Explainable Cautious Classifiers

Haifei ZHANG

Seminar - LHC

October 17, 2024

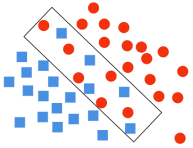
Outline

Uncertainty

Classifier $\boldsymbol{h}$ is trained with pictures of dog and cat.

$$\mathbb{P}(\text{dog} \,|\, \boldsymbol{x}, \boldsymbol{h}) = 0.51$$

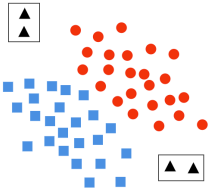
$$\mathbb{P}(\text{cat} \,|\, \boldsymbol{x}, \boldsymbol{h}) = 0.49$$



Aleatory uncertainty

$$\mathbb{P}(\text{dog} \,|\, \boldsymbol{x}, \boldsymbol{h}) = 0.85$$

$$\mathbb{P}(\text{cat} \,|\, \boldsymbol{x}, \boldsymbol{h}) = 0.15$$

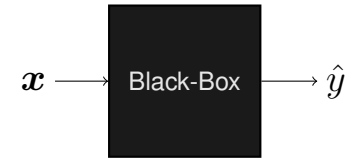


Epistemic uncertainty

Determinate predictions may be unreliable.

Alternative: set-valued predictions $\boldsymbol{h}(\boldsymbol{x}) \subseteq \Omega = \{c_1, \ldots, c_K\}$.

Explainability



We know the prediction $\hat{y}$ is made for $\boldsymbol{x}$.

But we don't know

- why the prediction $\hat{y}$ is made;
- how to get another desired prediction different from $\hat{y}$.

eXplainable AI: we need explanations.[1]

[1] A. B. Arrieta, N. Díaz-Rodríguez, D. Ser, *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.

Objectives

1. Propose a new cautious classifier: **cautious random forests**

2. Provide explanations for indeterminate predictions: **counterfactual examples**

Outline

Cautious classification

- Input space: $\boldsymbol{\mathcal{X}} = \{\mathcal{X}^1, \ldots, \mathcal{X}^M\}$
- Output space: $\mathcal{Y} = \mathscr{P}(\Omega)$, where $\Omega = \{c_1, \ldots, c_K\}$ are class labels
- Cautious classifier $\boldsymbol{h}$: $\boldsymbol{\mathcal{X}} \to \mathscr{P}(\Omega)$
- Set-valued prediction: $\hat{Y} = \boldsymbol{h}(\boldsymbol{x}) \subseteq \Omega,\ \forall \boldsymbol{x} \in \boldsymbol{\mathcal{X}}$

Cautious classification

- Input space: $\boldsymbol{\mathcal{X}} = \{\mathcal{X}^1, \ldots, \mathcal{X}^M\}$
- Output space: $\mathcal{Y} = \mathscr{P}(\Omega)$, where $\Omega = \{c_1, \ldots, c_K\}$ are class labels
- Cautious classifier $\boldsymbol{h}$: $\boldsymbol{\mathcal{X}} \to \mathscr{P}(\Omega)$
- Set-valued prediction: $\hat{Y} = \boldsymbol{h}(\boldsymbol{x}) \subseteq \Omega,\ \forall \boldsymbol{x} \in \boldsymbol{\mathcal{X}}$

$\boldsymbol{h}(\boldsymbol{x}) = \{\text{dog}, \text{cat}\}$

$\boldsymbol{h}(\boldsymbol{x}) = \neg\{\text{dog}, \text{cat}\} = \emptyset$

Cautious classification

- Input space: $\boldsymbol{\mathcal{X}} = \{\mathcal{X}^1, \ldots, \mathcal{X}^M\}$
- Output space: $\mathcal{Y} = \mathscr{P}(\Omega)$, where $\Omega = \{c_1, \ldots, c_K\}$ are class labels
- Cautious classifier $\boldsymbol{h}$: $\boldsymbol{\mathcal{X}} \to \mathscr{P}(\Omega)$
- Set-valued prediction: $\hat{Y} = \boldsymbol{h}(\boldsymbol{x}) \subseteq \Omega,\ \forall \boldsymbol{x} \in \boldsymbol{\mathcal{X}}$

$\boldsymbol{h}(\boldsymbol{x}) = \{\text{dog}, \text{cat}\}$

$\boldsymbol{h}(\boldsymbol{x}) = \neg\{\text{dog}, \text{cat}\} = \emptyset$

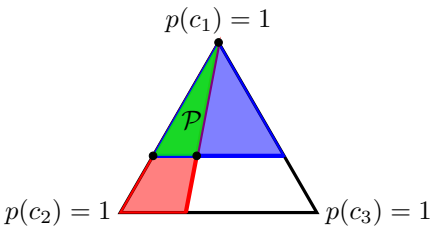
Strategies to make set-valued predictions:

1. Create a partial order among all single classes (precise assignments)
2. Create a complete order among all subsets of classes (partial assignments)

Imprecise probabilities

Credal set[2]:

$$p(c_1) \geq \frac{1}{3}, \quad p(c_2) - 2p(c_3) \geq 0,$$



Lower and upper expected utilities:

$$\underline{\mathbb{E}}_{\mathcal{P}}(c_i, \mathbf{U}) = \min_{\boldsymbol{p} \in \mathcal{P}} \mathbb{E}_{\boldsymbol{p}}(c_i, \mathbf{U}) = \min_{\boldsymbol{p} \in \mathcal{P}} \sum_{k=1}^{K} p_k \cdot u_{ik} \tag{1}$$

$$\overline{\mathbb{E}}_{\mathcal{P}}(c_i, \mathbf{U}) = \max_{\boldsymbol{p} \in \mathcal{P}} \mathbb{E}_{\boldsymbol{p}}(c_i, \mathbf{U}) = \max_{\boldsymbol{p} \in \mathcal{P}} \sum_{k=1}^{K} p_k \cdot u_{ik} \tag{2}$$

where $\mathcal{P}$ is a credal set on $\Omega = \{c_1, \ldots, c_K\}$ and $\mathbf{U}$ a utility matrix.

