Q: HOW MANY PH.D.'S DOES IT TAKE TO GET A POWERPOINT PRESENTATION TO WORK?

ANSWER: $(n+1)$

WHERE n = THE NUMBER OF ACADEMICS IN THE ROOM WHO THINK THEY KNOW HOW TO FIX IT, AND 1 = THE PERSON WHO FINALLY CALLS THE A/V TECHNICIAN.

Uncertainty reasoning and machine learning

A Few Applications

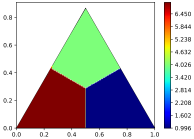
Vu-Linh Nguyen

Chaire de Professeur Junior, Laboratoire Heudiasyc
Université de technologie de Compiègne

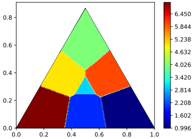
AOS4 master courses

Optimal Decision Rules

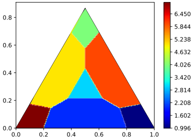
Frequentist approaches



0/1 loss



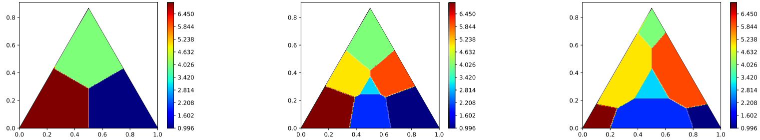
$u_{1.6}$



$u_{2.2}$

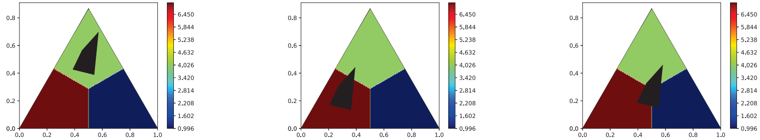
Optimal Decision Rules

Frequentist approaches



0/1 loss $u_{1.6}$ $u_{2.2}$

Credal approaches



Objectives

After this lecture, students should be able to describe (a few)

- probabilistic and credal classifiers
- and how to use them to make singleton and set-valued predictions,
- and their (potential) applications.

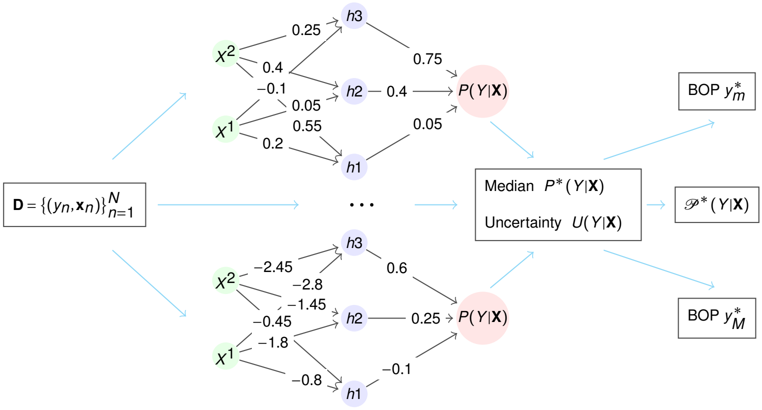
Outline

- Credal ensembling in multi-class classification
  - A median classifier: Learning and inference
  - A credal classifier: Learning and inference
  - Applications in machine learning
  - Compact deep ensembles
- Other applications

A formal framework [5]

Basic setup:

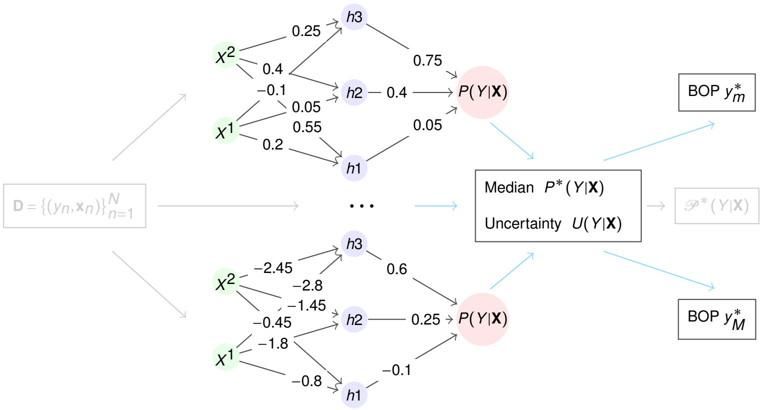
- Features $(X^1, \ldots, X^P)$ and a class variables Y
- An finite output space $\mathscr{Y} = \{y^1, \ldots, y^C\}$



Outline

- Credal ensembling in multi-class classification
  - A median classifier: Learning and inference
  - A credal classifier: Learning and inference
  - Applications in machine learning
  - Compact deep ensembles
- Other applications

A median classifier and its predictions [5]



Compute a median classifier

Basic setting:

- An ensemble $\mathbf{H} := \{\mathbf{h}^m | m \in [M] := \{1, \dots, M\}\}$ is made available
- A specified statistical distance d between distributions

A median classifier minimizes the average expected distance:

$$\mathbf{h}_d \in \operatorname*{argmin}_{\mathbf{h} \in \mathcal{H}} \mathbf{E}\left[\sum_{m=1}^{M} d(\mathbf{h}, \mathbf{h}^m)\right] = \operatorname*{argmin}_{\mathbf{h} \in \mathcal{H}} \int_{\mathbf{x} \in \mathcal{X}} \left[\sum_{m=1}^{M} d(\mathbf{h}(\mathbf{x}), \mathbf{h}^m(\mathbf{x}))\right] d\mathbf{x}.$$

Compute a median classifier

Basic setting:

- An ensemble $\mathbf{H} := \{\mathbf{h}^m | m \in [M] := \{1, \ldots, M\}\}$ is made available
- A specified statistical distance d between distributions

A median classifier minimizes the average expected distance:

$$\mathbf{h}_d \in \operatorname*{argmin}_{\mathbf{h} \in \mathcal{H}} \mathbf{E}\left[\sum_{m=1}^{M} d(\mathbf{h}, \mathbf{h}^m)\right] = \operatorname*{argmin}_{\mathbf{h} \in \mathcal{H}} \int_{\mathbf{x} \in \mathcal{X}} \left[\sum_{m=1}^{M} d(\mathbf{h}(\mathbf{x}), \mathbf{h}^m(\mathbf{x}))\right] d\mathbf{x}.$$

If no constraint on $\mathcal{H}$, $\mathbf{h}_d$ can be defined in an instance-wise manner:

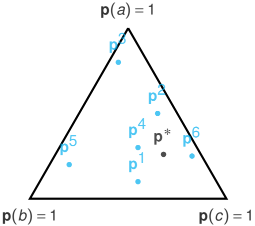
$$\mathbf{h}_d(\mathbf{x}) \in \operatorname*{argmin}_{\mathbf{h}(\mathbf{x}) \in \Delta^K} \sum_{m=1}^{M} d(\mathbf{h}(\mathbf{x}), \mathbf{h}^m(\mathbf{x})). \quad (1)$$

Compute a median classifier (cont.)

For each $\mathbf{x}$, dropping $\mathbf{x}$ and denoting $\mathbf{p} = \mathbf{h}$ give

$$\mathbf{p}_d \in \operatorname*{argmin}_{\mathbf{p} \in \Delta^K} \sum_{m=1}^{M} d(\mathbf{p}, \mathbf{p}^m). \qquad (2)$$

Examples of d are squared Euclidean distance (sE), L_1 distance, and KL divergence.



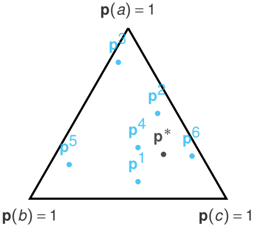
Ensemble $\mathbf{H}$ and $\mathbf{p}_{\text{sE}}$

Compute a median classifier (cont.)

