $(\boldsymbol{x}, \boldsymbol{y})$ of m points drawn with density $\boldsymbol{\rho}$
- Condition on query $\boldsymbol{q} \in \mathcal{X}$
- Estimate $\hat{\rho}(\cdot \mid \boldsymbol{q}) \approx \boldsymbol{\rho}(\cdot \mid \boldsymbol{q})$
- Decision trees:
  Fit PDF $\hat{\rho}$ to *leaf* $\boldsymbol{\ell} \ni \boldsymbol{q}$



Why Conditional Density Estimation?

Supervised Learning: for any $\boldsymbol{q} \in \mathcal{X}$, predict statistics of $\boldsymbol{\rho}(\,\cdot \mid \boldsymbol{q})$

Why Conditional Density Estimation?

Supervised Learning: for any $\boldsymbol{q} \in \mathcal{X}$, predict statistics of $\boldsymbol{\rho}(\,\cdot \mid \boldsymbol{q})$

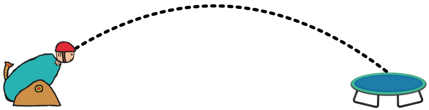
		Prediction Type	
		Summary	**Distribution**
$\mathcal{Y}$	**Discrete**	Hard Classification $\arg\max_y \mathbb{P}(y \mid \boldsymbol{q})$	Soft Classification $\mathbb{P}(\,\cdot \mid \boldsymbol{q})$
	Continuous	Regression $\mathbb{E}_{(x,y)\sim\boldsymbol{\rho}}\left[y \mid x = \boldsymbol{q}\right]$	CDE $\rho(\,\cdot \mid \boldsymbol{q})$

Why Conditional Density Estimation?

Supervised Learning: for any $\boldsymbol{q} \in \mathcal{X}$, predict statistics of $\boldsymbol{\rho}(\cdot \mid \boldsymbol{q})$

		Prediction Type	
		Summary	**Distribution**
$\mathcal{Y}$	**Discrete**	Hard Classification $\underset{y}{\operatorname{argmax}}\, \mathbb{P}(y \mid \boldsymbol{q})$	Soft Classification $\mathbb{P}(\cdot \mid \boldsymbol{q})$
	Continuous	Regression $\underset{(x,y)\sim\boldsymbol{\rho}}{\mathbb{E}}[y \mid x = \boldsymbol{q}]$	CDE $\rho(\cdot \mid \boldsymbol{q})$

- Regression is a lossy process
  - Only estimate *average outcome*

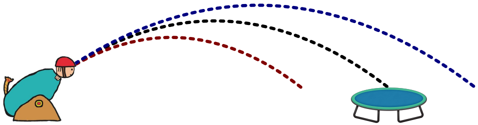


Why Conditional Density Estimation?

Supervised Learning: for any $\boldsymbol{q} \in \mathcal{X}$, predict statistics of $\boldsymbol{\rho}(\cdot \mid \boldsymbol{q})$

		Prediction Type	
		Summary	**Distribution**
$\mathcal{Y}$	**Discrete**	Hard Classification $\operatorname*{argmax}_y \mathbb{P}(y \mid \boldsymbol{q})$	Soft Classification $\mathbb{P}(\cdot \mid \boldsymbol{q})$
	Continuous	Regression $\mathop{\mathbb{E}}_{(x,y)\sim\boldsymbol{\rho}}[y \mid x = \boldsymbol{q}]$	CDE $\rho(\cdot \mid \boldsymbol{q})$

- Regression is a lossy process
  - Only estimate *average outcome*
  - Want to reason about *many possibilities*

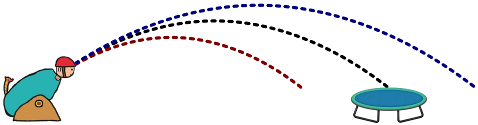


Why Conditional Density Estimation?

Supervised Learning: for any $\boldsymbol{q} \in \mathcal{X}$, predict statistics of $\boldsymbol{\rho}(\,\cdot \mid \boldsymbol{q})$

		Prediction Type	
		Summary	**Distribution**
$\mathcal{Y}$	**Discrete**	Hard Classification $\operatorname*{argmax}_y \mathbb{P}(y \mid \boldsymbol{q})$	Soft Classification $\mathbb{P}(\,\cdot \mid \boldsymbol{q})$
	Continuous	Regression $\mathbb{E}_{(x,y)\sim\boldsymbol{\rho}}\left[y \mid x = \boldsymbol{q}\right]$	CDE $\rho(\,\cdot \mid \boldsymbol{q})$

- Regression is a lossy process
  - Only estimate *average outcome*
  - Want to reason about *many possibilities*



- CDE quantifies *uncertainty* due to noise or ambiguity
  - Generalizes *soft classification* to arbitrary $\mathcal{Y}$
  - Postprocess to estimate mean, median, . . .

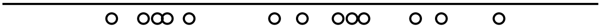
Parametric vs Nonparametric CDE Trees

- CDE with decision trees:
  - Tree splits $\mathcal{X}$ into *disjoint cover* (leaves)
  - Estimate distribution over $\mathcal{Y}$ at each leaf ℓ

Parametric vs Nonparametric CDE Trees

- CDE with decision trees:
  - Tree splits $\mathcal{X}$ into *disjoint cover* (leaves)
  - Estimate distribution over $\mathcal{Y}$ at each leaf ℓ

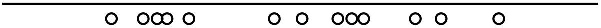
Fitting Labels at ℓ



Parametric vs Nonparametric CDE Trees

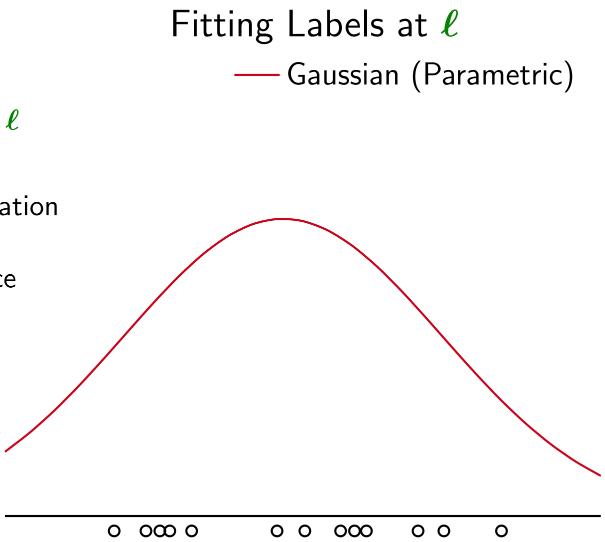
- CDE with decision trees:
  - Tree splits $\mathcal{X}$ into *disjoint cover* (leaves)
  - Estimate distribution over $\mathcal{Y}$ at each leaf ℓ
- Parametric CDE trees ($\textsc{CaDET}$)
  - Quickly converge to parametric approximation
  - Learn good splits with small samples
  - Simple predictions, understood at a glance