[2] I. Levi, *The enterprise of knowledge: An essay on knowledge, credal probability, and chance.* MIT press, 1980.

Decision-making with imprecise probabilities[3]

Strong dominance:

$$c_i \succcurlyeq_{sd} c_j, \text{ if } \underline{\mathbb{E}}_{\mathcal{P}}(c_i, \mathbf{U}) \geq \overline{\mathbb{E}}_{\mathcal{P}}(c_j, \mathbf{U}) \tag{3}$$

Weak dominance:

$$c_i \succcurlyeq_{wd} c_j, \text{ if } \underline{\mathbb{E}}_{\mathcal{P}}(c_i, \mathbf{U}) \geq \underline{\mathbb{E}}_{\mathcal{P}}(c_j, \mathbf{U}) \text{ and } \overline{\mathbb{E}}_{\mathcal{P}}(c_i, \mathbf{U}) \geq \overline{\mathbb{E}}_{\mathcal{P}}(c_j, \mathbf{U}) \tag{4}$$

Maximality:

$$c_i \succcurlyeq_{max} c_j, \text{ if } \underline{\mathbb{E}}_{\mathcal{P}}(c_i - c_j, \mathbf{U}) \geq 0 \tag{5}$$

E-admissibility:

$$c_i \text{ is E-admissible, if } \exists \boldsymbol{p} \in \mathcal{P}, \text{ s.t. } \forall c_j \in \Omega, \ \mathbb{E}_{\boldsymbol{p}}(c_i, \mathbf{U}) \geq \mathbb{E}_{\boldsymbol{p}}(c_j, \mathbf{U}) \tag{6}$$

[3] P. Walley, *Statistical reasoning with imprecise probabilities*, vol. 42. Springer, 1991.

Example: imprecise decision trees

Setting:

- leaf with class counts $(n_1, \ldots, n_k, \ldots, n_K)$
- $N = \sum n_k$: the total number of samples

Imprecise Dirichlet model (IDM)[4]:

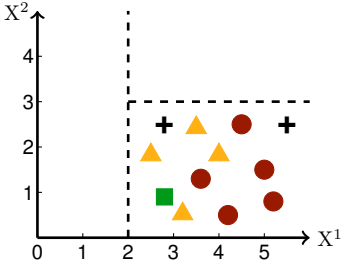
$$\mathcal{I}_k = \left[\underline{p}_k, \overline{p}_k\right] = \left[\frac{n_k}{N+s}, \frac{n_k+s}{N+s}\right], \; k = 1, \ldots, K \quad (7)$$

where s is interpreted as the number of virtual samples

Strong dominance:

retain the set of non-dominated classes

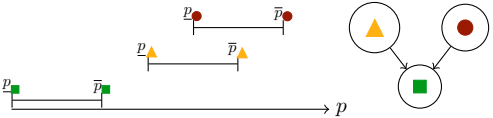
$$\hat{Y} = \{c_k : \nexists\, c_j \in \Omega \text{ s.t. } \underline{p}_j \geq \overline{p}_k\}$$



$$n(\blacksquare) = 1, \; n(\blacktriangle) = 4, \; n(\bullet) = 5$$

$$N = 10, \; s = 2$$

$$\mathcal{I}_{\blacksquare} = \left[\tfrac{1}{12}, \tfrac{1}{4}\right], \mathcal{I}_{\blacktriangle} = \left[\tfrac{1}{3}, \tfrac{1}{2}\right], \mathcal{I}_{\bullet} = \left[\tfrac{5}{12}, \tfrac{7}{12}\right]$$



[4] P. Walley, "Inferences from Multinomial Data: Learning About a Bag of Marbles," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, pp. 3–34, 1996.

Belief functions[5]

- Frame of discernment: $\Omega = \{c_1, \ldots, c_K\}$
- Mass function $m : 2^\Omega \to [0, 1]$, $m(\emptyset) = 0$ and $\sum_{A \subseteq \Omega} m(A) = 1$
- Focal element: $A \subseteq \Omega$ such that $m(A) > 0$
- Belief degree: total support

$$Bel(A) = \sum_{B \subseteq A} m(B), \ \forall A \subseteq \Omega \tag{8}$$

- Plausibility degree: potential support

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B), \ \forall A \subseteq \Omega \tag{9}$$

[5] A. P. Dempster, "Upper and Lower Probabilities Induced by a Multivalued Mapping," *The Annals of Mathematical Statistics*, vol. 38, pp. 325–339, 1967.

Cautious decision-making with belief functions[6]

The lower and upper expected utilities of taking $A \subseteq \Omega$ as prediction:

$$\underline{EU}(m, A, \mathbf{U}) = \sum_{B \subseteq \Omega} m(B) \min_{c_k \in B} u_{Ak} \quad \text{and} \quad \overline{EU}(m, A, \mathbf{U}) = \sum_{B \subseteq \Omega} m(B) \max_{c_k \in B} u_{Ak} \tag{10}$$

where u_{Ak} is the utility when A is taken as prediction and c_k is the actual class.

If 0/1 loss is considered, $\underline{EU}(m, A, \mathbf{U}) = Bel(A)$ and $\overline{EU}(m, A, \mathbf{U}) = Pl(A)$.

