Last Time:

- alignment
- active learning

Lecture 15

HRI, FALL '26

Andrea Bajcsy

This Time:

- uncertainty

Uncertainty in HRI .... beyond what we've seen so far?

So far, we have seen alot of uncertainty in HRI problems!

The key type of uncertainty has been in the intentions, goals, rewards that a person has in their mind.

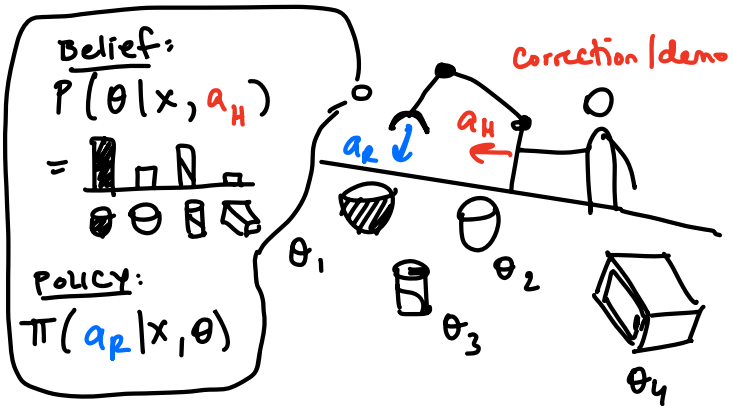
We designed: (by hand or via learning algorithms)

$\theta \in$  // reward/intention parameters

$P(\theta \mid x, a_H)$ // belief given data (from human)

intent → θ; state of world → x; human data → a_H

SO FAR

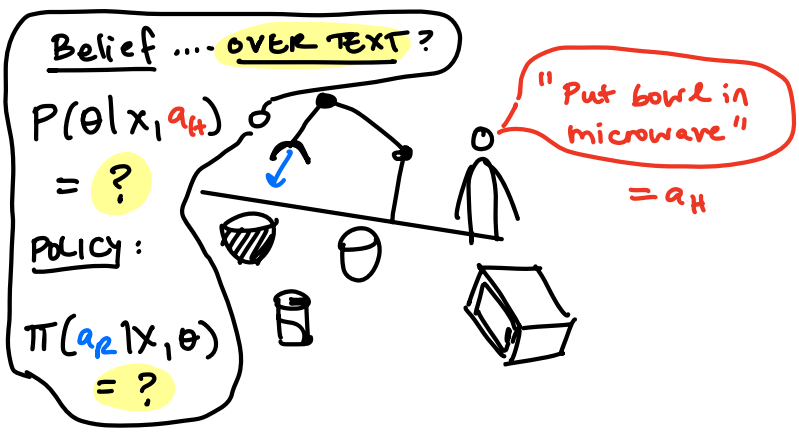


$\theta \in$ 

$= \{$  $\}$

(object poses)

NEXT UP



$\theta \in$  = { "put metal in microwave"
"put ceramic in microwave"
"put can in microwave"
"pick up microwave" }
....

Q1 How to get $P(\theta \mid x, a_H)$ when everything is in text?

↳ x = "on the table there is metal bowl, ceramic, ..."

θ = possible things human wants R to do (ex. "put the can in microwave")

a_H = task description from H (ex. "put bowl in microwave")

A1 Lets use a foundation model (LLM)!

1) Let Θ be a finite number of textual descriptions of what the robot could do in this env:

2) Pose a multiple choice Q & A problem to a LLM:

PROMPT:

Human: "put bowl in microwave"
State: There is metal, ceramic....
Which is best?
(A) "put metal in microwave"
(B) "put ceramic in microwave"
(C) "put can in microwave"
(D) "pick up microwave"

→ LLM →

ex. softmax of logits or self-reported!

(A) 0.65 → most likely
(B) 0.2
(C) 0.1
(D) 0.05

$= P_{LLM}(\theta | x, a_R)$

Call a language-conditioned policy to execute:

$\pi(a_R | x, \theta = (A) = \text{"put metal in microwave"})$



⚠ wait, something isn't right!

Q2 What went wrong?

"put bowl in microwave" ⇒ ambiguous if:  or θ ⇒

These should be equally likely!

A) "put metal in microwave"
B) "put ceramic in microwave"

(or maybe bias towards ceramic)

IDEALLY: The robot would accurately know that (A), (B) are both likely and ask for help:



A2 The LLM's "belief update" $P_{LLM}(\theta | x, a_R)$ is miscalibrated! We cannot trust the probabilities!

B/c the LLM is pre-trained (& closed-source and/OR LARGE), it's super hard to change the way it assigns probabilities by retraining, ensembling etc... ☹