For each $\mathbf{x}$, dropping $\mathbf{x}$ and denoting $\mathbf{p} = \mathbf{h}$ give

$$\mathbf{p}_d \in \operatorname*{argmin}_{\mathbf{p} \in \Delta^K} \sum_{m=1}^{M} d(\mathbf{p}, \mathbf{p}^m). \tag{2}$$

Examples of d are squared Euclidean distance (sE), L_1 distance, and KL divergence.



Ensemble $\mathbf{H}$ and $\mathbf{p}_{sE}$

For any convex distance d:

- The convex optimization problem (2) can be solved using any solver.
- Close-form solution $\mathbf{p}_{sE}$ = averaging the distributions class-wise.

Bayesian-optimal predictions

Basic set (instance-wise manner):

- The median distribution $\mathbf{p}_d$ is given.
- A the higher the better utility $u : \mathscr{Y} \times \mathscr{Y} \longmapsto \mathbb{R}_+$

A **Bayesian-optimal prediction (BOP) of** u is

$$y_d^u \in \underset{y' \in \mathscr{Y}}{\operatorname{argmax}} \mathbf{E}\left[u(y', y)\right] = \underset{y' \in \mathscr{Y}}{\operatorname{argmax}} \sum_{y \in \mathscr{Y}} u(y', y) \mathbf{p}_d(y). \tag{3}$$

Bayesian-optimal predictions

Basic set (instance-wise manner):

- The median distribution $\mathbf{p}_d$ is given.
- A the higher the better utility $u : \mathscr{Y} \times \mathscr{Y} \longmapsto \mathbb{R}_+$

A **Bayesian-optimal prediction (BOP) of** u is

$$y_d^u \in \underset{y' \in \mathscr{Y}}{\operatorname{argmax}} \mathbf{E}[u(y', y)] = \underset{y' \in \mathscr{Y}}{\operatorname{argmax}} \sum_{y \in \mathscr{Y}} u(y', y) \mathbf{p}_d(y). \quad (3)$$

Commonly used utilities, such as $0/1$ and cost-sensitive accuracies:

- Find a BOP (3) takes from $O(K)$ to $O(K^2)$
- A BOP $y_d^{0/1}$ (3) of $0/1$ accuracy = a most probable class

Bayesian-optimal set-valued predictions

Basic set (instance-wise manner):

- The median distribution $\mathbf{p}_d$ is given.
- A the higher the better utility $U : \mathscr{Y} \times 2^{\mathscr{Y}} \longmapsto \mathbb{R}_+$

A **Bayesian-optimal prediction (BOP) of** U is

$$Y_d^U \in \underset{Y' \subset \mathscr{Y}}{\operatorname{argmax}} \mathbf{E}\left[U(Y', y)\right] = \underset{Y' \subset \mathscr{Y}}{\operatorname{argmax}} \sum_{y \in \mathscr{Y}} U(Y', y) \mathbf{p}_d(y). \tag{4}$$

Bayesian-optimal set-valued predictions

Basic set (instance-wise manner):

- The median distribution $\mathbf{p}_d$ is given.
- A the higher the better utility $U : \mathscr{Y} \times 2^{\mathscr{Y}} \longmapsto \mathbb{R}_+$

A **Bayesian-optimal prediction (BOP) of** U is

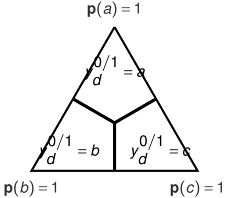
$$Y_d^U \in \operatorname*{argmax}_{Y' \subset \mathscr{Y}} \mathbf{E}\left[U(Y', y)\right] = \operatorname*{argmax}_{Y' \subset \mathscr{Y}} \sum_{y \in \mathscr{Y}} U(Y', y)\mathbf{p}_d(y). \tag{4}$$

Commonly used utilities, such as utility-discounted accuracies:

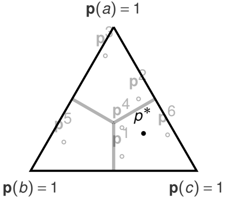
$$U(Y', y) = \frac{1}{g(|Y'|)} [\![y \in Y']\!], \tag{5}$$

- Find a BOP Y_d^U (4) takes $O(K \log(K))$.
- A BOP Y_d^U (4) consists of the most probable classes on $\mathbf{p}_d$.

Probabilistic uncertainty scores

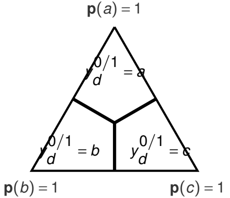


Regions $\Delta_{\mathbf{x}}^{st}$ for different $y_d^{0/1}$

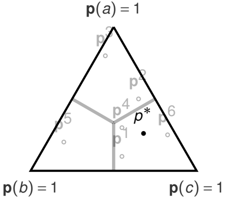


Ensemble **H** and $\mathbf{p}_{sE}$

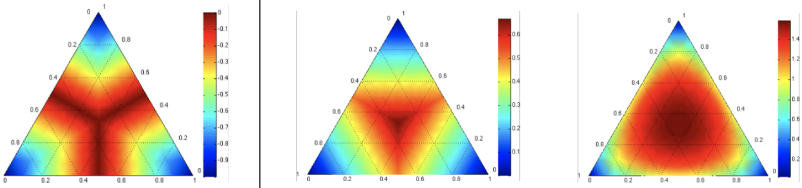
Probabilistic uncertainty scores



Regions $\Delta_{\mathbf{x}}^{st}$ for different $y_d^{0/1}$



Ensemble **H** and $\mathbf{p}_{sE}$



(a) Smallest margin (↑) (b) Least confidence (↓) (c) Entropy (↓)

Heatmaps illustrating the **behavior of probabilistic uncertainty scores**

Probabilistic uncertainty scores (Cont.)

Smallest margin (↑) is defined as

$$S_{\mathrm{SM}}(\mathbf{p}_d) = \mathbf{p}_d\left(y^{\mathrm{st}}\right) - \mathbf{p}_d\left(y^{\mathrm{nd}}\right). \tag{6}$$

Example: A classification problem with $\mathscr{Y} = \{a, b, c\}$:

	$\mathbf{x}_1$	$\mathbf{x}_2$
	50→(0.6, 0.4, 0.0) 50→(0.0, 0.4, 0.6)	100→(0.3, 0.4, 0.3)
$\mathbf{p}_{\mathrm{sE}}$	(0.3, 0.4, 0.3)	
S_{SM} (↑)	0.1	
	Should we consider $\mathbf{x}_1$ and $\mathbf{x}_2$ the same?	

Probabilistic uncertainty scores (Cont.)

Smallest margin (↑) is defined as

$$S_{\mathrm{SM}}(\mathbf{p}_d) = \mathbf{p}_d\left(y^{\mathrm{st}}\right) - \mathbf{p}_d\left(y^{\mathrm{nd}}\right). \tag{6}$$

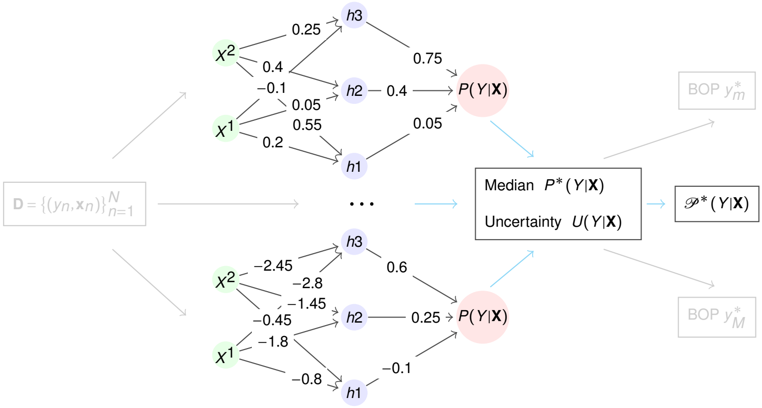
Example: A classification problem with $\mathscr{Y} = \{a, b, c\}$:

	$\mathbf{x}_3$	$\mathbf{x}_4$
	80→(1.0, 0.0, 0.0) 20→ (0.0, 1.0, 0.0)	100→(0.8, 0.2, 0.0)
$\mathbf{p}_{\mathrm{sE}}$	(0.8, 0.2, 0.0)	
S_{SM} (↑)	0.6	
	Should we consider $\mathbf{x}_3$ and $\mathbf{x}_4$ the same?	