Fitting Labels at ℓ



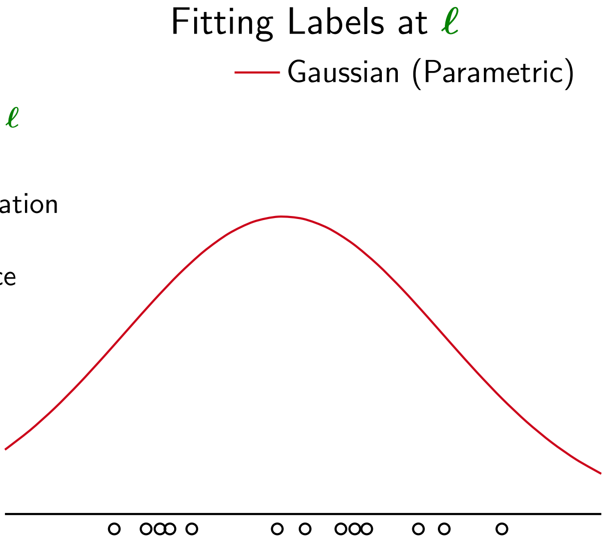
Parametric vs Nonparametric CDE Trees

- CDE with decision trees:
  - Tree splits $\mathcal{X}$ into *disjoint cover* (leaves)
  - Estimate distribution over $\mathcal{Y}$ at each leaf ℓ
- Parametric CDE trees (CaDET)
  - Quickly converge to parametric approximation
  - Learn good splits with small samples
  - Simple predictions, understood at a glance



Parametric vs Nonparametric CDE Trees

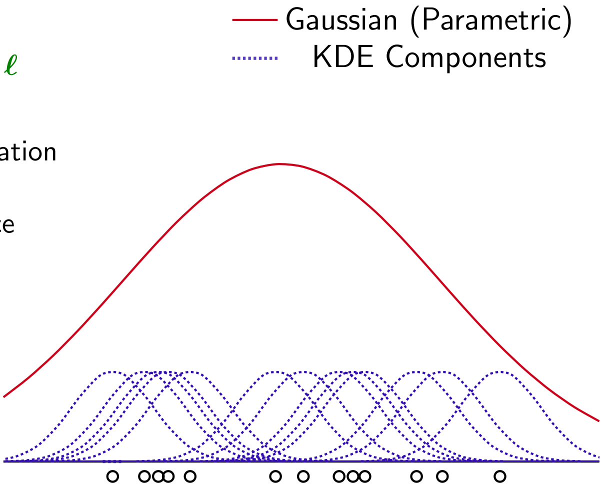
- CDE with decision trees:
  - Tree splits $\mathcal{X}$ into *disjoint cover* (leaves)
  - Estimate distribution over $\mathcal{Y}$ at each leaf ℓ
- Parametric CDE trees ($\textsc{CaDET}$)
  - Quickly converge to parametric approximation
  - Learn good splits with small samples
  - Simple predictions, understood at a glance
- Nonparametric CDE trees
  - Asymptotic consistency
    - *Eventually* get it right
  - Poor sample complexity
    - Must fit distribution *at each leaf*
    - More susceptible to *overfitting*



Parametric vs Nonparametric CDE Trees

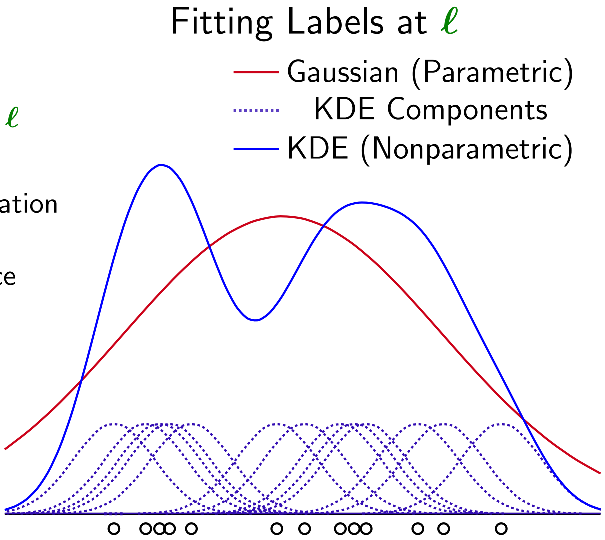
- CDE with decision trees:
  - Tree splits $\mathcal{X}$ into *disjoint cover* (leaves)
  - Estimate distribution over $\mathcal{Y}$ at each leaf ℓ
- Parametric CDE trees (CADET)
  - Quickly converge to parametric approximation
  - Learn good splits with small samples
  - Simple predictions, understood at a glance
- Nonparametric CDE trees
  - Asymptotic consistency
    - *Eventually* get it right
  - Poor sample complexity
    - Must fit distribution *at each leaf*
    - More susceptible to *overfitting*

Fitting Labels at ℓ



Parametric vs Nonparametric CDE Trees

- CDE with decision trees:
  - Tree splits $\mathcal{X}$ into *disjoint cover* (leaves)
  - Estimate distribution over $\mathcal{Y}$ at each leaf ℓ
- Parametric CDE trees (CaDET)
  - Quickly converge to parametric approximation
  - Learn good splits with small samples
  - Simple predictions, understood at a glance
- Nonparametric CDE trees
  - Asymptotic consistency
    - *Eventually* get it right
  - Poor sample complexity
    - Must fit distribution *at each leaf*
    - More susceptible to *overfitting*



What are Decision Trees Again?

- Fitting optimal tree to $(\boldsymbol{x}, \boldsymbol{y})$ is NP-hard
  Standard heuristic: impurity reduction
  (1) Start with a singleton tree, and impurity criterion $\mathrm{I}(\cdot)$
    - $\mathrm{I}(\boldsymbol{y})$ measures *disagreement* among $\boldsymbol{y}$
    - MSE, entropy, GINI, ...

What are Decision Trees Again?

- Fitting optimal tree to $(\boldsymbol{x}, \boldsymbol{y})$ is NP-hard
  Standard heuristic: impurity reduction
  (1) Start with a singleton tree, and impurity criterion $\mathrm{I}(\cdot)$
    - $\mathrm{I}(\boldsymbol{y})$ measures *disagreement* among $\boldsymbol{y}$
    - MSE, entropy, GINI, ...

  (2) Select split of $(\boldsymbol{x}, \boldsymbol{y})$ into $(\boldsymbol{x}_L, \boldsymbol{y}_L)$ and $(\boldsymbol{x}_R, \boldsymbol{y}_R)$ to maximize *impurity reduction*:

$$(m_L + m_R)\,\mathrm{I}(\boldsymbol{y}) - \big(m_L\,\mathrm{I}(\boldsymbol{y}_L) + m_R\,\mathrm{I}(\boldsymbol{y}_R)\big)$$

What are Decision Trees Again?

- Fitting optimal tree to $(\boldsymbol{x}, \boldsymbol{y})$ is NP-hard

  Standard heuristic: impurity reduction

  (1) Start with a singleton tree, and impurity criterion $\mathrm{I}(\cdot)$
      - $\mathrm{I}(\boldsymbol{y})$ measures *disagreement* among $\boldsymbol{y}$
      - MSE, entropy, GINI, ...