[6] T. Denoeux, "Decision-making with belief functions: a review," *International Journal of Approximate Reasoning*, vol. 109, pp. 87–110, 2019.

Cautious decision-making with belief functions[6]

The lower and upper expected utilities of taking $A \subseteq \Omega$ as prediction:

$$\underline{EU}(m, A, \mathbf{U}) = \sum_{B \subseteq \Omega} m(B) \min_{c_k \in B} u_{Ak} \quad \text{and} \quad \overline{EU}(m, A, \mathbf{U}) = \sum_{B \subseteq \Omega} m(B) \max_{c_k \in B} u_{Ak} \tag{10}$$

where u_{Ak} is the utility when A is taken as prediction and c_k is the actual class.

If 0/1 loss is considered, $\underline{EU}(m, A, \mathbf{U}) = Bel(A)$ and $\overline{EU}(m, A, \mathbf{U}) = Pl(A)$.

1. Partial order over precise assignments ($c \in \Omega$), e.g., via strong dominance:

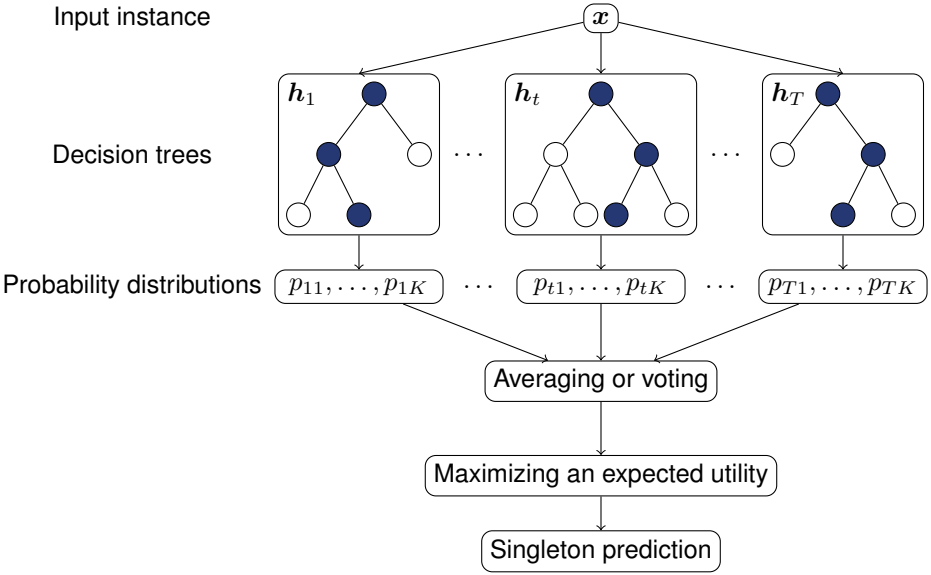
$$\hat{Y} = \{c_k \colon \nexists\, c_j \in \Omega \text{ s.t. } \underline{EU}(\{c_j\}) \geq \overline{EU}(\{c_k\})\} \tag{11}$$

2. Complete order over partial assignments ($A \subseteq \Omega$):

$$\hat{Y} = \arg\max_{A \subseteq \Omega} \underline{EU}(m, A, \mathbf{U}) \text{ or } \hat{Y} = \arg\max_{A \subseteq \Omega} \overline{EU}(m, A, \mathbf{U}) \tag{12}$$

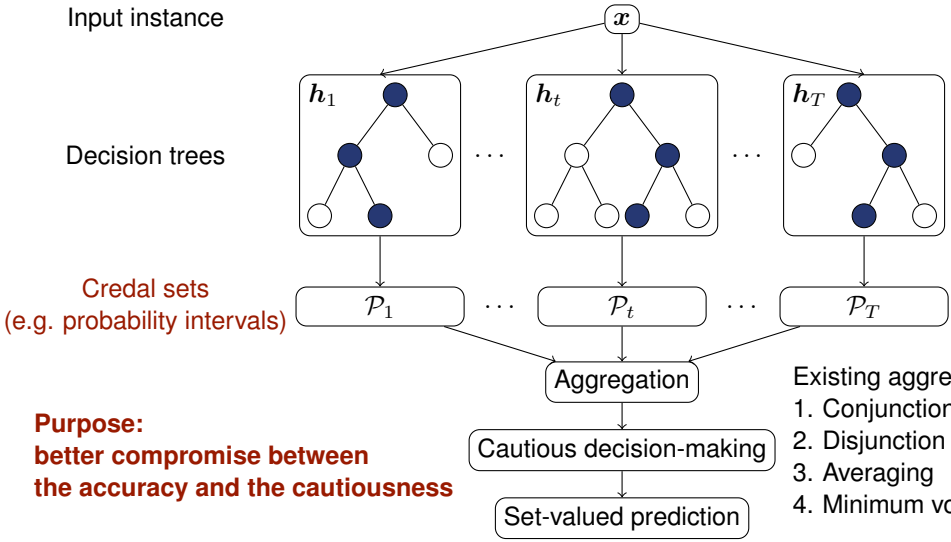
[6] T. Denoeux, "Decision-making with belief functions: a review," *International Journal of Approximate Reasoning*, vol. 109, pp. 87–110, 2019.

