Conditional Density Estimation with Random Forests and Sufficient Statistics

Cyrus Cousins
Brown University

and

Matteo Riondato
Two Sigma Labs

FouLaRD'18 – September 14, 2018

The problem

$(X, Y) \sim \Pi, \quad X \in \mathcal{D}, Y \in \mathbb{R}$

Given $(X_1, Y_1), \ldots, (X_\ell, Y_\ell) \overset{\text{iid}}{\sim} \Pi$,

Regression: estimate $\mathbb{E}_\Pi[Y \mid X]$

The problem

$(X, Y) \sim \Pi, \quad X \in \mathcal{D}, Y \in \mathbb{R}$

Given $(X_1, Y_1), \ldots, (X_\ell, Y_\ell) \overset{\text{iid}}{\sim} \Pi$,

Regression: estimate $\mathbb{E}_\Pi[Y \mid X]$

Conditional Density Estimation (CDE): estimate the *conditional PDF* (CD) $f(y \mid X)$

($\Pr(a \leq Y \leq b|X) = \int_a^b f(y \mid X) dy$)

CDE is *more nuanced, informative*:

detect skewness & multimodality, compute *quantiles*, ...

Our contribution

A *parametric* variant of *decision trees* and *random forests* for CDE.

Key properties:

- Fast to train, as it uses *cross-entropy* as *impurity criterion*
- Fast to query, as it stores *sufficient statistics* at the leaves to compute the CD.
- Small(er) storage requirements (than previous work)

Our contribution

A *parametric* variant of *decision trees* and *random forests* for CDE.

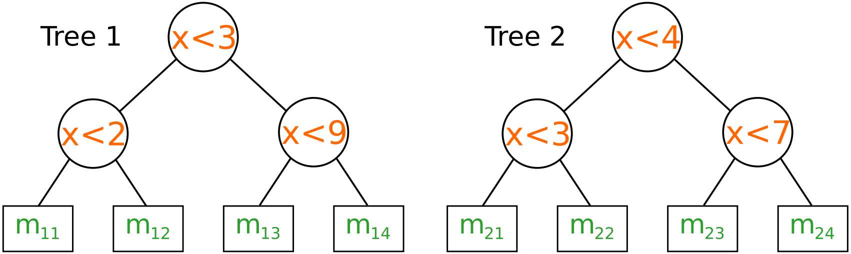
KEY PROPERTIES:

- Fast to train, as it uses *cross-entropy* as *impurity criterion*
- Fast to query, as it stores *sufficient statistics* at the leaves to compute the CD.
- Small(er) storage requirements (than previous work)

LIMITATION: (for now)

- Only estimate CDs from *parametric* families of distributions with *sufficient statistics*

Random forests

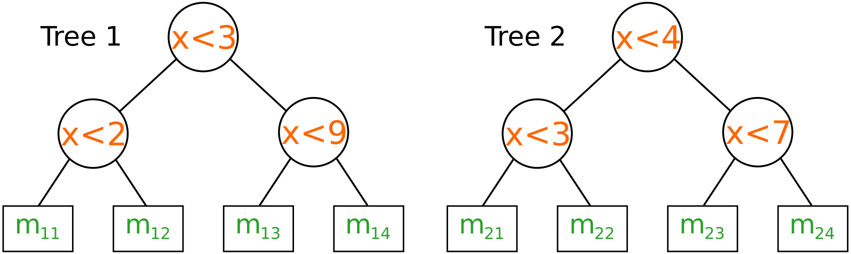


Split the feature space in *hyperrectangles*

Diversity: Each tree T_i is built using a *resampled* $\mathcal{S}_i$ from training set $\mathcal{S}$.

m_{ij}: average of Y values of subset of $\mathcal{S}$ mapped to leaf j of tree i.

Random forests



Split the feature space in *hyperrectangles*

Diversity: Each tree T_i is built using a *resampled* $\mathcal{S}_i$ from training set $\mathcal{S}$.

m_{ij}: average of Y values of subset of $\mathcal{S}$ mapped to leaf j of tree i.

Prediction for $x \in \mathcal{D} = \mathbb{R}$: $\frac{1}{t}\sum_{i=1}^{t} m_{i,\phi_i(x)}$,
$\phi_i(x) =$ leaf of T_i that x is mapped to.

Why random forests?