  (2) Select split of $(\boldsymbol{x}, \boldsymbol{y})$ into $(\boldsymbol{x}_L, \boldsymbol{y}_L)$ and $(\boldsymbol{x}_R, \boldsymbol{y}_R)$ to maximize *impurity reduction*:

$$(m_L + m_R)\,\mathrm{I}(\boldsymbol{y}) - \big(m_L\,\mathrm{I}(\boldsymbol{y}_L) + m_R\,\mathrm{I}(\boldsymbol{y}_R)\big)$$

  (3) Repeat until termination condition is met
      - Maximum depth, minimum samples per leaf, ...

What are Decision Trees Again?

- Fitting optimal tree to $(\boldsymbol{x}, \boldsymbol{y})$ is NP-hard
  Standard heuristic: impurity reduction
  (1) Start with a singleton tree, and impurity criterion $\mathrm{I}(\cdot)$
    - $\mathrm{I}(\boldsymbol{y})$ measures *disagreement* among $\boldsymbol{y}$
    - MSE, entropy, GINI, ...

  (2) Select split of $(\boldsymbol{x}, \boldsymbol{y})$ into $(\boldsymbol{x}_L, \boldsymbol{y}_L)$ and $(\boldsymbol{x}_R, \boldsymbol{y}_R)$ to maximize *impurity reduction*:

$$(m_L + m_R)\,\mathrm{I}(\boldsymbol{y}) - \big(m_L\,\mathrm{I}(\boldsymbol{y}_L) + m_R\,\mathrm{I}(\boldsymbol{y}_R)\big)$$

  (3) Repeat until termination condition is met
    - Maximum depth, minimum samples per leaf, ...
- Lower impurity $\implies$ leaf label more accurately describes $\boldsymbol{y}$

CaDET Trees

- Hyperparameter: *parametric distribution family* $\mathcal{F}$ over $\mathcal{Y}$
  - All CaDET predictions (distributions) belong to $\mathcal{F}$
  - For this talk, $\mathcal{F}$ in *exponential class*

CaDET Trees

- Hyperparameter: *parametric distribution family* $\mathcal{F}$ over $\mathcal{Y}$
  - All CaDET predictions (distributions) belong to $\mathcal{F}$
  - For this talk, $\mathcal{F}$ in *exponential class*
- Given query point $\boldsymbol{q}$, CaDET trees
  (1) Find the leaf that contains $\boldsymbol{q}$, with training labels $\boldsymbol{y}$
  (2) Return $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y})$

CaDET Trees

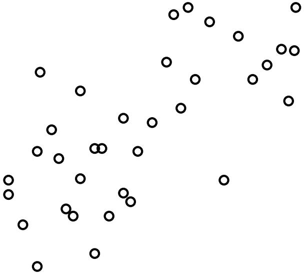
- Hyperparameter: *parametric distribution family* $\mathcal{F}$ over $\mathcal{Y}$
  - All CaDET predictions (distributions) belong to $\mathcal{F}$
  - For this talk, $\mathcal{F}$ in *exponential class*
- Given query point $\boldsymbol{q}$, CaDET trees
  (1) Find the leaf that contains $\boldsymbol{q}$, with training labels $\boldsymbol{y}$
  (2) Return $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y})$
- CaDET trees optimize *cross entropy impurity*
  - Evaluate CDE with *cross entropy loss* $\ell_{\mathsf{CE}}(y \,|\, \hat{\boldsymbol{\rho}}) \doteq -\ln \hat{\boldsymbol{\rho}}(y)$
  - *Cross entropy impurity* $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}) \doteq \dfrac{1}{m} \displaystyle\sum_{i=1}^{m} \ell_{\mathsf{CE}}\big(\boldsymbol{y}_i \,|\, \mathrm{MLE}(\boldsymbol{y}; \mathcal{F})\big)$

CaDET Trees

- Hyperparameter: *parametric distribution family* $\mathcal{F}$ over $\mathcal{Y}$
  - All CaDET predictions (distributions) belong to $\mathcal{F}$
  - For this talk, $\mathcal{F}$ in *exponential class*
- Given query point $\boldsymbol{q}$, CaDET trees
  (1) Find the leaf that contains $\boldsymbol{q}$, with training labels $\boldsymbol{y}$
  (2) Return $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y})$
- CaDET trees optimize *cross entropy impurity*
  - Evaluate CDE with *cross entropy loss* $\ell_{\mathsf{CE}}(y \,|\, \hat{\boldsymbol{\rho}}) \doteq -\ln \hat{\boldsymbol{\rho}}(y)$
  - *Cross entropy impurity* $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}) \doteq \frac{1}{m}\sum_{i=1}^{m} \ell_{\mathsf{CE}}\big(\boldsymbol{y}_i \,|\, \mathrm{MLE}(\boldsymbol{y}; \mathcal{F})\big)$
- CaDET organizes computation such that:
  - Evaluating impurity reduction requires *constant work*
  - Leaves require *constant storage*

Training CADET Trees

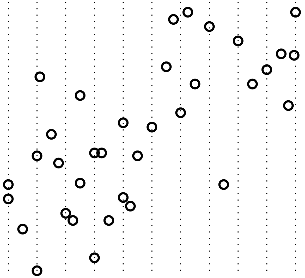
Cross Entropy Impurity: $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}) \doteq \frac{1}{m}\sum_{i=1}^{m} \ell_{\mathsf{CE}}\big(\boldsymbol{y}_i \big|\, \mathrm{MLE}(\boldsymbol{y};\mathcal{F})\big)$



(1) Given training
points $\boldsymbol{x}_{i=1}^{m} \in \mathbb{R}^m$ (x axis)
labels $\boldsymbol{y}_{i=1}^{m} \in \mathbb{R}^m$ (y axis)

Training CADET Trees

Cross Entropy Impurity: $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}) \doteq \frac{1}{m}\sum_{i=1}^{m} \ell_{\mathsf{CE}}\big(\boldsymbol{y}_i \,|\, \mathrm{MLE}(\boldsymbol{y};\mathcal{F})\big)$



(1) Given training

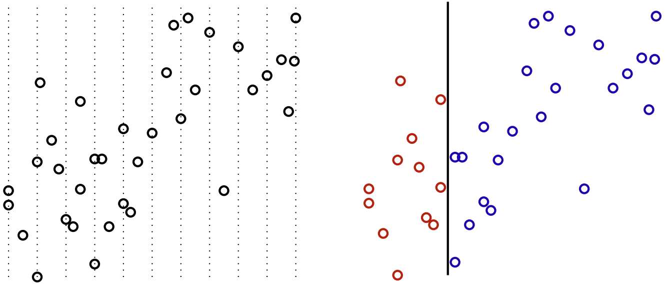
points $\boldsymbol{x}_{i=1}^{m} \in \mathbb{R}^m$ (x axis)

labels $\boldsymbol{y}_{i=1}^{m} \in \mathbb{R}^m$ (y axis)

(2) Evaluate possible splits

Training CADET Trees

Cross Entropy Impurity: $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}) \doteq \frac{1}{m}\sum_{i=1}^{m} \ell_{\mathsf{CE}}\big(\boldsymbol{y}_i \,|\, \mathrm{MLE}(\boldsymbol{y};\mathcal{F})\big)$