Outline

- Credal ensembling in multi-class classification
  - A median classifier: Learning and inference
  - A credal classifier: Learning and inference
  - Applications in machine learning
  - Compact deep ensembles
- Other applications

A credal classifier and its predictions [5]



For any query instance, once $\mathscr{P}^*(\mathscr{Y}|\mathbf{x})$ is estimated:

- IP decision rules can be called to make set-valued predictions
- uncertainty scores defined for credal sets can be computed.

Estimate a credal classifier

Each credal classifier $\mathbf{CH}_{\alpha}^{d}$ is defined in a point-wise manner:

$$\mathbf{CH}_{\alpha}^{d} := \left\{ \mathbf{p} := \sum_{m=1}^{M_{\alpha}} \gamma_m \mathbf{p}^{(m)} | \gamma_m \geq 0, m \in [M_{\alpha}], \sum_{m=1}^{M_{\alpha}} \gamma_m = 1 \right\}, \quad (7)$$

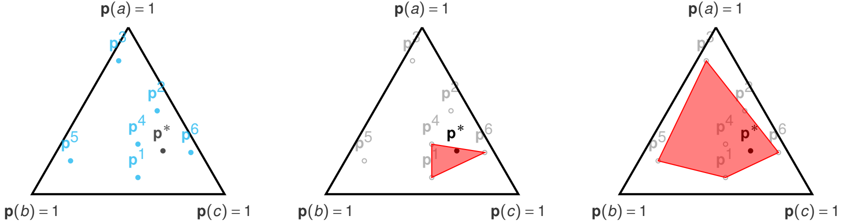
where $\mathbf{p}^{(m)}$ is the m-th closet point to $\mathbf{p}_d$ **according to the distance** d.

Estimate a credal classifier

Each credal classifier $\mathbf{CH}_\alpha^d$ is defined in a point-wise manner:

$$\mathbf{CH}_\alpha^d := \left\{ \mathbf{p} := \sum_{m=1}^{M_\alpha} \gamma_m \mathbf{p}^{(m)} | \gamma_m \geq 0, m \in [M_\alpha], \sum_{m=1}^{M_\alpha} \gamma_m = 1 \right\}, \quad (7)$$

where $\mathbf{p}^{(m)}$ is the m-th closet point to $\mathbf{p}_d$ **according to the distance** d.



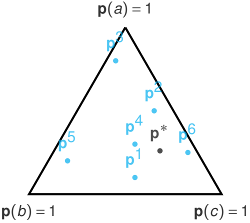
Ensemble **H** and $\mathbf{p}_{\text{sE}}^*$ Credal set $\mathbf{CH}_{0.5}^{\text{sE}}$ Credal set $\mathbf{CH}_{0}^{\text{sE}}$

Estimate a credal classifier

Each credal classifier $\mathbf{CH}_\alpha^d$ is defined in a point-wise manner:

$$\mathbf{CH}_\alpha^d := \left\{ \mathbf{p} := \sum_{m=1}^{M_\alpha} \gamma_m \mathbf{p}^{(m)} | \gamma_m \geq 0, m \in [M_\alpha], \sum_{m=1}^{M_\alpha} \gamma_m = 1 \right\}, \quad (7)$$

where $\mathbf{p}^{(m)}$ is the m-th closet point to $\mathbf{p}_d$ **according to the distance** d.

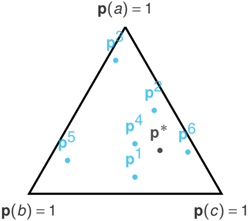


Ensemble **H** and $\mathbf{p}^*_{sE}$ Credal set $\mathbf{CH}^{sE}_{0.5}$ Credal set $\mathbf{CH}^{sE}_{0}$

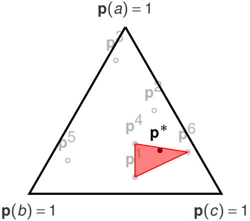
The hyperparameter α^* ← nested cross validation or a validation set.

Optimal set-valued predictions under IP decision rules

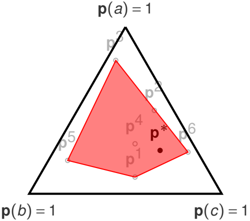
Basic set (instance-wise manner): The credal set $\mathbf{CH}^{d}_{\alpha^*}$ is given.



Ensemble $\mathbf{H}$ and $\mathbf{p}_{\text{sE}}$



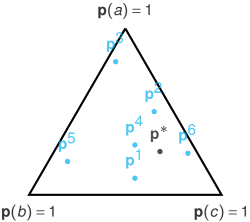
Credal set $\mathbf{CH}^{\text{sE}}_{0.5}$



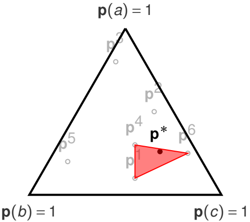
Credal set $\mathbf{CH}^{\text{sE}}_{0}$

Optimal set-valued predictions under IP decision rules

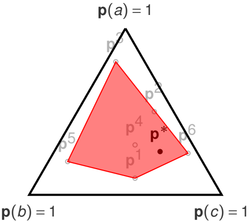
Basic set (instance-wise manner): The credal set $\mathbf{CH}^{d}_{\alpha^*}$ is given.



Ensemble **H** and $\mathbf{p}_{\text{sE}}$



Credal set $\mathbf{CH}^{\text{sE}}_{0.5}$



Credal set $\mathbf{CH}^{\text{sE}}_{0}$

- Any IP decision rule $R_{\text{IP}} : 2^{\Delta^K} \longmapsto 2^{\mathscr{Y}}$ can be applied.
- Any related algorithmic solutions can be leveraged.

Optimal set-valued predictions under IP decision rules (Cont.)

Basic set (instance-wise manner):

- The credal set $\mathbf{CH}^{d}_{\alpha^*}$ is given.
- A the higher the better utility $u : \mathscr{Y} \times \mathscr{Y} \longmapsto \mathbb{R}_+$

E-admissibility under u:

- A class y is E-admissible if there exist $\mathbf{p} \in \mathbf{CH}^{d}_{\alpha^*}$ so that $y = y^u$.
- This can be checked by solving a linear program.

Optimal set-valued predictions under IP decision rules (Cont.)

Basic set (instance-wise manner):

- The credal set $\mathbf{CH}^d_{\alpha^*}$ is given.
- A the higher the better utility $u : \mathscr{Y} \times \mathscr{Y} \longmapsto \mathbb{R}_+$

E-admissibility under u:

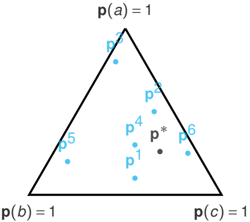
- A class y is E-admissible if there exist $\mathbf{p} \in \mathbf{CH}^d_{\alpha^*}$ so that $y = y^u$.
- This can be checked by solving a linear program.