Random forest



Laboratoire Hubert Curien
UMR • CNRS • 5516 • Saint-Étienne

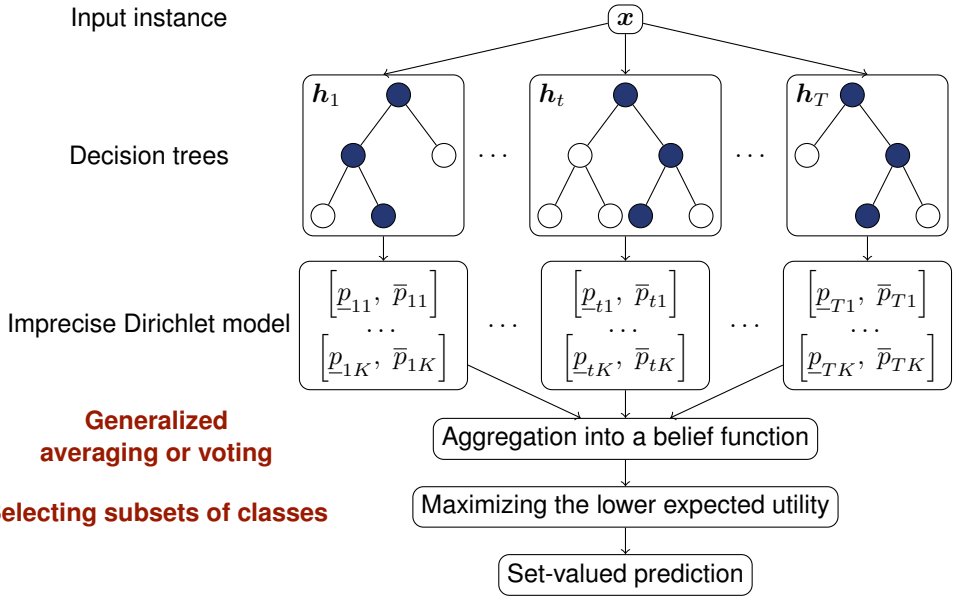
Extend random forest to a cautious one



Purpose:
better compromise between the accuracy and the cautiousness

Existing aggregation methods:
1. Conjunction
2. Disjunction
3. Averaging
4. Minimum vote against

Cautious random forest



Cautious random forests

Setting

- $\Omega = \{c_1, \ldots, c_K\},\ K > 2$
- Probability intervals: $\{\mathcal{I}_{tk} = \left[\underline{p}_{tk}, \overline{p}_{tk}\right],\ k = 1, \ldots, K,\ t = 1, \ldots, T\}$

Interpretation

Probability intervals provided by each tree are turned into a quasi-Bayesian mass function

$$m_t(\{c_k\}) = \underline{p}_{tk},\ k = 1, \ldots, K, \quad m_t(\Omega) = 1 - \sum_{k=1}^{K} m_t(\{c_k\}) \tag{13}$$

Problems

1. **How to aggregate them into a single mass function?**
2. **How to make cautious predictions based on it?**

Aggregation via generalized averaging and voting

1. Generalized averaging

From averaging probability distributions to averaging mass functions:

$$m(\{c_j\}) = \frac{\sum_{t=1}^{T} m_t(\{c_j\})}{T},\ j = 1, \ldots, K \quad m(\Omega) = \frac{\sum_{t=1}^{T} m_t(\Omega)}{T} \tag{14}$$

The mass function m is still quasi-Bayesian

2. Generalized voting

From voting for a single class to voting for a subset of classes:

$$m(A) = \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}(\mathrm{ID}(m_t) = A), \tag{15}$$

where $\mathrm{ID}(\cdot)$ is the interval dominance that returns the set of non-dominated classes for each tree

The mass function m is no longer quasi-Bayesian

Cautious decision-making: maximizing the lower expected utility

Objective:

Consider mass function m and the discounted utility $u_{Aj} = d_\alpha(|A|)\mathbb{1}(c_j \in A)$

where $d_\alpha(|A|) = \frac{1.6}{|A|} - \frac{0.6}{|A|^2}$

Cautious predictions can be made according to $\hat{Y} = \underset{A \subseteq \Omega}{\arg\max}\ \underline{EU}(m, A, \mathbf{U})$

Strategy:

1. **We showed that $\underline{EU}(m, A, \mathbf{U}) = d_\alpha(|A|)Bel(A)$**
2. Consider $A \subseteq \Omega$ with $|A| = i$, $d_\alpha(|A|)$ is a constant
3. Then, $\underset{|A|=i, A \subseteq \Omega}{\arg\max}\ \underline{EU}(m, A, \mathbf{U}) = \underset{|A|=i, A \subseteq \Omega}{\arg\max}\ Bel(A), \quad i = 1, \ldots, |\Omega|$

Search for the maximizer of the $\underline{EU}$ by scanning $A \subseteq \Omega$ with increasing cardinality

Example of cautious random forest

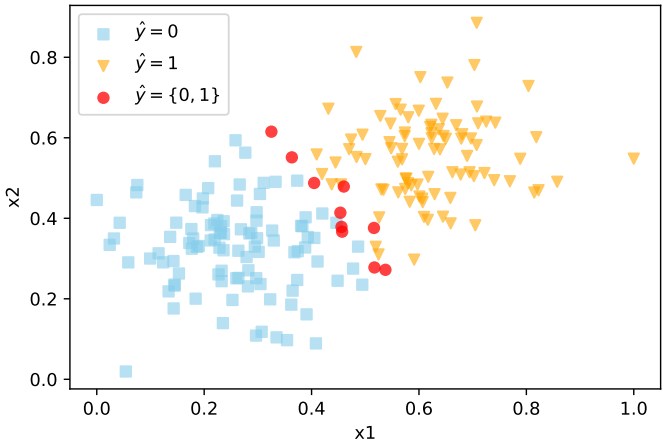
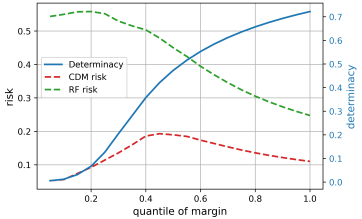
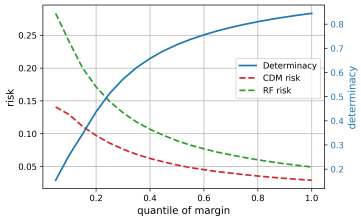


Figure: Cautious predictions made by the cautious random forest

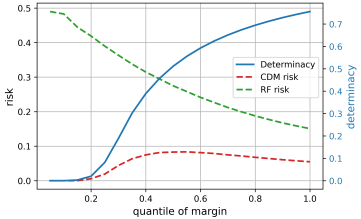
Adaptation of CRF to sample difficulty



(a) Vehicle



(b) Vowel



(c) Waveform

Figure: Behaviors of the cautious and classical random forests as a function of test set difficulty.