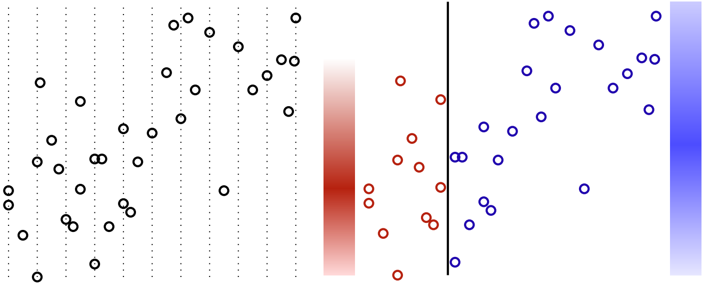
(1) Given training *points* $\boldsymbol{x}_{i=1}^m \in \mathbb{R}^m$ (x axis) *labels* $\boldsymbol{y}_{i=1}^m \in \mathbb{R}^m$ (y axis)

(2) Evaluate possible splits

(1) Consider split $\boldsymbol{y}_L$, $\boldsymbol{y}_R$

Training CaDET Trees

Cross Entropy Impurity: $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}) \doteq \frac{1}{m} \sum_{i=1}^{m} \ell_{\mathsf{CE}}\left(\boldsymbol{y}_i \middle| \, \mathrm{MLE}(\boldsymbol{y}; \mathcal{F})\right)$



(1) Given training points $\boldsymbol{x}_{i=1}^{m} \in \mathbb{R}^m$ (x axis) labels $\boldsymbol{y}_{i=1}^{m} \in \mathbb{R}^m$ (y axis)

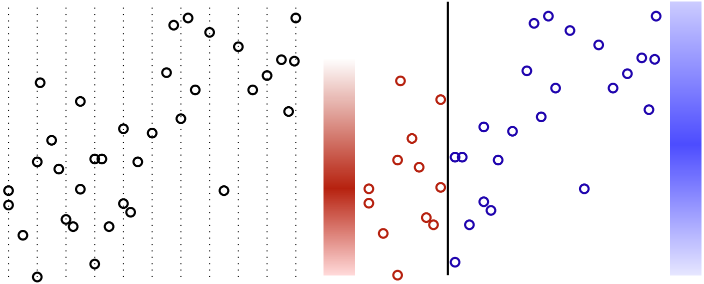
(2) Evaluate possible splits

(1) Consider split $\boldsymbol{y}_L$, $\boldsymbol{y}_R$

(2) Fit $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y}_L)$, $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y}_R)$

Training CaDET Trees

Cross Entropy Impurity: $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}) \doteq \frac{1}{m}\sum_{i=1}^{m} \ell_{\mathsf{CE}}\left(\boldsymbol{y}_i \middle| \mathrm{MLE}(\boldsymbol{y};\mathcal{F})\right)$



(1) Given training *points* $\boldsymbol{x}_{i=1}^{m} \in \mathbb{R}^m$ (x axis) *labels* $\boldsymbol{y}_{i=1}^{m} \in \mathbb{R}^m$ (y axis)

(2) Evaluate possible splits

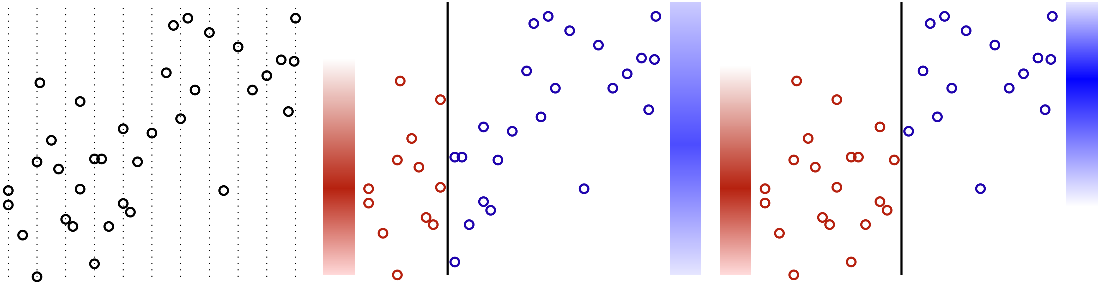
(1) Consider split $\boldsymbol{y}_L$, $\boldsymbol{y}_R$

(2) Fit $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y}_L)$, $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y}_R)$

(3) $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}_L)$: low
$\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}_R)$: high

Training CADET Trees

Cross Entropy Impurity: $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}) \doteq \frac{1}{m}\sum_{i=1}^{m} \ell_{\mathsf{CE}}\left(\boldsymbol{y}_i \middle| \mathrm{MLE}(\boldsymbol{y};\mathcal{F})\right)$



(1) Given training *points* $\boldsymbol{x}_{i=1}^{m} \in \mathbb{R}^m$ (x axis) *labels* $\boldsymbol{y}_{i=1}^{m} \in \mathbb{R}^m$ (y axis)

(2) Evaluate possible splits

(1) Consider split $\boldsymbol{y}_L$, $\boldsymbol{y}_R$

(2) Fit $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y}_L), \mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y}_R)$

(3) $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}_L)$: low
$\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}_R)$: high

(1) Now consider $\boldsymbol{y}'_L$, $\boldsymbol{y}'_R$

(2) Fit $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y}'_L)$, $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y}'_R)$

(3) $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}'_L)$: same
$\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}'_R)$: lower

Why CDE-Specific Impurity Criteria?

CaDET sounds complicated, why not just use $\mathrm{I}_{\mathrm{MSE}}(\cdot)$?

Why CDE-Specific Impurity Criteria?

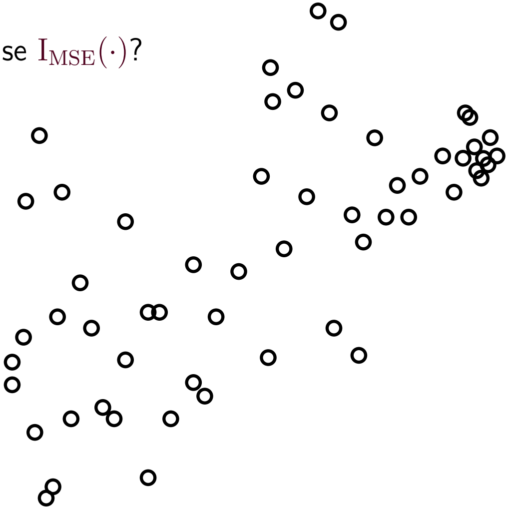
CaDET sounds complicated, why not just use $\mathrm{I}_{\mathrm{MSE}}(\cdot)$?

- Consider maximizing $\mathrm{I}_{\mathrm{MSE}}(\cdot)$-reduction
  - Only sensitive to changes in *expectation*
  - Insensitive to changes in *variance*
    - ⚠ Problem even for Gaussian family

Why CDE-Specific Impurity Criteria?

CADET sounds complicated, why not just use $\mathrm{I}_{\mathrm{MSE}}(\cdot)$?