Avoid decision tree tendency to *overfit* (or have high variance)

By having *multiple trees*, the variance is controlled

Very interpretable: hyperrectangles, *weighted nearest neighbor*

Can rank features using out-of-bag estimates

Fast to train, fast to query

Very effective in practice, especially if the data has some structure

Building decision trees

A (resampled) $\mathcal{S}_i$, it is *recursively split* until a *stopping criterion* is satisfied.

At each step, a *single feature* f is chosen for the split.

To identify the split:

Given $Q \subseteq \mathcal{S}_i$ find b^* in the domain of f s.t. b^* minimizes an *impurity criterion* $h(b)$

E.g., for regression:

Let $L_b = \{(X,Y) \in Q \ : \ X_f < b\}$ and $U_b = \{(X,Y) \in Q \ : \ X_f \geq b\}$

$$h(b) = \sum_{(X,Y)\in L_b} \left(Y - \frac{1}{|L_b|} \sum_{(X,Y)\in L_b} Y \right)^2 + \sum_{(X,Y)\in U_b} \left(Y - \frac{1}{|U_b|} \sum_{(X,Y)\in U_b} Y \right)^2$$

Defining variants of random forests

Ingredients:

1. *Impurity criterion* h

2. ...

Defining variants of random forests

INGREDIENTS:

1. *Impurity criterion* h

2. ...

(Each leaf ℓ_{ij} can store $Z_{ij} = \{(x, y) \in \mathcal{S}_i \ : \ \phi_i(x) = j\}$, rather than a single value m_{ij})

A *prediction function* $\mathsf{m} \ : \ 2^{\mathcal{D}} \times \cdots \times 2^{\mathcal{D}} \to \mathbb{R}$:

$$\text{prediction for } x = \mathsf{m}\left(Z_{1\phi_1(x)}, \ldots, Z_{k\phi_1(x)}\right)$$

Random forests for CDE – Previous approaches

1. Impurity criterion: Same as for *regression*

2. Prediction function: *Kernel estimation*:

   Let $Z(x) = \bigcup_{i=1}^{t} Z_{i\phi_i(x)}$

$$\hat{f}(y|x) = \frac{1}{|Z(x)|} \sum_{(X,Y)\in Z(x)} \mathsf{K}(Y - y)$$

Random forests for CDE – Previous approaches

1. Impurity criterion: Same as for *regression* Not appropriate for *CDE*, bwidth choice

2. Prediction function: *Kernel estimation*:

   Let $Z(x) = \bigcup_{i=1}^{t} Z_{i\phi_i(x)}$

$$\hat{f}(y|x) = \frac{1}{|Z(x)|} \sum_{(X,Y)\in Z(x)} \mathsf{K}(Y - y)$$

Random forests for CDE – Previous approaches

1. Impurity criterion: Same as for *regression* Not appropriate for *CDE*, bwidth choice

2. Prediction function: *Kernel estimation*:

   Let $Z(x) = \bigcup_{i=1}^{t} Z_{i\phi_i(x)}$

$$\hat{f}(y|x) = \frac{1}{|Z(x)|} \sum_{(X,Y)\in Z(x)} \mathsf{K}(Y - y)$$

Expensive in terms of *storage* and *evaluation time*

Sufficient statistics

$\mathcal{F}$: a *parametric family* of probability distributions:

$$f \in \mathcal{F} = f_\theta, \theta \in \Theta$$

A: sample from f_θ.

$\mathrm{s}(A) \in \mathcal{R}^w$ is a *sufficient statistic* for $\mathcal{F}$ iff

$$\Pr(\theta \mid A, \mathrm{s}(A)) = \Pr(\theta \mid \mathrm{s}(A))$$

E.g., for $\mathcal{F} =$ Gaussians, $\mathrm{s}(A) = (\mathrm{avg}(A), \mathrm{var}(A))$ or $\mathrm{s}(A) = (|A|, \mathrm{sum}(A), \mathrm{sumsq}(A))$

Sufficient statistics

$\mathcal{F}$: a *parametric family* of probability distributions:

$$f \in \mathcal{F} = f_\theta, \theta \in \Theta$$

A: sample from f_θ.

$\mathrm{s}(A) \in \mathcal{R}^w$ is a *sufficient statistic* for $\mathcal{F}$ iff

$$\Pr(\theta \mid A, \mathrm{s}(A)) = \Pr(\theta \mid \mathrm{s}(A))$$

E.g., for $\mathcal{F} =$ Gaussians, $\mathrm{s}(A) = (\mathrm{avg}(A), \mathrm{var}(A))$ or $\mathrm{s}(A) = (|A|, \mathrm{sum}(A), \mathrm{sumsq}(A))$

Not all $\mathcal{F}$ have s, but $\mathcal{F}$ of the *exponential family* have s s.t.

$$\mathrm{s}(A \cup B) = \mathrm{s}(A) + \mathrm{s}(B)$$

Random forests for CDE – Our method

$\mathcal{F}$: *user-specified* parametric (exponential) family of *probability distributions*.