Comparison of different imprecise tree aggregation strategies

Table: Average evaluation results across 12 datasets

Criteria	MVA	AVE	CDM_Vote	CDM_AVE
determinacy	**0.9943**	0.7983	0.8382	0.8351
set size	**2.0272**	4.9464	2.2013	2.2180
single-set accuracy	0.8810	**0.9458**	0.9357	0.9376
set accuracy	0.9328	**0.9610**	0.9185	0.9251
u_{65} score	0.8792	0.8490	0.8794	**0.8798**
u_{80} score	0.8798	0.8712	0.8999	**0.9008**

Compared models:

1. MVA: minimum vote against
2. AVE: averaging with strong dominance

$$u_{65} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1.6}{|\boldsymbol{h}(\boldsymbol{x}_i)|} - \frac{0.6}{|\boldsymbol{h}(\boldsymbol{x}_i)|^2}\right)\cdot \mathbb{1}(y_i \in \boldsymbol{h}(\boldsymbol{x}_i))$$

$$u_{80} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{2.2}{|\boldsymbol{h}(\boldsymbol{x}_i)|} - \frac{1,2}{|\boldsymbol{h}(\boldsymbol{x}_i)|^2}\right)\cdot \mathbb{1}(y_i \in \boldsymbol{h}(\boldsymbol{x}_i))$$

Outline

Counterfactual explanations in XAI[7]

A counterfactual example of the prediction $\hat{y} = \boldsymbol{h}(\boldsymbol{x})$:

the closest instance to $\boldsymbol{x}$ that produces the predefined output $y' \neq \hat{y}$,

$$\boldsymbol{x}' = \underset{\boldsymbol{z} \in \mathcal{X}}{\arg\min}\, d(\boldsymbol{x}, \boldsymbol{z}) \text{ s.t. } \boldsymbol{h}(\boldsymbol{z}) = y' \tag{16}$$

[7] S. Verma, V. Boonsanong, M. Hoang, *et al.*, "Counterfactual explanations and algorithmic recourses for machine learning: A review," *arXiv preprint arXiv:2010.10596*, 2020.

Counterfactual explanations in XAI[7]

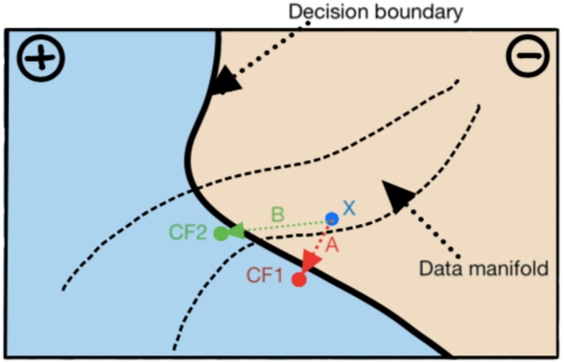
A counterfactual example of the prediction $\hat{y} = h(x)$:

the closest instance to x that produces the predefined output $y' \neq \hat{y}$,

$$x' = \underset{z \in \mathcal{X}}{\arg\min}\, d(x, z) \text{ s.t. } h(z) = y' \tag{16}$$

Desiderata of counterfactual examples

- Validity: reach the desired prediction
- Proximity: the smallest possible changes
- Sparsity: only a few changed features
- Actionability: feasible changes for users
- Plausibility: similar to instances in data
- Efficiency: quick generation process



[7] S. Verma, V. Boonsanong, M. Hoang, *et al.*, "Counterfactual explanations and algorithmic recourses for machine learning: A review," *arXiv preprint arXiv:2010.10596*, 2020.

Counterfactual explanations for indeterminate predictions

For instance $\boldsymbol{x} \in \mathcal{X}$ such that $\boldsymbol{h}(\boldsymbol{x}) = \{c_1, c_2\}$, its counterfactual instances $\boldsymbol{x}^{c_1}$ and $\boldsymbol{x}^{c_2}$ are defined as

$$\boldsymbol{x}^{c_1} = \underset{\boldsymbol{z} \in \mathcal{X}}{\arg\min}\, d(\boldsymbol{x}, \boldsymbol{z}) \text{ s.t. } \boldsymbol{h}(\boldsymbol{z}) = \{c_1\} \tag{17a}$$

$$\boldsymbol{x}^{c_2} = \underset{\boldsymbol{z} \in \mathcal{X}}{\arg\min}\, d(\boldsymbol{x}, \boldsymbol{z}) \text{ s.t. } \boldsymbol{h}(\boldsymbol{z}) = \{c_2\} \tag{17b}$$

Counterfactual explanations for indeterminate predictions

For instance $\boldsymbol{x} \in \mathcal{X}$ such that $\boldsymbol{h}(\boldsymbol{x}) = \{c_1, c_2\}$, its counterfactual instances $\boldsymbol{x}^{c_1}$ and $\boldsymbol{x}^{c_2}$ are defined as

$$\boldsymbol{x}^{c_1} = \underset{\boldsymbol{z} \in \mathcal{X}}{\arg\min}\, d(\boldsymbol{x}, \boldsymbol{z}) \text{ s.t. } \boldsymbol{h}(\boldsymbol{z}) = \{c_1\} \tag{17a}$$

$$\boldsymbol{x}^{c_2} = \underset{\boldsymbol{z} \in \mathcal{X}}{\arg\min}\, d(\boldsymbol{x}, \boldsymbol{z}) \text{ s.t. } \boldsymbol{h}(\boldsymbol{z}) = \{c_2\} \tag{17b}$$

Utilities of counterfactual examples:

1. indicate the smallest necessary modifications to obtain a determinate prediction.
2. identify the closest class to an indeterminate instance.
3. reveal the differences between the two classes.