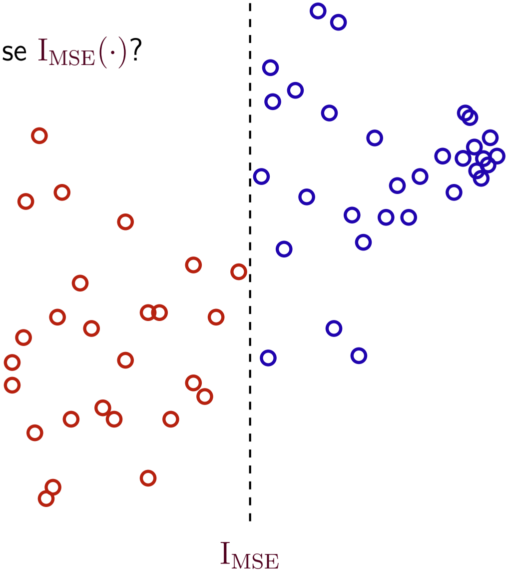
- Consider maximizing $\mathrm{I}_{\mathrm{MSE}}(\cdot)$-reduction
  - Only sensitive to changes in *expectation*
  - Insensitive to changes in *variance*
    - ⚠ Problem even for Gaussian family



Why CDE-Specific Impurity Criteria?

CaDET sounds complicated, why not just use $I_{\mathrm{MSE}}(\cdot)$?

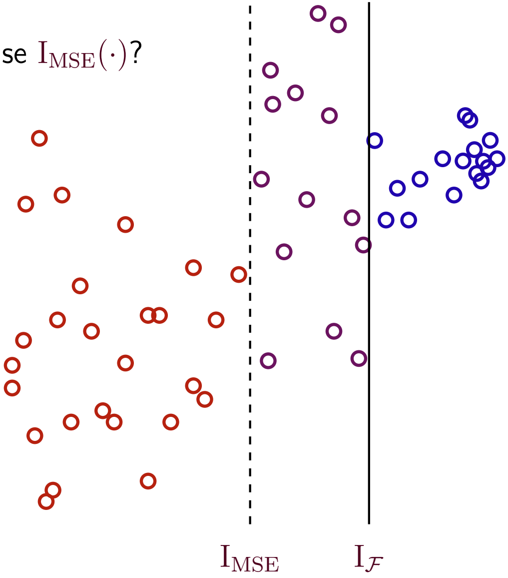
- Consider maximizing $I_{\mathrm{MSE}}(\cdot)$-reduction
  - Only sensitive to changes in *expectation*
  - Insensitive to changes in *variance*
    - ⚠ Problem even for Gaussian family



Why CDE-Specific Impurity Criteria?

CaDET sounds complicated, why not just use $\mathrm{I}_{\mathrm{MSE}}(\cdot)$?

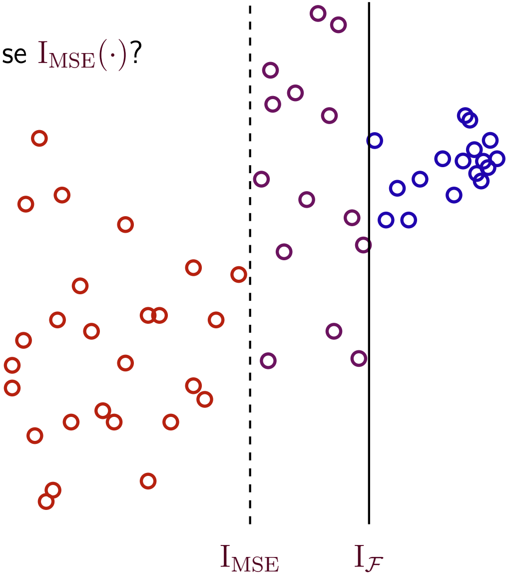
- Consider maximizing $\mathrm{I}_{\mathrm{MSE}}(\cdot)$-reduction
  - Only sensitive to changes in *expectation*
  - Insensitive to changes in *variance*
    - ⚠ Problem even for Gaussian family



Why CDE-Specific Impurity Criteria?

CaDET sounds complicated, why not just use $\mathrm{I}_{\mathrm{MSE}}(\cdot)$?

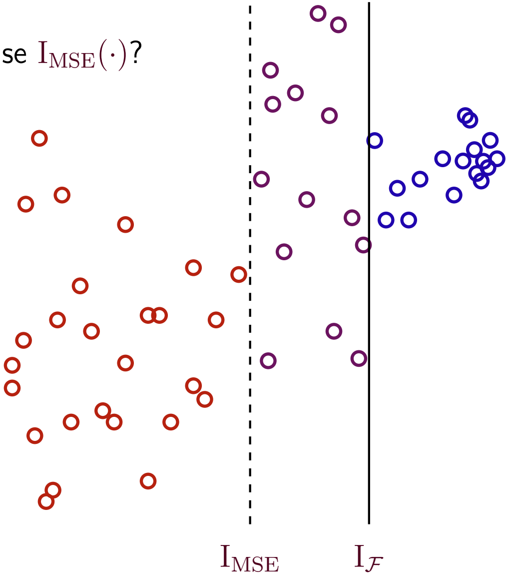
- Consider maximizing $\mathrm{I}_{\mathrm{MSE}}(\cdot)$-reduction
  - Only sensitive to changes in *expectation*
  - Insensitive to changes in *variance*
    - ⚠ Problem even for Gaussian family
  - $\mathrm{I}_{\mathrm{MSE}}(\cdot)$ undefined for $\mathcal{Y} \neq \mathbb{R}^d$



Why CDE-Specific Impurity Criteria?

CADET sounds complicated, why not just use $\mathrm{I}_{\mathrm{MSE}}(\cdot)$?

- Consider maximizing $\mathrm{I}_{\mathrm{MSE}}(\cdot)$-reduction
  - Only sensitive to changes in *expectation*
  - Insensitive to changes in *variance*
    - ⚠ Problem even for Gaussian family
  - $\mathrm{I}_{\mathrm{MSE}}(\cdot)$ undefined for $\mathcal{Y} \neq \mathbb{R}^d$
- *Takeaway*: $\mathrm{I}_{\mathcal{F}}(\cdot)$ depends on $\mathcal{F}$
  - "Pick the splits that improve the fits"



Organizing Computation with Sufficient Statistics

- A leaf needs training labels $\boldsymbol{y}$ to select splits in training and answer queries
  - CART *summarize* $\boldsymbol{y}$ with *class frequencies* or *means*
  - "Incremental updates" make split search fast
  - CaDET needs $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y})$ to train and $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y})$ to query

Organizing Computation with Sufficient Statistics

- A leaf needs training labels $\boldsymbol{y}$ to select splits in training and answer queries
  - CART *summarize* $\boldsymbol{y}$ with *class frequencies* or *means*
  - "Incremental updates" make split search fast
  - $\textsc{CaDET}$ needs $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y})$ to train and $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y})$ to query
- *Sufficient statistics* $\mathrm{w}^{(m)}(\boldsymbol{y})$ w.r.t. $\mathcal{F}$ summarize $\boldsymbol{y}$ in $\mathcal{Y}^m$
  - Compute $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y})$ and minimize $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y})$ from $\mathrm{w}^{(m)}(\boldsymbol{y})$