Maximality under u:

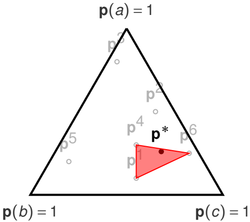
- A class y is maximal if there doesn't exist $y' \neq y$ such that y' dominates y on all $\mathbf{p} \in \mathbf{CH}^d_{\alpha^*}$ (w.r.t. u).
- This can be checked by solving $K-1$ linear programs.
- We can also enumerate all the distributions $\mathbf{p}^m$, $m \in [M]$.

Credal set-based uncertainty scores

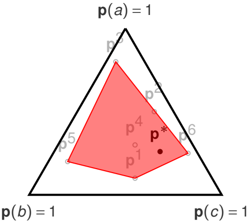
Basic set (instance-wise manner): The credal set $\mathbf{CH}^{d}_{\alpha^*}$ is given.



Ensemble $\mathbf{H}$ and $\mathbf{p}_{\mathrm{sE}}$



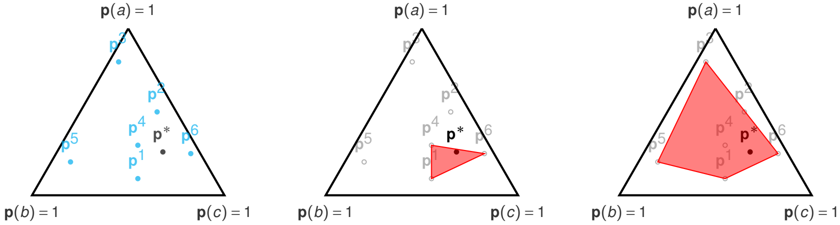
Credal set $\mathbf{CH}^{\mathrm{sE}}_{0.5}$



Credal set $\mathbf{CH}^{\mathrm{sE}}_{0}$

Credal set-based uncertainty scores

Basic set (instance-wise manner): The credal set $\mathbf{CH}^{d}_{\alpha^*}$ is given.



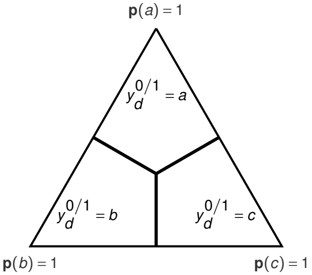
Ensemble **H** and $\mathbf{p}_{\mathrm{sE}}$ — Credal set $\mathbf{CH}^{\mathrm{sE}}_{0.5}$ — Credal set $\mathbf{CH}^{\mathrm{sE}}_{0}$

- Any credal set-based uncertainty score can be used.
- Any related algorithmic solutions can be leveraged.

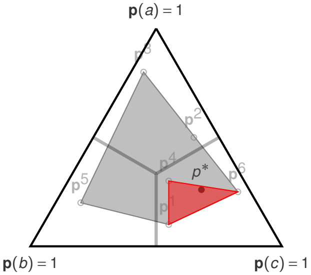
Credal set-based uncertainty scores (Cont.)

Decision-related uncertainty scores:

- How certain the ensemble **H** is about y_d^u?
- How consensus of the ensemble members is about y_d^u?



Regions $\Delta_{\mathbf{x}}^{st}$ for different $y_d^{0/1}$



Credal set $\mathbf{CH}_0^{sE}(\mathbf{x})$ with $\mathbf{p}_{sE}$

Decision-related uncertainty scores

Basic setting (instance-wise manner):

- The median distribution $\mathbf{p}_d$ is given.
- The predictions $\{\mathbf{p}^m | m \in [M]\}$ are given.
- A probabilistic uncertainty score $S : \Delta^K \longmapsto \mathbb{R}$ is given.

Decision-related uncertainty scores

Basic setting (instance-wise manner):

- The median distribution $\mathbf{p}_d$ is given.
- The predictions $\{\mathbf{p}^m | m \in [M]\}$ are given.
- A probabilistic uncertainty score $S : \Delta^K \longmapsto \mathbb{R}$ is given.

A **decision-related uncertainty** version of S is (defined as its empirical expectation)

$$\mathrm{RS}(\mathbf{p}_d^u) := \frac{1}{M+1}\left(\sum_{m=1}^{M} \left[\!\left[\mathbf{p}^m \in \mathbf{CH}^d_{y_d^u} \right]\!\right] S(\mathbf{p}^m) + S(\mathbf{p}_d)\right), \quad (8)$$

where $\left[\!\left[\mathbf{p}^m \in \mathbf{CH}^d_{y_d^u} \right]\!\right] = 1$ implies y_d^u is a best solution on $\mathbf{p}^m$ under u.

Decision-related uncertainty scores (Cont.)

Smallest margin (↑) is defined as

$$S_{\text{SM}}\left(\mathbf{p}_d\right) = \mathbf{p}_d\left(y^{\text{st}}\right) - \mathbf{p}_d\left(y^{\text{nd}}\right). \tag{9}$$

Example: A classification problem with $\mathscr{Y} = \{a, b, c\}$:

	$\mathbf{x}_1$	$\mathbf{x}_2$
	50→(0.6, 0.4, 0.0) 50→(0.0, 0.4, 0.6)	100→(0.3, 0.4, 0.3)
$\mathbf{p}_{\text{sE}}$	(0.3, 0.4, 0.3)	
S_{SM} (↑)	0.1	
RS_{SM} (↑)	0.0	0.1

Decision-related uncertainty scores (Cont.)

Smallest margin (↑) is defined as

$$S_{\text{SM}}\left(\mathbf{p}_d\right) = \mathbf{p}_d\left(y^{\text{st}}\right) - \mathbf{p}_d\left(y^{\text{nd}}\right). \quad (9)$$

Example: A classification problem with $\mathscr{Y} = \{a, b, c\}$:

	$\mathbf{x}_3$	$\mathbf{x}_4$
	80→(1.0, 0.0, 0.0) 20→(0.0, 1.0, 0.0)	100→(0.8, 0.2, 0.0)
$\mathbf{p}_{\text{sE}}$	(0.8, 0.2, 0.0)	
S_{SM} (↑)	0.6	
RS_{SM} (↑)	0.798	0.6

Should we put weights on the impact of ensemble members?

Outline

- Credal ensembling in multi-class classification
  - A median classifier: Learning and inference
  - A credal classifier: Learning and inference
  - Applications in machine learning
  - Compact deep ensembles
- Other applications

Experimental setting

Basic setting:

- Use random forests of cardinality 100 as the ensembles
- Follow a 10-cross validation protocol.
- Use hyperparameter $\alpha^* \leftarrow$ nested 10 fold cross validation

Assess the impact of $\mathbf{p}_{\mathrm{sE}}$, $\mathbf{p}_{\mathrm{L1}}$ and $\mathbf{p}_{\mathrm{KL}}$ on

- the clean version of the data sets
- noisy version (randomly flip the class of 25% of training instances)

Experimental setting

Basic setting:

- Use random forests of cardinality 100 as the ensembles
- Follow a 10-cross validation protocol.
- Use hyperparameter $\alpha^* \leftarrow$ nested 10 fold cross validation

Assess the impact of $\mathbf{p}_{\text{sE}}$, $\mathbf{p}_{\text{L1}}$ and $\mathbf{p}_{\text{KL}}$ on

- the clean version of the data sets
- noisy version (randomly flip the class of 25% of training instances)

Once credal set $\text{CH}^{d}_{\alpha^*}$ is computed, it is used to

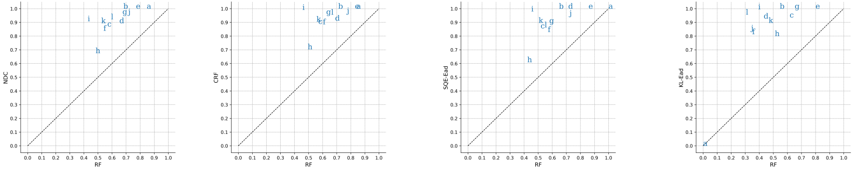
- find the set-valued prediction under the E-admissibility rule.