How to generate counterfactual examples with determinate predictions?

Some existing methods[8]

1. Solving the optimization problem $\underset{\boldsymbol{z}\in\mathcal{X}}{\arg\min}\ \lambda\mathcal{L}(\boldsymbol{h}(\boldsymbol{z}), y') + d(\boldsymbol{x}, \boldsymbol{z})$: only for differentiable models
2. Searching in dataset or leaves: low proximity
3. Surrogate model: low validity
4. Integer linear programming: low efficiency

[8] R. Guidotti, "Counterfactual explanations and how to find them: literature review and benchmarking," *Data Mining and Knowledge Discovery*, pp. 1–55, 2022.

Efficient counterfactual example generation for CRF

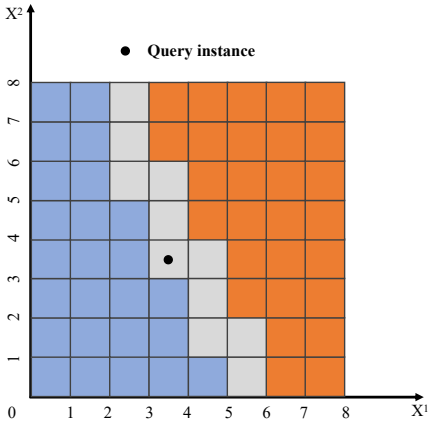
Branch-and-bound search algorithm for counterfactual generation[9]:

1. decompose the feature space into decision regions of the classifier
2. start the search from the region containing $\boldsymbol{x}$ and expand it to further regions

[9] P. Blanchart, "An exact counterfactual-example-based approach to tree-ensemble models interpretability," *arXiv preprint arXiv:2105.14820*, 2021.

Preprocessing: narrow the search region and ensure the actionability

1. **Decomposition of the feature space into elementary decision regions**



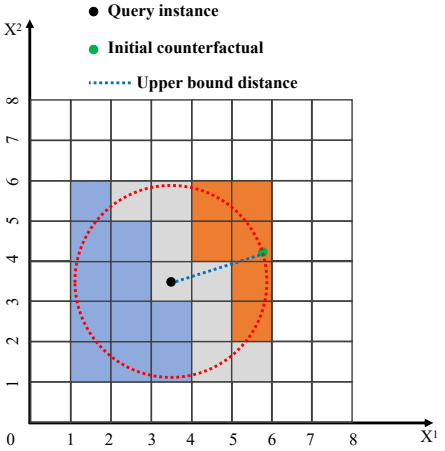
Preprocessing: narrow the search region and ensure the actionability

1. **Decomposition of the feature space into elementary decision regions**

2. **Upper-bound distance**

   $\boldsymbol{x}'$ is the initial counterfactual example for $\boldsymbol{x}$,

$$d_{sup} = d(\boldsymbol{x}, \boldsymbol{x}')$$



Preprocessing: narrow the search region and ensure the actionability

1. Decomposition of the feature space into elementary decision regions

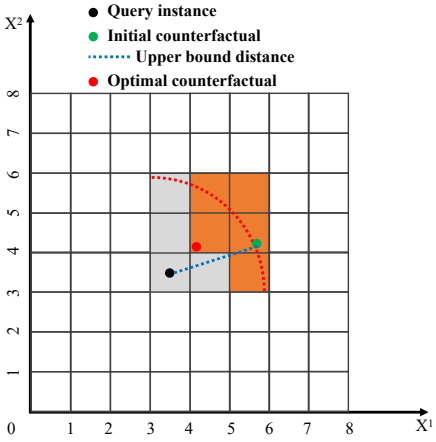
2. Upper-bound distance

$\boldsymbol{x}'$ is the initial counterfactual example for $\boldsymbol{x}$,

$$d_{sup} = d(\boldsymbol{x}, \boldsymbol{x}')$$

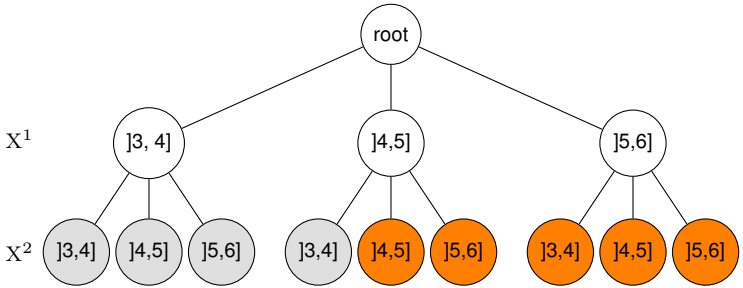
3. Feature constraints

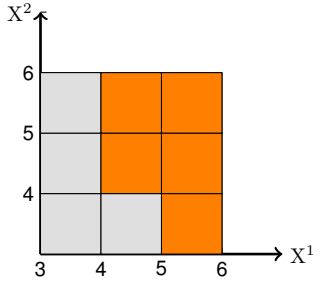
- immutable features
- features can be only increased
- features can be only decreased