Organizing Computation with Sufficient Statistics

- A leaf needs training labels $\boldsymbol{y}$ to select splits in training and answer queries
  - CART *summarize* $\boldsymbol{y}$ with *class frequencies* or *means*
  - "Incremental updates" make split search fast
  - CADET needs $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y})$ to train and $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y})$ to query
- *Sufficient statistics* $\mathrm{w}^{(m)}(\boldsymbol{y})$ w.r.t. $\mathcal{F}$ summarize $\boldsymbol{y}$ in $\mathcal{Y}^m$
  - Compute $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y})$ and minimize $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y})$ from $\mathrm{w}^{(m)}(\boldsymbol{y})$

Family $\mathcal{F}$	**Suff. Stat.** $\mathrm{w}^{(m)}(\boldsymbol{y})$	**Log Density** $\ln \boldsymbol{\rho}(y)$
$\textsc{Gaussian}(\mu, \sigma^2)$	$\sum_{i=1}^{m} \boldsymbol{y}_i, \sum_{i=1}^{m} \boldsymbol{y}_i^2$	$-\frac{1}{2}\ln(2\pi\sigma^2) - \frac{y^2 - \quad y\mu + \mu^2}{2\sigma^2}$

Organizing Computation with Sufficient Statistics

- A leaf needs training labels $\boldsymbol{y}$ to select splits in training and answer queries
  - CART *summarize* $\boldsymbol{y}$ with *class frequencies* or *means*
  - "Incremental updates" make split search fast
  - CADET needs $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y})$ to train and $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y})$ to query
- *Sufficient statistics* $\mathrm{w}^{(m)}(\boldsymbol{y})$ w.r.t. $\mathcal{F}$ summarize $\boldsymbol{y}$ in $\mathcal{Y}^m$
  - Compute $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y})$ and minimize $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y})$ from $\mathrm{w}^{(m)}(\boldsymbol{y})$

Family $\mathcal{F}$	**Suff. Stat.** $\mathrm{w}^{(m)}(\boldsymbol{y})$	**Log Density** $\ln \boldsymbol{\rho}(y)$
$\textsc{Gaussian}(\mu, \sigma^2)$	$\sum_{i=1}^{m} \boldsymbol{y}_i,\ \sum_{i=1}^{m} \boldsymbol{y}_i^2$	$-\frac{1}{2}\ln(2\pi\sigma^2) - \frac{y^2 - \quad y\mu + \mu^2}{2\sigma^2}$
$\textsc{Gamma}(\alpha, \beta)$	$\sum_{i=1}^{m} \boldsymbol{y}_i,\ \sum_{i=1}^{m} \ln(\boldsymbol{y}_i)$	$\alpha\ln(\beta) - \ln\Gamma(\alpha) - \quad y + (\alpha - 1 \quad \ln(y)$

Organizing Computation with Sufficient Statistics

- A leaf needs training labels $\boldsymbol{y}$ to select splits in training and answer queries
  - CART *summarize* $\boldsymbol{y}$ with *class frequencies* or *means*
  - "Incremental updates" make split search fast
  - CADET needs $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y})$ to train and $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y})$ to query
- *Sufficient statistics* $\mathrm{w}^{(m)}(\boldsymbol{y})$ w.r.t. $\mathcal{F}$ summarize $\boldsymbol{y}$ in $\mathcal{Y}^m$
  - Compute $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y})$ and minimize $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y})$ from $\mathrm{w}^{(m)}(\boldsymbol{y})$

Family $\mathcal{F}$	**Suff. Stat.** $\mathrm{w}^{(m)}(\boldsymbol{y})$	**Log Density** $\ln \boldsymbol{\rho}(y)$
$\textsc{Gaussian}(\mu, \sigma^2)$	$\sum_{i=1}^{m} \boldsymbol{y}_i, \sum_{i=1}^{m} \boldsymbol{y}_i^2$	$-\frac{1}{2}\ln(2\pi\sigma^2) - \frac{y^2 - \; y\mu + \mu^2}{2\sigma^2}$
$\textsc{Gamma}(\alpha, \beta)$	$\sum_{i=1}^{m} \boldsymbol{y}_i, \sum_{i=1}^{m} \ln(\boldsymbol{y}_i)$	$\alpha \ln(\beta) - \ln\Gamma(\alpha) - \; y + (\alpha - 1 \; \ln(y)$

- Always exist $\mathrm{w}(\cdot)$ for $\mathcal{F}$ in the *exponential class* s.t.

$$\mathrm{w}^{(m_L+m_R)}(\boldsymbol{y}_L \circ \boldsymbol{y}_R) \doteq \mathrm{w}^{(m_L)}(\boldsymbol{y}_L) + \mathrm{w}^{(m_R)}(\boldsymbol{y}_2)$$

Organizing Computation with Sufficient Statistics

- A leaf needs training labels $\boldsymbol{y}$ to select splits in training and answer queries
  - CART *summarize* $\boldsymbol{y}$ with *class frequencies* or *means*
  - "Incremental updates" make split search fast
  - CaDET needs $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y})$ to train and $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y})$ to query
- *Sufficient statistics* $\mathsf{w}^{(m)}(\boldsymbol{y})$ w.r.t. $\mathcal{F}$ summarize $\boldsymbol{y}$ in $\mathcal{Y}^m$
  - Compute $\mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y})$ and minimize $\mathrm{I}_{\mathcal{F}}(\boldsymbol{y})$ from $\mathsf{w}^{(m)}(\boldsymbol{y})$

Family $\mathcal{F}$	**Suff. Stat.** $\mathsf{w}^{(m)}(\boldsymbol{y})$	**Log Density** $\ln \boldsymbol{\rho}(y)$
$\textsc{Gaussian}(\mu, \sigma^2)$	$\sum_{i=1}^{m} \boldsymbol{y}_i, \sum_{i=1}^{m} \boldsymbol{y}_i^2$	$-\frac{1}{2}\ln(2\pi\sigma^2) - \frac{y^2 - \quad y\mu + \mu^2}{2\sigma^2}$
$\textsc{Gamma}(\alpha, \beta)$	$\sum_{i=1}^{m} \boldsymbol{y}_i, \sum_{i=1}^{m} \ln(\boldsymbol{y}_i)$	$\alpha \ln(\beta) - \ln\Gamma(\alpha) - \quad y + (\alpha - \quad \ln(y)$

  - Always exist $\mathsf{w}(\cdot)$ for $\mathcal{F}$ in the *exponential class* s.t.

$$\mathsf{w}^{(m_L+m_R)}(\boldsymbol{y}_L \circ \boldsymbol{y}_R) \doteq \mathsf{w}^{(m_L)}(\boldsymbol{y}_L) + \mathsf{w}^{(m_R)}(\boldsymbol{y}_2)$$

- Time to compute impurity reduction:
  - With $\mathsf{w}(\cdot)$: $\mathcal{O}(1)$ amortized time
  - Without $\mathsf{w}(\cdot)$: $\mathcal{O}(m)$ time