Results on clean data sets: U_{65} scores (in %) [5]

Data set: (N,P,K)	NDC	SQE-E	L1-E	KL-E	CRF	**CH_0**
eco.: (336,7,8)	85.51	86.07	85.81	**87.07**	84.46	43.60
der.: (358,34,6)	97.18	97.05	97.22	**98.59**	96.19	51.74
lib.: (360, 90, 15)	76.58	73.35	75.24	**79.41**	73.45	14.60
vow.: (990, 10, 11)	86.63	86.35	87.65	**92.35**	82.68	17.75
win.: (1599, 11, 6)	**68.66**	68.32	68.39	68.63	67.35	36.53
seg.: (2300, 19, 7)	97.17	97.12	96.99	**97.64**	96.73	71.00

Results on clean data sets: U_{65} scores (in %) [5]

Data set: (N,P,K)	NDC	SQE-E	L1-E	KL-E	CRF	**CH_0**
eco.: (336,7,8)	85.51	86.07	85.81	**87.07**	84.46	43.60
der.: (358,34,6)	97.18	97.05	97.22	**98.59**	96.19	51.74
lib.: (360, 90, 15)	76.58	73.35	75.24	**79.41**	73.45	14.60
vow.: (990, 10, 11)	86.63	86.35	87.65	**92.35**	82.68	17.75
win.: (1599, 11, 6)	**68.66**	68.32	68.39	68.63	67.35	36.53
seg.: (2300, 19, 7)	97.17	97.12	96.99	**97.64**	96.73	71.00



(a) NDC vs RF (b) CRF vs RF (e) SQE-Ead vs RF (f) KL-Ead vs RF

Correctness of cautious predictors (vertical) vs accuracy of RF (horizontal)

Experimental setting [5]

Basic setting:

- Randomly flip the class of 25% of training instances
- Using random forests of cardinality 100 as the ensembles
- The median $\mathbf{p}_{\mathrm{sE}}$ is employed
- Assess smallest margin S_{SM} (↑) and $\mathbf{RS}_{\mathrm{SM}}$ (↑)

Experimental setting [5]

Basic setting:

- Randomly flip the class of 25% of training instances
- Using random forests of cardinality 100 as the ensembles
- The median $\mathbf{p}_{\mathrm{sE}}$ is employed
- Assess smallest margin S_{SM} (↑) and $\mathbf{RS}_{\mathrm{SM}}$ (↑)

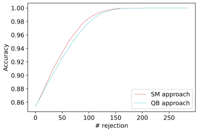
Budget based rejection protocol requires

- a sufficiently large number of test instances,
- a predefined number (or proportion) of rejections.

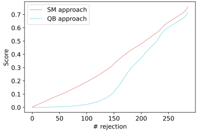
Threshold-based rejection protocol

- requires a predefined threshold on uncertainty score (↑),
- rejects instances whose scores are lower than the threshold.

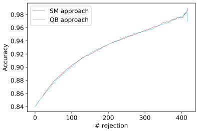
Results on noisy data sets [5]



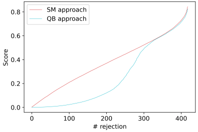
(a) derma. + SM



(b) derma. + SM



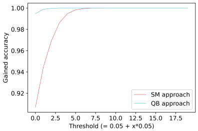
(c) forest + SM



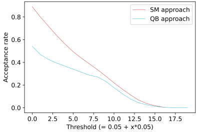
(d) forest + SM

Test accuracy and chosen score as the functions of the number of rejections

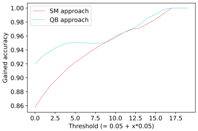
20×5 cross-validation with (train, test) = (20%, 80%)



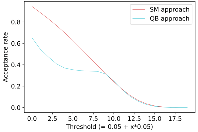
(a) derma. + SM



(b) derma. + SM



(c) forest + SM



(d) forest + SM

Test accuracy and acceptance rate as the functions of the threshold

20×5 cross-validation with (train, test) = (20%, 80%)

Experimental setting

Basic setting:

- Randomly flip the class of 25% of training + pool instances
- Using random forests of cardinality 100 as the ensembles
- The median $\mathbf{p}_{\mathrm{sE}}$ is employed
- Assess smallest margin S_{SM} (↑) and $\mathbf{RS}_{\mathrm{SM}}$ (↑)

Experimental setting

Basic setting:

- Randomly flip the class of 25% of training + pool instances
- Using random forests of cardinality 100 as the ensembles
- The median $\mathbf{p}_{\text{sE}}$ is employed
- Assess smallest margin S_{SM} (↑) and $\mathbf{RS}_{\text{SM}}$ (↑)

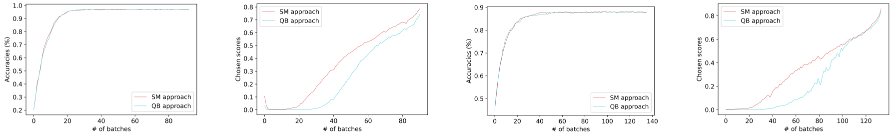
Budget based sampling protocol

- requires a predefined number (or proportion) of queries,
- stops when the predefined number is reached.

Threshold-based sampling protocol

- requires a predefined threshold on uncertainty score (↑),
- stops when the predefined threshold is reached.

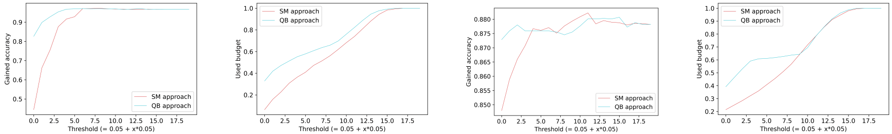
Results on noisy data sets [5]



(a) derma. + SM (b) derma. + SM (c) forest + SM (d) forest + SM

Test accuracy and chosen score as the functions of the number of queries

10×5 cross-validation with (train, pool, test) = (3%, 77%, 20%)



(a) derma. + SM (b) derma. + SM (e) forest + SM (f) forest + SM

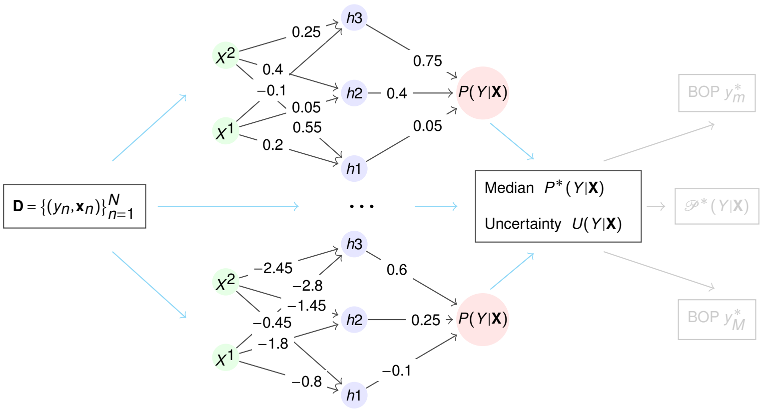
Test accuracy and used budget as the functions of the threshold

10×5 cross-validation with (train, pool, test) = (3%, 77%, 20%)

Outline

- Credal ensembling in multi-class classification
  - A median classifier: Learning and inference
  - A credal classifier: Learning and inference
  - Applications in machine learning
  - Compact deep ensembles
- Other applications