Search tree

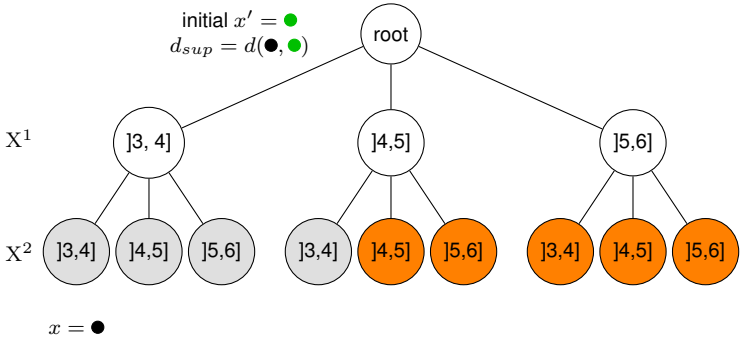
- Each level of the tree: an input feature
- Each node of the tree: a split interval
- Split intervals for each feature are sorted in ascending order according to their distance to x^j
- Each path from the root to a leaf: a decision region

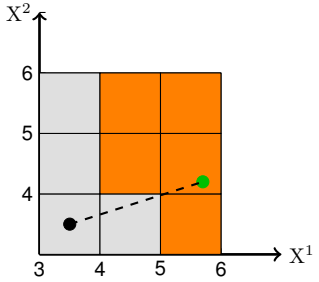




Branch-and-bound search for counterfactual examples

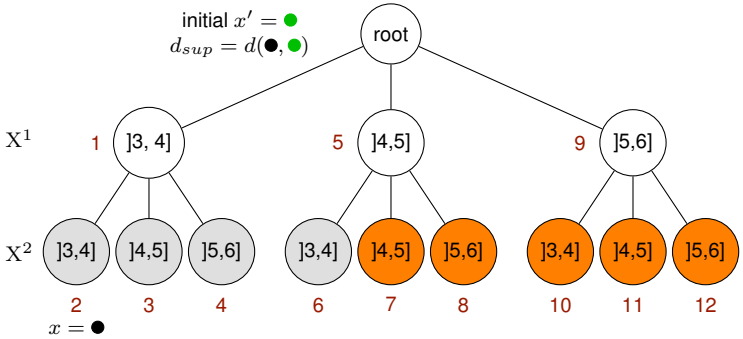
- Conduct a pre-order tree traversal (DFS)
- In eligible leaves: generate counterfactual examples and update d_{sup}
- Use d_{sup} to skip far regions

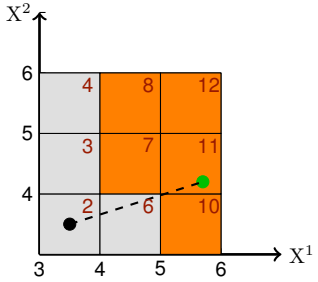




Branch-and-bound search for counterfactual examples

- Conduct a pre-order tree traversal (DFS)
- In eligible leaves: generate counterfactual examples and update d_{sup}
- Use d_{sup} to skip far regions
- Visit nodes in the order 1, 2, 3...

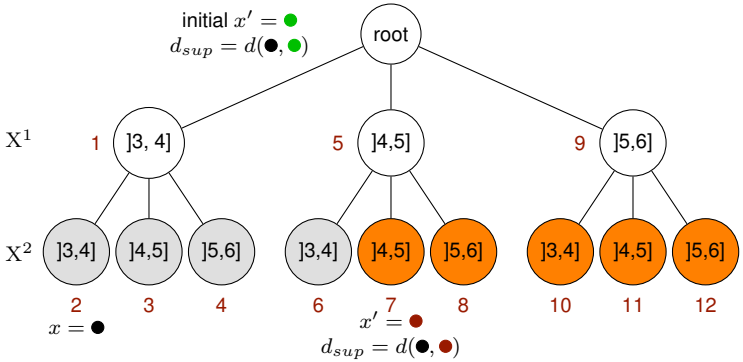


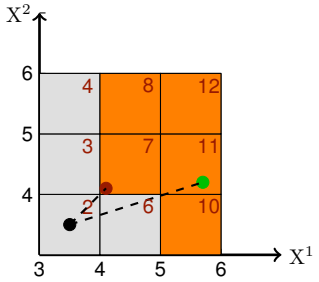


Branch-and-bound search for counterfactual examples

- Conduct a pre-order tree traversal (DFS)
- In eligible leaves: generate counterfactual examples and update d_{sup}
- Use d_{sup} to skip far regions

- Visit nodes in the order 1, 2, 3...
- Node 7: eligible leaf

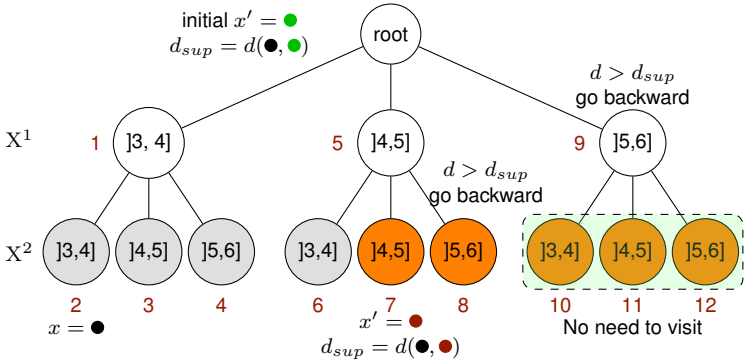


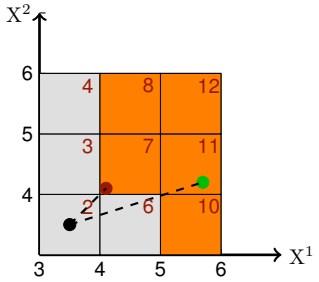


Branch-and-bound search for counterfactual examples

- Conduct a pre-order tree traversal (DFS)
- In eligible leaves: generate counterfactual examples and update d_{sup}
- Use d_{sup} to skip far regions

