Aalto University
School of Electrical
Engineering

Active Robot Learning for Temporal Task Models

Mattia Racca, Ville Kyrki

mattia.racca@aalto.fi

ACM/IEEE International Conference on Human Robot Interaction (HRI), 2018

Motivation

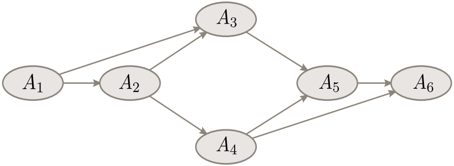
Programming a robot is hard, **pre-programming** a robot for all situations is ~~harder~~ impossible.

Robots should **learn by interacting** with their users.

We present a **active learning** approach that uses questions to learn in an interactive way to model **temporal task** (like making a sandwich) .

Problem Statement and Task Model

To model the task and the related user preferences, we choose a **Markov Chain**, whose states represent the task's actions A.



MODEL PARAMETERS

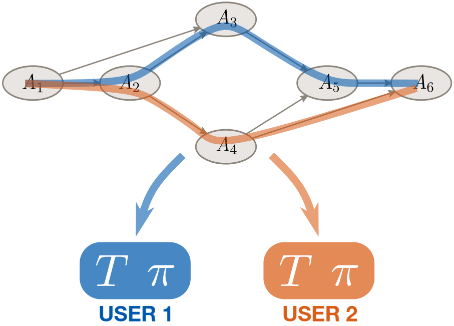
$$T = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1A} \\ p_{21} & p_{22} & \cdots & p_{2A} \\ \vdots & \vdots & \ddots & \vdots \\ p_{A1} & p_{A2} & \cdots & p_{AA} \end{bmatrix} \begin{matrix} \leftarrow \\ \leftarrow \\ \vdots \\ \leftarrow \end{matrix} \begin{matrix} \mathbf{Dir}(T_1|\boldsymbol{\alpha}) \\ \mathbf{Dir}(T_2|\boldsymbol{\alpha}) \\ \vdots \\ \mathbf{Dir}(T_A|\boldsymbol{\alpha}) \end{matrix}$$

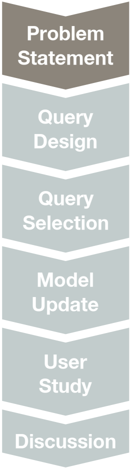
$$\pi = \begin{bmatrix} p_1 & p_2 & \cdots & p_A \end{bmatrix} \leftarrow \mathbf{Dir}(\pi|\boldsymbol{\alpha})$$

HYPERPARAMETERS

A Learning from Demonstration Approach

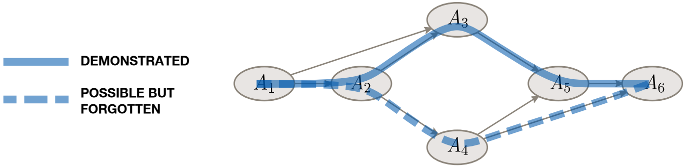
To learn the model parameters $\theta = \{\pi, T\}$, we can use a **Learning from Demonstration** approach. The parameters can be learned incrementally as new demonstrations are obtained.





A Learning from Demonstration Approach

This LfD strategy assumes the user to be able to provide **informative demonstrations**, covering all specifications needed.

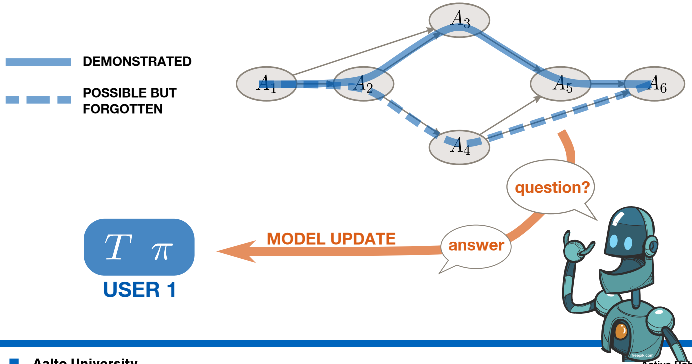


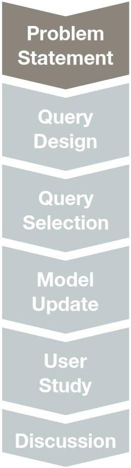
$$\tilde{T}\ \tilde{\pi} \neq T\ \pi$$

LEARNED ≠ **GROUND TRUTH**

Can we use Active Learning?

We can drop this assumption by allowing learning agent to **actively ask questions** about **missing** or **uncertain** details of the task and integrate the user's answer in the model.





Can we use Active Learning?

We can drop this assumption by allowing learning agent to **actively ask questions** about **missing** or **uncertain** details of the task and integrate the user's answer in the model.

We have 4 smaller problems.

- When to ask a question?
- How to construct a question?
- Which questions to ask?
- How to integrate the answer back in the model?

Problem Statement
Query Design
Query Selection
Model Update
User Study
Discussion

Can we use Active Learning?