CaDET as a Unifying Framework

- Cross-entropy impurity criterion $\mathrm{I}_{\mathcal{F}}(\cdot)$ tailored to $\mathcal{F}$

CaDET as a Unifying Framework

- Cross-entropy impurity criterion $\mathrm{I}_{\mathcal{F}}(\cdot)$ tailored to $\mathcal{F}$

$\mathcal{F}$	$\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}) \equiv$	Tree Model
$\textsc{Gaussian}(\cdot, 1)$	$\mathrm{I}_{\mathrm{MSE}}(\boldsymbol{y})$	Regression Tree

$\textsc{CaDET}$ as a Unifying Framework

- Cross-entropy impurity criterion $\mathrm{I}_{\mathcal{F}}(\cdot)$ tailored to $\mathcal{F}$

$\mathcal{F}$	$\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}) \equiv$	Tree Model
$\textsc{Gaussian}(\cdot, 1)$	$\mathrm{I}_{\mathrm{MSE}}(\boldsymbol{y})$	Regression Tree
$\textsc{Categorical}(\cdot)$	$\mathrm{I}_{\mathrm{H}}(\boldsymbol{y})$	Information-Gain Tree

$\textsc{CaDET}$ as a Unifying Framework

- Cross-entropy impurity criterion $\mathrm{I}_{\mathcal{F}}(\cdot)$ tailored to $\mathcal{F}$

$\mathcal{F}$	$\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}) \equiv$	Tree Model
$\textsc{Gaussian}(\cdot, 1)$	$\mathrm{I}_{\mathrm{MSE}}(\boldsymbol{y})$	Regression Tree
$\textsc{Categorical}(\cdot)$	$\mathrm{I}_{\mathrm{H}}(\boldsymbol{y})$	Information-Gain Tree

- We've reconstructed two models from the 80s...

CADET as a Unifying Framework

- Cross-entropy impurity criterion $\mathrm{I}_{\mathcal{F}}(\cdot)$ tailored to $\mathcal{F}$

$\mathcal{F}$	$\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}) \equiv$	Tree Model
$\textsc{Gaussian}(\cdot, 1)$	$\mathrm{I}_{\mathrm{MSE}}(\boldsymbol{y})$	Regression Tree
$\textsc{Categorical}(\cdot)$	$\mathrm{I}_{\mathrm{H}}(\boldsymbol{y})$	Information-Gain Tree

- We've reconstructed two models from the 80s. . .
- Two underlying philosophies for split selection

(1) *Maximum likelihood*, maximize *sum-log-likelihood* of $\boldsymbol{y}$ $\quad \mathrm{I}_{\mathcal{F}}(\boldsymbol{y}) = \mathrm{H}(\boldsymbol{y}, \mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y}))$

(2) *Minimax entropy*, minimize *uncertainty* of predictions $\quad \mathrm{I}_{\mathrm{H},\mathcal{F}}(\boldsymbol{y}) = \mathrm{H}(\boldsymbol{y})$

$\textsc{CaDET}$ as a Unifying Framework

- Cross-entropy impurity criterion $\mathrm{I}_{\mathcal{F}}(\cdot)$ tailored to $\mathcal{F}$

$\mathcal{F}$	$\mathrm{I}_{\mathcal{F}}(\boldsymbol{y}) \equiv$	Tree Model
$\textsc{Gaussian}(\cdot, 1)$	$\mathrm{I}_{\mathrm{MSE}}(\boldsymbol{y})$	Regression Tree
$\textsc{Categorical}(\cdot)$	$\mathrm{I}_{\mathrm{H}}(\boldsymbol{y})$	Information-Gain Tree

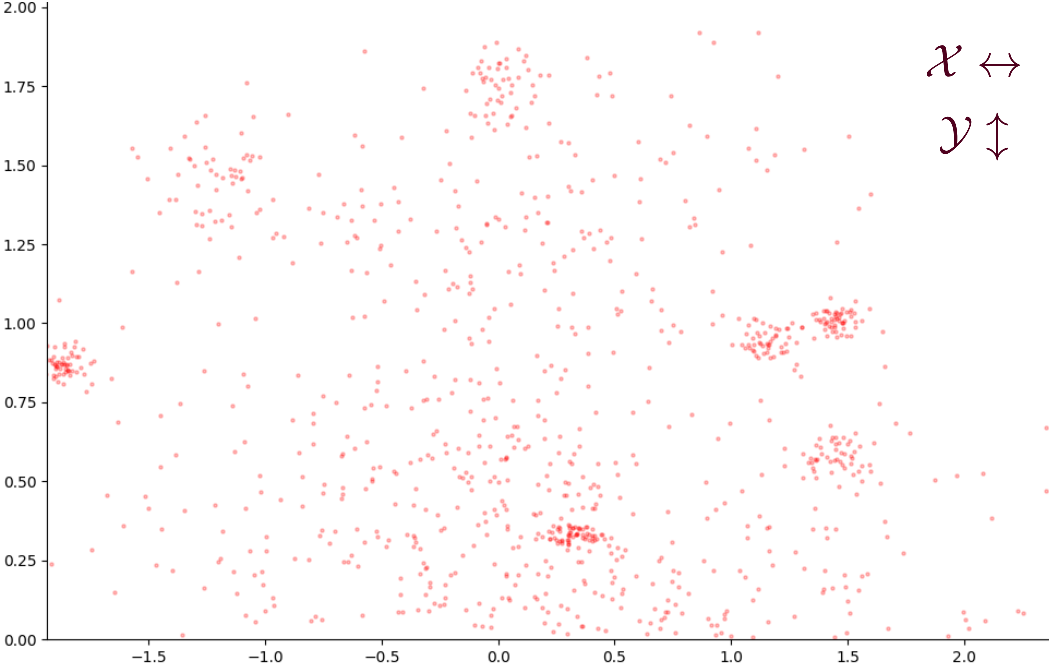
- We've reconstructed two models from the 80s...
- Two underlying philosophies for split selection

(1) *Maximum likelihood*, maximize *sum-log-likelihood* of $\boldsymbol{y}$ $\quad \mathrm{I}_{\mathcal{F}}(\boldsymbol{y}) = \mathrm{H}(\boldsymbol{y}, \mathrm{MLE}_{\mathcal{F}}(\boldsymbol{y}))$

(2) *Minimax entropy*, minimize *uncertainty* of predictions $\quad \mathrm{I}_{\mathrm{H},\mathcal{F}}(\boldsymbol{y}) = \mathrm{H}(\boldsymbol{y})$

- Lemma 1: conditions on $\mathcal{F}$ under which $\mathrm{I}_{\mathcal{F}}(\cdot) = \mathrm{I}_{\mathrm{H},\mathcal{F}}(\cdot)$