- Visit nodes in the order 1, 2, 3...
- Node 7: eligible leaf
- Node 8 and 9: distance larger than d_{sup}





Comparison of different counterfactual generation methods

Table: Average evaluation results of generated counterfactuals across 10 datasets

Metrics	MO[10]	DisCERN[11]	OFCC[12]	Ours
$L_2 \downarrow$	8.10	4.05	1.43	**1.00**
$L_1 \downarrow$	18.28	6.10	1.15	**1.00**
Sparsity (opt L_2) $\downarrow$	8.77	3.12	**1.18**	1.90
Sparsity (opt L_1) $\downarrow$	8.19	3.00	**1.17**	1.40
Plausibility $\uparrow$	**0.975**	0.961	0.931	0.934
Elapsed time $\downarrow$	**0.047**	0.087	1.517	3.632

[10] R. R. Fernández, I. M. De Diego, V. Aceña, A. Fernández-Isabel, and J. M. Moguerza, "Random forest explainability using counterfactual sets," *Information Fusion*, vol. 63, pp. 196–207, 2020.

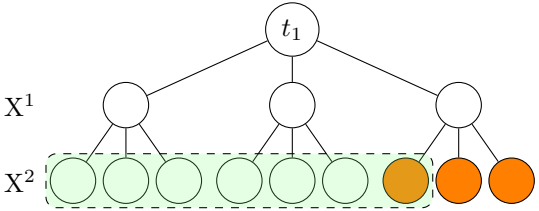
[11] N. Wiratunga, A. Wijekoon, Nkisi-Orji, *et al.*, "Discern: discovering counterfactual explanations using relevance features from neighbourhoods," in *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 1466–1473, IEEE, 2021.

[12] H. Zhang, B. Quost, and M.-H. Masson, "Explaining cautious random forests via counterfactuals," in *Building Bridges between Soft and Statistical Methodologies for Data Science*, pp. 390–397, Springer, 2022.

Accelerate the counterfactual generation

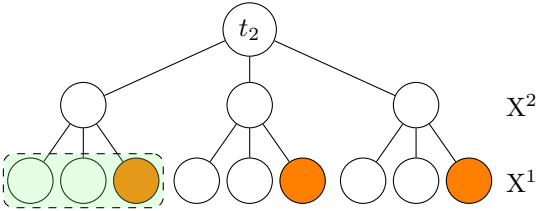
Proposition

- Determine decisive features that must be modified to get the desired prediction
- Put decisive features close to the bottom of the search tree



Orange leaves can generate counterfactuals

X^1 is the decisive feature



Help reducing the d_{sup} more quickly

Increase the chance to skip more regions

Determine decisive features by feature importance

Feature importance: asses the influence of input features on the output of a classifier

Output of cautious random forests:

- prediction uncertainty[13]: $\mathrm{Imp}(\boldsymbol{h}, \boldsymbol{x}) = \min(\overline{p}_1, \overline{p}_2) = \min(\underline{p}_1, \underline{p}_2) + \overline{p}_1 - \underline{p}_1$
- determinacy

Assessment methods[14]:

Methods	Scope	Used to explain
LIME: Local Interpretable Model-agnostic Explanations	Local	$\mathrm{Imp}(\boldsymbol{h}, \boldsymbol{x})$
SHAP: SHapley Additive exPlanations	Local	$\mathrm{Imp}(\boldsymbol{h}, \boldsymbol{x})$
SHAP-FI: SHAP Feature Importance	Global	$\mathrm{Imp}(\boldsymbol{h}, \boldsymbol{x})$
PFI: Permutation Feature Importance	Global	Determinacy
MDI: Mean Decrease in Impurity	Global	Tree structure

[13] E. Hüllermeier, S. Destercke, and M. H. Shaker, "Quantification of credal uncertainty in machine learning: A critical analysis and empirical comparison," in *Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence*, pp. 548–557, PMLR, 2022.

[14] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

Experiment 4: acceleration of counterfactual generation

Table: Impact of feature importance for the acceleration of counterfactual generation, reported in the percentage of the improvement compared with the original feature order

Data	MDI (Global)	PFI (Global)	SHAP-FI (Global)	SHAP (Local)	LIME (Local)
ADLT	**33.02**	28.39	15.37	9.14	-1.89
BIOD	**99.58**	98.53	97.39	29.45	46.27
COMP	**24.22**	23.60	23.60	17.39	-265.22
GERM	**98.75**	98.75	98.75	89.84	86.12
HELO	**49.63**	46.98	45.83	29.25	13.54
LIVR	6.59	7.72	**13.28**	1.91	-5.10
MAMO	**86.96**	82.61	80.43	78.26	23.91
PIMA	**6.83**	6.23	6.13	-13.41	-5.66
SPAM	10.41	3.64	13.76	**22.10**	13.42
WINE	2.80	**4.30**	3.18	-1.69	0.77

Outline

Summary of the research

Cautious random forests

- Generalized averaging and voting
- Efficient maximization of the lower expected utility to make cautious predictions

Resolving indeterminacy

1. Counterfactual examples as explanations for indeterminate predictions
2. Improved branch-and-bound search method for counterfactual generation
3. Feature importance to accelerate counterfactual example generation

Research Interests