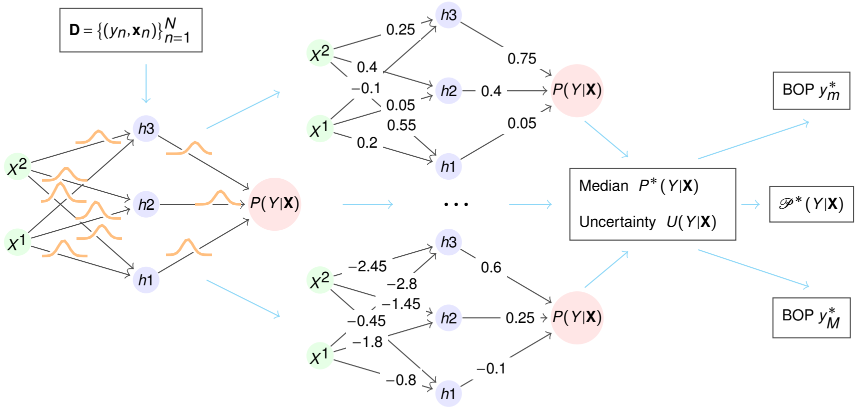
Conventional deep ensembles



Compared to the use of a single network:

- Much longer training time + Much larger storage memory
- Longer inference time

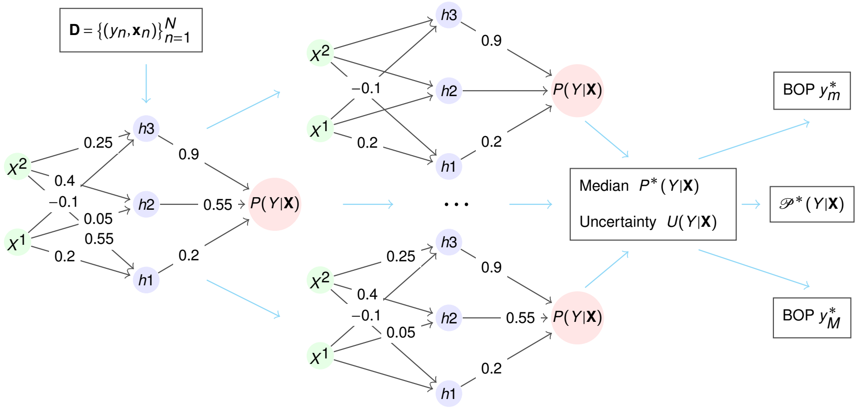
A BNN as an ensemble [8]



Compared to the use of a single network:

- A bit longer training time + A bit larger storage memory
- Longer inference time

A CNN with dropout predictions as an ensemble [9]



Compared to the use of a single network:

- Similar training time + Similar storage memory
- Longer inference time

Experimental setting

Basic setting:

- Use BNNs with 100 Monte Carlo runs as the ensembles
- Use the clean version of the data sets

Assess the impact of $\mathbf{p}_{\mathrm{sE}}$, $\mathbf{p}_{\mathrm{L1}}$ and $\mathbf{p}_{\mathrm{KL}}$ on

	Image	train/test	# classes
CIFAR-10	32x32 color	50,000/10,000	10
Fashion-MNIST	grayscale	60,000/10,000	10

Experimental setting

Basic setting:

- Use BNNs with 100 Monte Carlo runs as the ensembles
- Use the clean version of the data sets

Assess the impact of $\mathbf{p}_{\mathrm{sE}}$, $\mathbf{p}_{\mathrm{L1}}$ and $\mathbf{p}_{\mathrm{KL}}$ on

	Image	train/test	# classes
CIFAR-10	32x32 color	50,000/10,000	10
Fashion-MNIST	grayscale	60,000/10,000	10

Once $\mathbf{p}_d$ is computed, it is used to

- find precise prediction optimizing the $u_{0,1}$,
- find set-valued predictions optimizing the u_{65} and u_{80} [3].

Average $u_{0,1}$, u_{65} and u_{80} on the test set

Results [8]	CIFAR-10			Fashion MNIST		
	sE	L1	KL	sE	L1	KL
$u_{0/1}$ (↑)	90.04	90.10	**90.14**	93.07	**93.11**	93.08
opt_u65_eva_u65 (↑)	90.47	**90.51**	90.46	**93.38**	93.31	93.26
opt_u80_eva_u80 (↑)	**91.77**	91.76	91.76	**94.41**	94.39	94.27
u65_set_size (↓)	2.03	**2.02**	2.03	**2.02**	**2.02**	**2.02**
u80_set_size (↓)	2.04	**2.02**	2.03	**2.02**	**2.02**	**2.02**

A closer look at $u_{0,1}$ and u_{65}

Results [8]	CIFAR-10			Fashion MNIST		
	sE	L1	KL	sE	L1	KL
c_pr_u65_c_si (↑)	94.91	95.91	**97.53**	97.53	97.19	**98.43**
c_pr_u65_c_se (↓)	5.08	4.08	**2.46**	2.46	2.80	**1.56**
w_pr_u65_c_se (↑)	**32.12**	26.86	17.64	24.96	**25.39**	15.75
w_pr_u65_w_se (↓)	15.26	11.81	**7.50**	5.05	5.95	**4.62**
w_pr_u65_w_si (↓)	**52.61**	61.31	74.84	69.98	**68.65**	79.62

A closer look at $u_{0,1}$ and u_{80}

Results [8]	CIFAR-10			Fashion MNIST		
	sE	L1	KL	sE	L1	KL
c_pr_u80_c_si (↑)	86.89	93.22	**94.28**	94.34	94.03	**95.47**
c_pr_u80_c_se (↓)	13.10	6.77	**5.71**	5.65	5.96	**4.52**
w_pr_u80_c_se (↑)	**53.21**	37.07	34.38	43.86	**44.26**	37.28
w_pr_u80_w_se (↓)	23.89	19.19	**16.32**	10.82	10.44	**8.67**
w_pr_u80_w_si (↓)	**22.89**	43.73	49.29	45.31	**45.28**	54.04

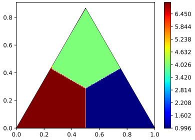
Results [8]	CIFAR-10			Fashion MNIST		
	sE	L1	KL	sE	L1	KL
$u_{0/1}$ (↑)	90.04	90.10	**90.14**	93.07	**93.11**	93.08
u65_set_size (↓)	2.03	**2.02**	2.03	**2.02**	**2.02**	**2.02**
u80_set_size (↓)	2.04	**2.02**	2.03	**2.02**	**2.02**	**2.02**
c_pr_u65_c_si (↑)	94.91	95.91	**97.53**	97.53	97.19	**98.43**
c_pr_u65_c_se (↓)	5.08	4.08	**2.46**	2.46	2.80	**1.56**
w_pr_u65_c_se (↑)	**32.12**	26.86	17.64	24.96	**25.39**	15.75
w_pr_u65_w_se (↓)	15.26	11.81	**7.50**	5.05	5.95	**4.62**
w_pr_u65_w_si (↓)	**52.61**	61.31	74.84	69.98	**68.65**	79.62
c_pr_u80_c_si (↑)	86.89	93.22	**94.28**	94.34	94.03	**95.47**
c_pr_u80_c_se (↓)	13.10	6.77	**5.71**	5.65	5.96	**4.52**
w_pr_u80_c_se (↑)	**53.21**	37.07	34.38	43.86	**44.26**	37.28
w_pr_u80_w_se (↓)	23.89	19.19	**16.32**	10.82	10.44	**8.67**
w_pr_u80_w_si (↓)	**22.89**	43.73	49.29	45.31	**45.28**	54.04