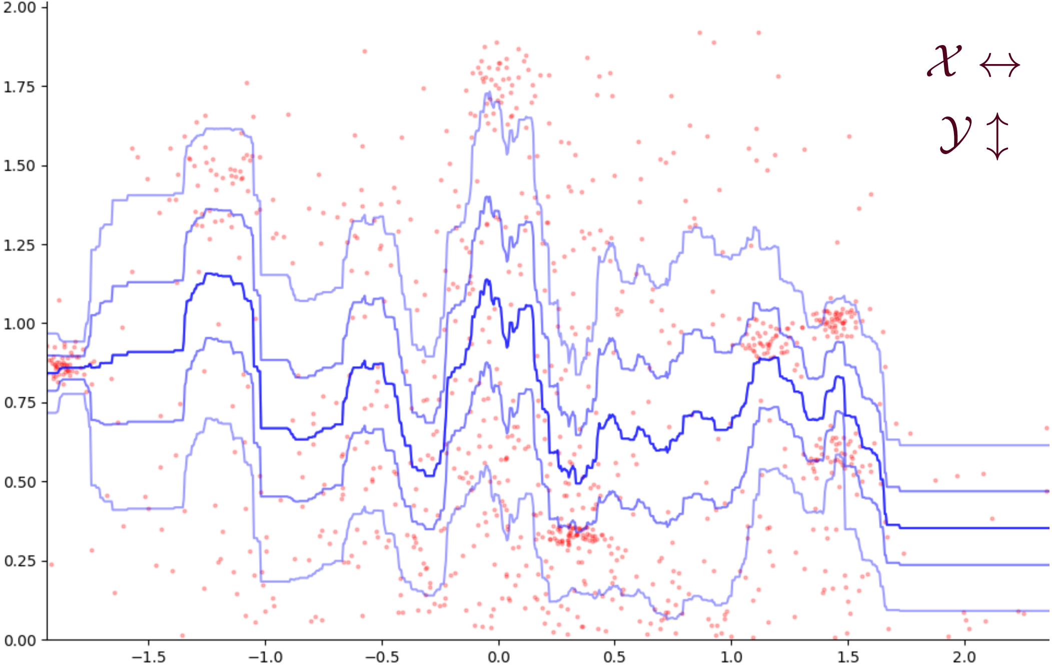
Visualizing Gaussian-CaDET Forests



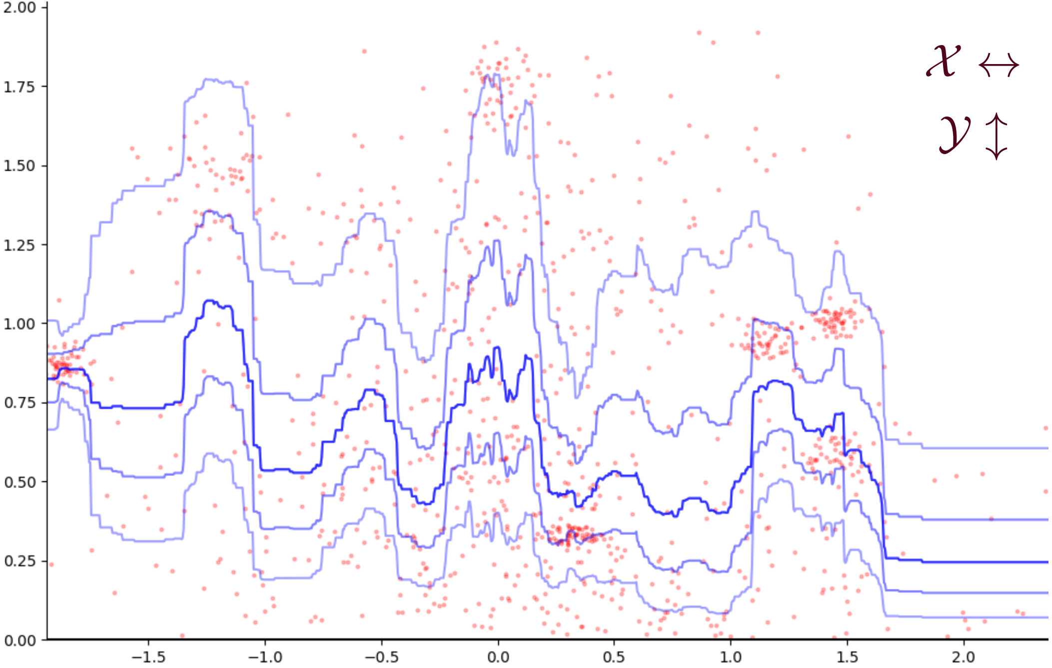
$\mathcal{X} \leftrightarrow$

$\mathcal{Y} \updownarrow$

Visualizing Gaussian-CaDET Forests



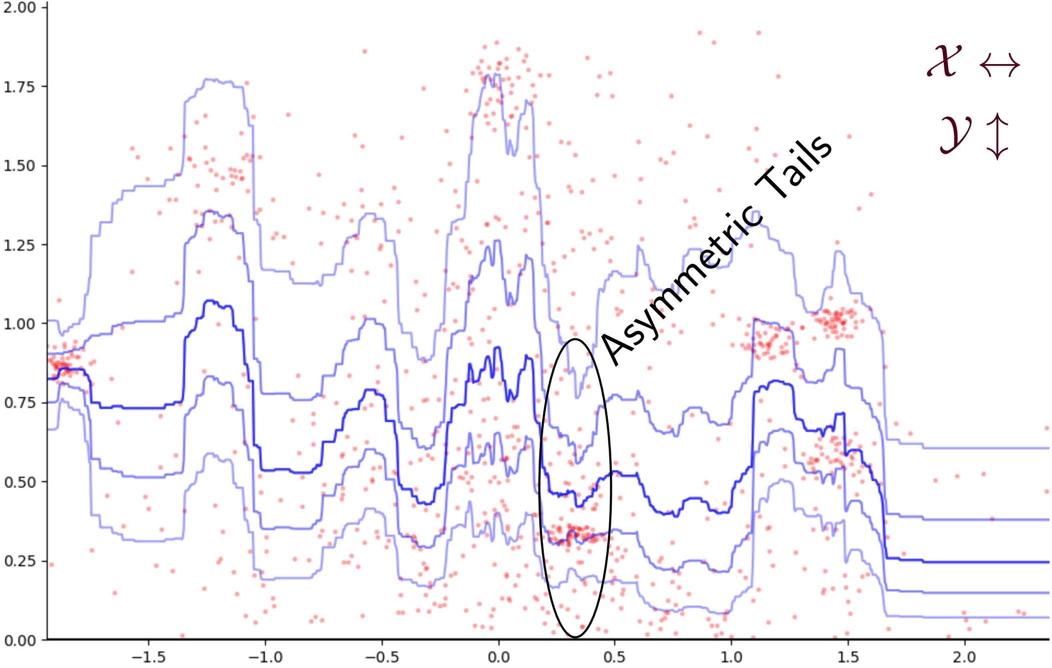
Visualizing Gamma-CaDET Forests



$\mathcal{X} \leftrightarrow$

$\mathcal{Y} \updownarrow$

Visualizing Gamma-CADET Forests



A Brief Recapitulation

- CaDET: simple, interpretable, parametric CDE trees
  - Nonparametric methods uninterpretable
- Efficient training, query, and storage costs with
  - Additive sufficient statistics
  - Efficiency matches CART
  - $\Omega(m)$ speedup over nonparametric CDE trees
- Generalize existing tree methods
  - Information-gain classification trees
  - MSE regression trees