1. Impurity criterion: *cross-entropy* :

   (For discrete distribs p and q, $\mathrm{crentr}(p, q) = -\sum_{x \in X} p(x) \log(q(x))$)

   For $b \in \mathbb{R}$, let $L_b = \{(X, Y) \in Q \ : \ X_f < b\}$ and $U_b = \{(X, Y) \in Q \ : \ X_f \geq b\}$.

   Fit $f_{L_b}, f_{U_b} \in \mathcal{F}$ (from $\mathrm{s}(L_b)$, $\mathrm{s}(U_b)$)

$$h(b) = -\sum_{(X,Y) \in L_b} \log(f_{L_b}(Y)) - \sum_{(X,Y) \in U_b} \log(f_{L_b}(Y))$$

Random forests for CDE – Our method

$\mathcal{F}$: *user-specified* parametric (exponential) family of *probability distributions*.

1. Impurity criterion: *cross-entropy* :

   (For discrete distribs p and q, $\mathrm{crentr}(p,q) = -\sum_{x \in X} p(x) \log(q(x))$)

   For $b \in \mathbb{R}$, let $L_b = \{(X,Y) \in Q \ : \ X_f < b\}$ and $U_b = \{(X,Y) \in Q \ : \ X_f \geq b\}$.

   Fit $f_{L_b}, f_{U_b} \in \mathcal{F}$ (from $\mathrm{s}(L_b)$, $\mathrm{s}(U_b)$)

$$h(b) = -\sum_{(X,Y)\in L_b} \log(f_{L_b}(Y)) - \sum_{(X,Y)\in U_b} \log(f_{L_b}(Y))$$

Specific to CDE!

Random forests for CDE – Our method

2. Prediction function: *fit using sufficient statistics*:
(Store $s_{ij} = \mathsf{s}(Z_{ij})$ in leaf $\ell_{i,j}$). When a point $x \in \mathcal{D}$ arrives, compute

$$s_x = \mathsf{s}(Z(x)) = \mathsf{s}\left(\bigcup_{i=1}^{t} Z_{i\phi_i(x)}\right) = \sum_{i=1}^{t} s_{i\phi_i(x)}$$

Fit $f \in \mathcal{F}$ using s_x and return it

Random forests for CDE – Our method

2. Prediction function: *fit using sufficient statistics*:
(Store $s_{ij} = \mathsf{s}(Z_{ij})$ in leaf $\ell_{i,j}$). When a point $x \in \mathcal{D}$ arrives, compute

$$s_x = \mathsf{s}(Z(x)) = \mathsf{s}\left(\bigcup_{i=1}^{t} Z_{i\phi_i(x)}\right) = \sum_{i=1}^{t} s_{i\phi_i(x)}$$

Fit $f \in \mathcal{F}$ using s_x and return it 🙂 Small memory, fast computation

Preliminary Experiments

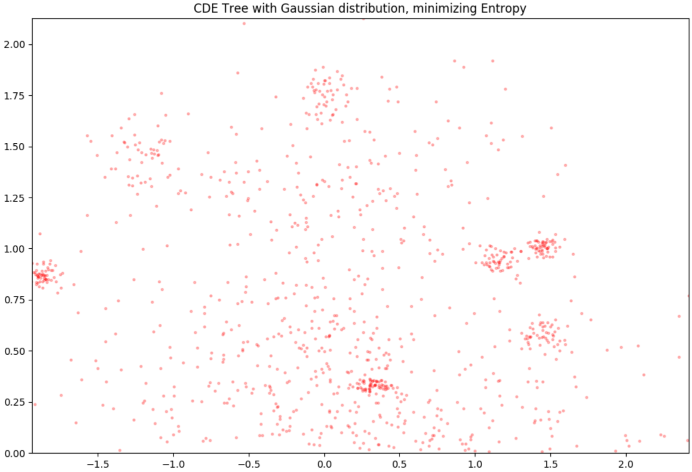
GOALS: does any of this make any sense?