Outline

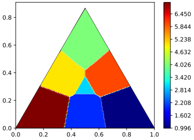
- Credal ensembling in multi-class classification
- Other applications
  - Large language models with safety requirements
  - Credal Predictions for Out-of-distribution detection

Optimal Decision Rules

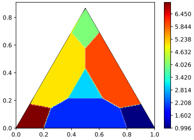
Frequentist approaches



0/1 loss

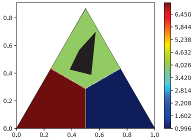


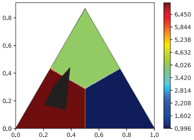
$u_{1.6}$

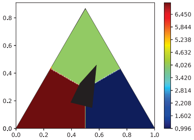


$u_{2.2}$

Credal approaches







Outline

- Credal ensembling in multi-class classification
- Other applications
  - Large language models with safety requirements
  - Credal Predictions for Out-of-distribution detection

Basic setup [4]

Multiple-choice question answering:

- Each prompt $\boldsymbol{x}$ is associated with a response $r_k \in \mathcal{Y} = \mathcal{R} = \{r_1, \ldots, r_K\}$.
- Predict the probability masses $\mathbf{p}(r_k|\boldsymbol{x})$ $k = 1, \ldots, K$.
- Return a **Bayesian-optimal prediction (BOP) of** u

$$r^u \in \operatorname*{argmax}_{r' \in \mathcal{Y}} \mathbf{E}\left[u(r', r)\right] = \operatorname*{argmax}_{y' \in \mathcal{Y}} \sum_{r \in \mathcal{Y}} u(r', r)\mathbf{p}(r|\boldsymbol{x}).$$

Basic setup [4]

Multiple-choice question answering:

- Each prompt $\boldsymbol{x}$ is associated with a response $r_k \in \mathscr{Y} = \mathscr{R} = \{r_1, \ldots, r_K\}$.
- Predict the probability masses $\mathbf{p}(r_k|\boldsymbol{x})$ $k = 1, \ldots, K$.
- Return a **Bayesian-optimal prediction (BOP) of** u

$$r^u \in \underset{r' \in \mathscr{Y}}{\operatorname{argmax}} \mathbf{E}\left[u(r', r)\right] = \underset{y' \in \mathscr{Y}}{\operatorname{argmax}} \sum_{r \in \mathscr{Y}} u(r', r)\mathbf{p}(r|\boldsymbol{x}).$$

Helpfulness and safety scores are given for each prompt:

- A helpfulness score $h_k(\uparrow)$, measuring how well r_k answers the query
- A safetyrisk score $s_k(\downarrow)$, measuring the likelihood that r_k violates safety policies

Basic setup [4]

Multiple-choice question answering:

- Each prompt $\boldsymbol{x}$ is associated with a response $r_k \in \mathscr{Y} = \mathscr{R} = \{r_1, \ldots, r_K\}$.
- Predict the probability masses $\mathbf{p}(r_k|\boldsymbol{x})$ $k = 1, \ldots, K$.
- Return a **Bayesian-optimal prediction (BOP) of** u

$$r^u \in \underset{r' \in \mathscr{Y}}{\operatorname{argmax}} \mathbf{E}[u(r', r)] = \underset{y' \in \mathscr{Y}}{\operatorname{argmax}} \sum_{r \in \mathscr{Y}} u(r', r)\mathbf{p}(r|\boldsymbol{x}).$$

Helpfulness and safety scores are given for each prompt:

- A helpfulness score $h_k(\uparrow)$, measuring how well r_k answers the query
- A safetyrisk score $s_k(\downarrow)$, measuring the likelihood that r_k violates safety policies
- Expose the helpfulness-safety trade-off when predicting $\mathbf{p}(r_k|\boldsymbol{x})$ $k = 1, \ldots, K$.

Helpfulness and safety scores

Let $r_1 \in \mathscr{Y} = \mathscr{R} = \{r_1, \ldots, r_K\}$ be a special safe fallback answer:

- r_1 is anodyne and policycompliant (for example, a refusal or a generic safe statement).
- r_1 contains 0 useful information ($h_k(\uparrow)$) but also incurs 0 risk $s_k(\downarrow)$.

Helpfulness and safety scores

Let $r_1 \in \mathcal{Y} = \mathcal{R} = \{r_1, \ldots, r_K\}$ be a special safe fallback answer:

- r_1 is anodyne and policycompliant (for example, a refusal or a generic safe statement).
- r_1 contains 0 useful information ($h_k(\uparrow)$) but also incurs 0 risk $s_k(\downarrow)$.

For each $r_k \in \mathcal{Y} = \mathcal{R}$, we can define

- The helpfulness lift ($\uparrow$): $H_k = h_k - h_1$
- The extra risk ($\downarrow$): $S_k = s_k - s_1$

Helpfulness and safety scores

Let $r_1 \in \mathscr{Y} = \mathscr{R} = \{r_1, \ldots, r_K\}$ be a special safe fallback answer:

- r_1 is anodyne and policycompliant (for example, a refusal or a generic safe statement).
- r_1 contains 0 useful information ($h_k(\uparrow)$) but also incurs 0 risk $s_k(\downarrow)$.

For each $r_k \in \mathscr{Y} = \mathscr{R}$, we can define

- The helpfulness lift ($\uparrow$): $H_k = h_k - h_1$
- The extra risk ($\downarrow$): $S_k = s_k - s_1$

For each probability distribution $\mathbf{p}(\cdot|\boldsymbol{x})$, we define

$$\text{expected helpfulness lift} \quad \sum_{r_k \in \mathscr{Y}} H_k \mathbf{p}(r_k|\boldsymbol{x}) \tag{10}$$

$$\text{expected extra risk} \quad \sum_{r_k \in \mathscr{Y}} S_k \mathbf{p}(r_k|\boldsymbol{x}) \tag{11}$$

Constrained optimization problem

Predict $\mathbf{p}(\cdot|\boldsymbol{x})$ to maximize expected helpfulness lift with acceptable expected risk

$$\mathbf{p} \in \underset{\mathbf{p}\in H^K}{\operatorname{argmax}} \sum_{r_k \in \mathcal{Y}} H_k \mathbf{p}(r_k|\boldsymbol{x}) \quad (12)$$

$$\text{s.t} \sum_{r_k \in \mathcal{Y}} S_k \mathbf{p}(r_k|\boldsymbol{x}) \leq T\,. \quad (13)$$

A more elaborate version + computational aspects can be found in [4].

Constrained optimization problem

Predict $\mathbf{p}(\cdot|\boldsymbol{x})$ to maximize expected helpfulness lift with acceptable expected risk

$$\mathbf{p} \in \underset{\mathbf{p} \in H^K}{\operatorname{argmax}} \sum_{r_k \in \mathcal{Y}} H_k \mathbf{p}(r_k|\boldsymbol{x}) \tag{12}$$

$$\text{s.t} \sum_{r_k \in \mathcal{Y}} S_k \mathbf{p}(r_k|\boldsymbol{x}) \leq T. \tag{13}$$

A more elaborate version + computational aspects can be found in [4].