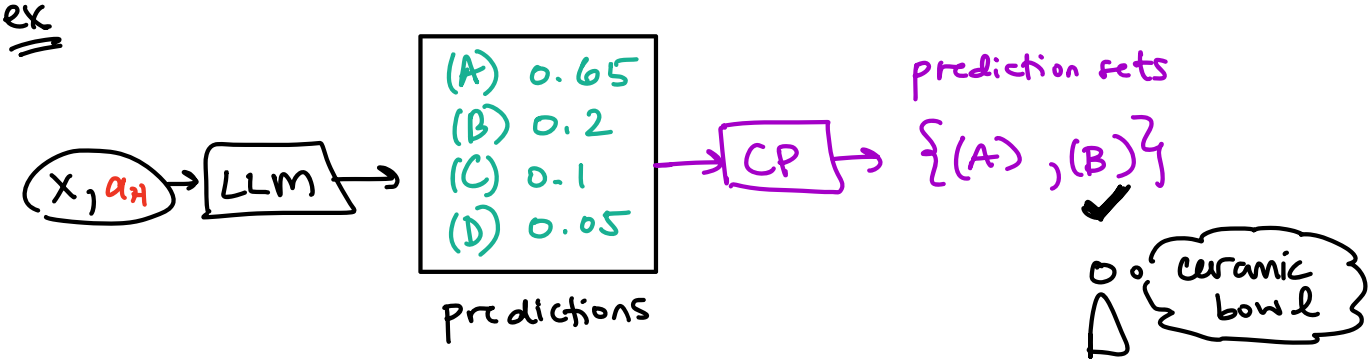
IDEA (from Ren et al, CoRL 2023): Let's use a technique called conformal prediction to ensure that these types of robots "know when they don't know".

CONFORMAL PREDICTION (CP) (Vovk, 2005)

This is a deep topic in statistics and all the proofs on why it works are outside the scope of the class. BUT, the idea + algorithm are quite straightforward:

IDEA Lets take an unreliable predictive model & construct prediction sets which include the true label in the set with high probability.

ex



⇒ if set sz = 1 → execute! if set sz > 1 → ask Q!

Conformal Prediction (CP) ensures that:

$$\underbrace{C(x^{test}, a_H^{test})}_{\text{set of labels (ex. } \{(A),(B)\}} \quad \text{s.t.} \quad \underbrace{P(\theta^* \in C(x^{test}, a_H^{test}))}_{\text{probability that true label } (\theta^*) \text{ is in the set...}} \geq \overbrace{1-\alpha}^{\text{... is at least } 1-\alpha!}$$

error tolerance!

ex. $\alpha = 0.2$ then 80% of the time, true label should be in set C!

even when we can't trust P!

5 STEPS TO CP

⊛ see code ex!

① collect a calibration dataset of inputs & labels:

"put ceramic bowl in microwave"

$$\mathcal{D}_{calib} = \{((\underbrace{x, a_H}_{\text{inputs}}), \underbrace{\theta^* = (B)}_{\text{labels}})\}$$

② Run your predictive model on all inputs in $\mathcal{D}_{calib}$ & measure how badly it mis-predicted!

⇒ prediction: $P_{LLM}(\theta = (B) \mid x, a_H) = 0.2$

true label

⇒ mis-prediction score: $1 - P_{LLM} = 0.8$ // higher the score, worse the prediction!

③ choose your error tolerance: α

(ex. $\alpha = 0.2$ means I'm ok not including θ^* in the set 20% of the time)

④ look @ the empirical distribution of $\mathcal{D}_{calib}$ scores and find $\hat{q}$, which is the $\lceil (N+1)(1-\alpha) \rceil / N$ empirical quantile of the scores.

⚠ why this works is out of class scope!

NOTE: Computing plain $1-\alpha$ empirical quantile is bit too optimistic, b/c test point (x^{test}, a_H^{test}) is new pt. we didn't see in $\mathcal{D}_{calib}$.

$\Rightarrow$ Using $\lceil (N+1)(1-\alpha) \rceil / N$ empirical quantile (i.e. $\lceil (N+1)(1-\alpha) \rceil^{th}$ smallest score) accounts for this xtra point at test-time!

(5) At deployment time, the robot carries around $\hat{q}_f$ and constructs the prediction set by including all labels that the predictor (P_{LLM}) is @ least $1-\hat{q}_f$ confident in:

$$C(x^{test}, a_H^{test}) = \{ \theta : P_{LLM}(\theta \mid x^{test}, a_H^{test}) \geq 1 - \hat{q}_f \}$$

In Ren et al, CoRL 2023, the assumption was that any option the LLM was choosing from ((A) – (D)), it could perfectly execute – only uncertainty was semantic.

BUT THIS IS STRONG ASSUMPTION!

ex. Imagine you learn (via VLA) a low-level policy

$$a \sim \pi(a \mid o, \ell)$$

Sometimes the model fails, sometimes the request ℓ is outside of robot's capabilities (ex. π trained on pick + place and ℓ = "pour some milk").

Yuan*, Wu*, Bajcsy. RSS 2026 Handles this!

Supplementary Notes:

Def^n: The α-quantile is the smallest x s.t.

$$P(X \leq x) \geq \alpha$$

(P: cumulative dist. function over rand. variable X)

EXAMPLE: Assume 10 scores, sorted:

$\alpha = 0.75$

0.02, 0.05, 0.08, 0.10, 0.15, 0.3, 0.45, 0.55, 0.7, 0.9

$\alpha = 0.5$ $\alpha = 0.8$

- empirical median ($\alpha = 0.5$): $\lceil 10 \times 0.5 \rceil = 5^{th}$ smallest $= 0.15$
  - ↳ 1/2 scores are @ or below value
- empirical 0.8-quantile: $\lceil 10 \times 0.8 \rceil = 8^{th}$ smallest $= 0.55$
  - ↳ 8/10 scores are @ or below value
- empirical 0.75-quantile: $\lceil 10 \times 0.75 \rceil = 8^{th}$ smallest $= 0.55$

→ Here, the CDF $P(X \leq x)$ counts each value once (l/c unique) so each value contributes 1/10 probability mass.