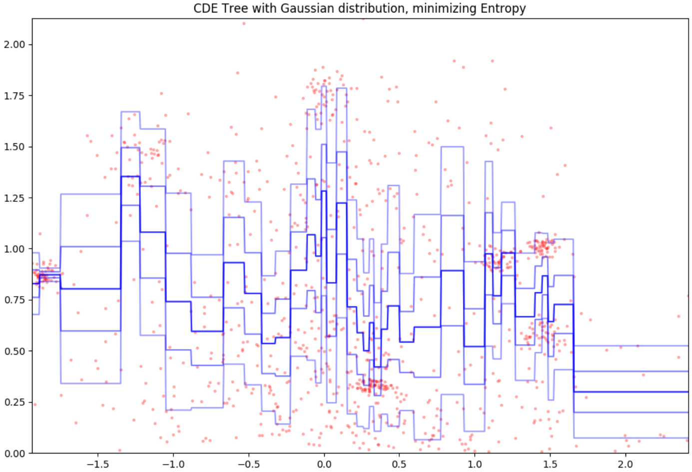
Preliminary Experiments

GOALS: does any of this make any sense?

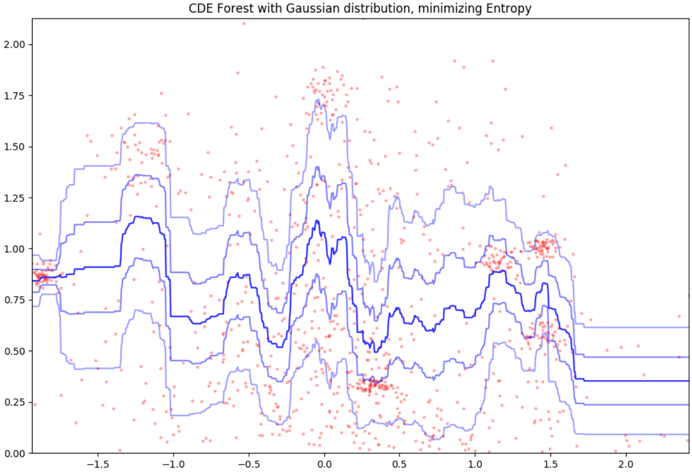
Data generating process:

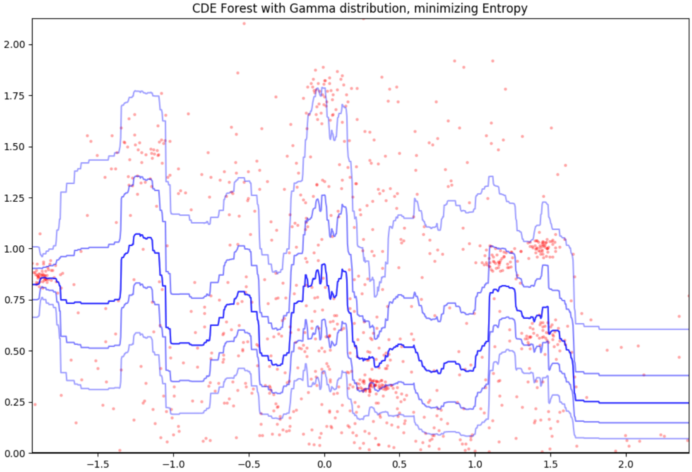
1. Choose centers $c_1, \ldots, c_\ell$ from standard bivariate Gaussian
2. Sample points (x, y) from bivariate Gaussian mixture with centers c_i with diagonal covariance matrices
3. Exponentially transform $y = e^y$


CDE Tree with Gaussian distribution, minimizing Entropy
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0



CDE Tree with Gaussian distribution, minimizing Entropy
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


CDE Forest with Gaussian distribution, minimizing Entropy
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0



CDE Forest with Gamma distribution, minimizing Entropy
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

Future directions

1) *Relax parametrization* requirement!

IDEA: Generalized Method of Moments 

1) At ℓ_{ij}, store *many statistics* of Z_{ij}, and a *small sample* of it

2) For prediction, use stats and sample to fit a *semi-parametric* model under *moment conditions*

2) Use *soft splits* to allow for "gentler" changes of values

Recap

A method for *conditional density estimation* using *random forests*

It uses *cross-entropy* as impurity criterion for tree-growth

It stores *sufficient statistics* at tree leaves for inference

Fast to build, fast to query, small memory footprint

Cyrus Cousins
ccousins@cs.brown.edu

Matteo Riondato
@teorionda
http://matteo.rionda.to

Image credits

Figure on page 1: Public domain https://libreshot.com/foggy-forest/