Probe classifiers

For each prompt $\boldsymbol{x}$ and each response $r_k \neq r_1$,

- Query helpfulness probe with question “Is this answer helpful? (Yes/No)" on $(\boldsymbol{x}, r_k)$ to obtain

$$h_k = \log\left(\frac{p_h^{\text{yes}}}{p_h^{\text{yes}} + p_h^{\text{no}}}\right). \tag{14}$$

Probe classifiers

For each prompt $\boldsymbol{x}$ and each response $r_k \neq r_1$,

- Query helpfulness probe with question “Is this answer helpful? (Yes/No)" on $(\boldsymbol{x}, r_k)$ to obtain

$$h_k = \log\left(\frac{p_h^{\text{yes}}}{p_h^{\text{yes}} + p_h^{\text{no}}}\right). \tag{14}$$

- Query safety probe with question “Is this answer risky? (Yes/No)" $(\boldsymbol{x}, r_k)$ to obtain

$$s_k = \log\left(\frac{p_s^{\text{yes}}}{p_s^{\text{yes}} + p_s^{\text{no}}}\right). \tag{15}$$

Probe classifiers

For each prompt $\boldsymbol{x}$ and each response $r_k \neq r_1$,

- Query helpfulness probe with question “Is this answer helpful? (Yes/No)" on $(\boldsymbol{x}, r_k)$ to obtain

$$h_k = \log\left(\frac{p_h^{\text{yes}}}{p_h^{\text{yes}} + p_h^{\text{no}}}\right). \tag{14}$$

- Query safety probe with question “Is this answer risky? (Yes/No)" $(\boldsymbol{x}, r_k)$ to obtain

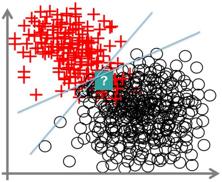
$$s_k = \log\left(\frac{p_s^{\text{yes}}}{p_s^{\text{yes}} + p_s^{\text{no}}}\right). \tag{15}$$

Probe classifiers can be LLM models [4].

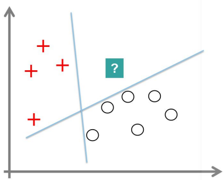
Outline

- Credal ensembling in multi-class classification
- Other applications
  - Large language models with safety requirements
  - Credal Predictions for Out-of-distribution detection

Epistemic and aleatoric uncertainties [1, 7]

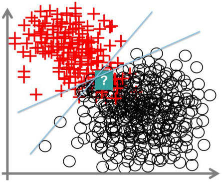


Aleatoric uncertainty: Classes are really mixed,
→ irreducible by collecting more information

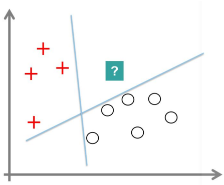


Epistemic uncertainty: Lack of information,
→ reducible by collecting more information

Epistemic and aleatoric uncertainties [1, 7]



Aleatoric uncertainty: Classes are really mixed,
→ irreducible by collecting more information



Epistemic uncertainty: Lack of information,
→ reducible by collecting more information

- Which kind of uncertainty is more interesting in OoD?
- How to quantify these degrees of uncertainty?

Basic setup

The given training data $\boldsymbol{D} \subset \mathscr{X} \times \mathscr{Y}$

- is used to estimate a classifier,
- which predicts, for each instance $\boldsymbol{x}$, $\mathscr{P}(Y|\boldsymbol{X})$.

Basic setup

The given training data $\boldsymbol{D} \subset \mathscr{X} \times \mathscr{Y}$

- is used to estimate a classifier,
- which predicts, for each instance $\boldsymbol{x}$, $\mathscr{P}(Y|\boldsymbol{X})$.

$\mathscr{P}(Y|\boldsymbol{X})$ is used to quantify the degree of epistemic uncertainty $\mathrm{EU}(\boldsymbol{x})$:

- using its volume (not recommended!) [6].
- using the disconsensus of its members [2].

Basic setup

The given training data $\boldsymbol{D} \subset \mathscr{X} \times \mathscr{Y}$

- is used to estimate a classifier,
- which predicts, for each instance $\boldsymbol{x}$, $\mathscr{P}(Y|\boldsymbol{X})$.

$\mathscr{P}(Y|\boldsymbol{X})$ is used to quantify the degree of epistemic uncertainty $\mathrm{EU}(\boldsymbol{x})$:

- using its volume (not recommended!) [6].
- using the disconsensus of its members [2].

Prediction: Instances with high $\mathrm{EU}(\boldsymbol{x})$ are predicted as OoD instances.

An (approximate) detail framework [2]

The degree of epistemic uncertainty on $\mathscr{P}(Y|\boldsymbol{X})$

$$\mathrm{EU}(\boldsymbol{x}) = \overline{U}_E(\boldsymbol{x}) - \underline{U}_E(\boldsymbol{x}), \tag{16}$$

where U_E is the entropy (see Lecture 2) and

$$\overline{U}_E(\boldsymbol{x}) = \max_{\mathbf{p} \in \mathscr{P}(Y|\boldsymbol{X})} U_E(\mathbf{p}(Y|\boldsymbol{x})), \tag{17}$$

$$\underline{U}_E(\boldsymbol{x}) = \min_{\mathbf{p} \in \mathscr{P}(Y|\boldsymbol{X})} U_E(\mathbf{p}(Y|\boldsymbol{x})) \tag{18}$$

An (approximate) detail framework [2]

The degree of epistemic uncertainty on $\mathscr{P}(Y|\boldsymbol{X})$

$$\mathrm{EU}(\boldsymbol{x}) = \overline{U}_E(\boldsymbol{x}) - \underline{U}_E(\boldsymbol{x}), \tag{16}$$

where U_E is the entropy (see Lecture 2) and

$$\overline{U}_E(\boldsymbol{x}) = \max_{\mathbf{p} \in \mathscr{P}(Y|\boldsymbol{X})} U_E(\mathbf{p}(Y|\boldsymbol{x}), \tag{17}$$

$$\underline{U}_E(\boldsymbol{x}) = \min_{\mathbf{p} \in \mathscr{P}(Y|\boldsymbol{X})} U_E(\mathbf{p}(Y|\boldsymbol{x}) \tag{18}$$

$\mathscr{P}(Y|\boldsymbol{X})$ is estimated using interval probabilities (See lecture 2)

$$\left[\underline{\mathbf{p}}(y|\boldsymbol{x}), \overline{\mathbf{p}}(y|\boldsymbol{x})\right], \forall y \in \mathscr{Y}. \tag{19}$$

References I

[1] E. Hüllermeier and W. Waegeman.
Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods.
Machine learning, 110(3):457–506, 2021.

[2] T. Löhr, P. Hofman, F. Mohr, and E. Hüllermeier.
Credal prediction based on relative likelihood.
arXiv preprint arXiv:2505.22332, 2025.

[3] T. Mortier, M. Wydmuch, K. Dembczyński, E. Hüllermeier, and W. Waegeman.
Efficient set-valued prediction in multi-class classification.
Data Mining and Knowledge Discovery, 35(4):1435–1469, 2021.

[4] T. Nguyen and L. Tran-Thanh.
Safety game: Balancing safe and informative conversations with blackbox agentic ai using lp solvers.
arXiv preprint arXiv:2510.09330, 2025.

[5] V.-L. Nguyen, H. Zhang, and S. Destercke.
Credal ensembling in multi-class classification.
Machine Learning, pages 1–64, 2024.

[6] Y. Sale, M. Caprio, and E. Hüllermeier.
Is the volume of a credal set a good measure for epistemic uncertainty?
In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 1795–1804, 2023.

[7] R. Senge, S. Bösner, K. Dembczyński, J. Haasenritter, O. Hirsch, N. Donner-Banzhoff, and E. Hüllermeier.
Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty.
Information Sciences, 255:16–29, 2014.

References II

[8] K.-D. Tran, X.-T. Hoang, V.-L. Nguyen, S. Destercke, and V.-N. Huynh.
Robust classification in bayesian neural networks.
In *Proceedings of the Eleventh International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making (IUKM)*, pages 29–41. Springer, 2025.

[9] K.-D. Tran, D.-M. Nguyen, V.-L. Nguyen, X.-T. Hoang, S. Destercke, and V.-N. Huynh.
Compact cautious deep ensembling in multi-class classification.
Under preparation, pages 1–50, 2025.