We can drop this assumption by allowing learning agent to **actively ask questions** about **missing** or **uncertain** details of the task and integrate the user's answer in the model.

We have 4 smaller problems.

- When to ask a question? **During the demonstrations, after each action**
- How to construct a question?
- Which questions to ask?
- How to integrate the answer back in the model?

Query Design

Questions have to gather **useful information** for the training. However, they also need to be **understandable** for a non-expert user and easy to answer.

MODEL PARAMETERS

$$T = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1A} \\ p_{21} & p_{22} & \cdots & p_{2A} \\ \vdots & \vdots & \ddots & \vdots \\ p_{A1} & p_{A2} & \cdots & p_{AA} \end{bmatrix}$$

$$\pi = \begin{bmatrix} p_1 & p_2 & \cdots & p_A \end{bmatrix}$$

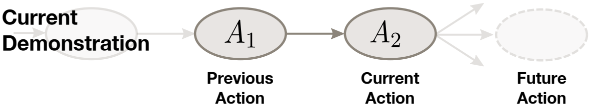
What is the probability of doing Action 2 after Action 1?
32% / 65% / ... / 99%

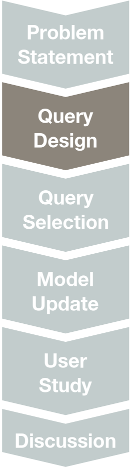
After Action 1, do you often do Action 2?
Yes / No.

Query Design

Questions have to gather **useful information** for the training. However, they also need to be **understandable** for a non-expert user and easy to answer.

Given the **temporal nature** of the model, we also have to take care that questions are **in context**.





Query Design

We design two types of queries, **Frequency Queries** and **Disambiguation Queries**, each further divided in two subtypes.

Query Design

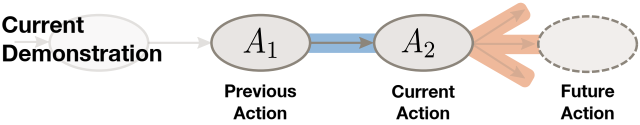
Frequency Queries aim to obtain ordering probability of a pair of actions.

Past FQ

Future FQ

After doing $\boxed{\mathbf{a_{pre}}}$, do you $\boxed{\mathbf{freq}}$ do $\boxed{\mathbf{a_{post}}}$?

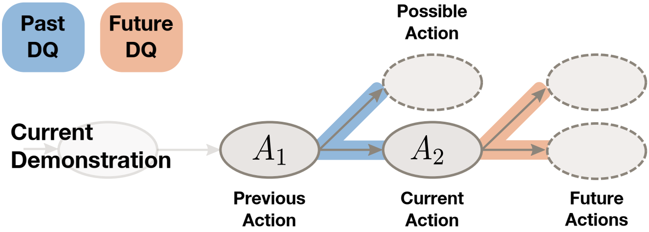
F: {never, sometimes, always}



Query Design

Disambiguation Queries aim to obtain ordering probability of an action with respect to a pair of actions.

After doing $\boxed{\mathbf{a}_{\mathrm{pre}}}$, do you prefer to do $\boxed{\mathbf{a}_i}$ or $\boxed{\mathbf{a}_j}$?



Query Design

These queries form the **query pool** $\mathcal{Q}$ of the learning agent.

Query Design

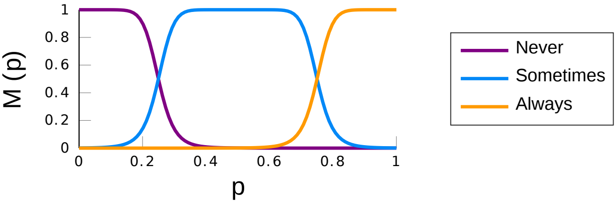
These queries form the **query pool** $\mathcal{Q}$ of the learning agent.

The core idea of our approach is that, given the query pool $\mathcal{Q}$, if we can select the **most informative queries**, the learning will be faster (than asking random questions).

Model Update

As we use a **Dirichlet-Multinomial** on the Markov Chain parameters, we want to compute the **posterior Dir**$(\theta|q, r)$.

We need to first map the **concepts of the questions/answers** back to **probabilities**.



Problem Statement

Query Design

Query Selection

Model Update

User Study

Discussion

Model Update

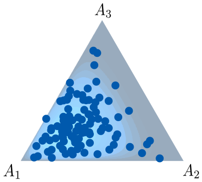
As we use a **Dirichlet-Multinomial** on the Markov Chain parameters, we want to compute the **posterior Dir**$(\theta|q, r)$.

Current Model

HYPERPARAMETERS

$$\mathbf{Dir}(T_1|\boldsymbol{\alpha})$$
$$\mathbf{Dir}(T_2|\boldsymbol{\alpha})$$
$$\vdots$$
$$\mathbf{Dir}(T_A|\boldsymbol{\alpha})$$
$$\mathbf{Dir}(\pi|\boldsymbol{\alpha})$$

Question q



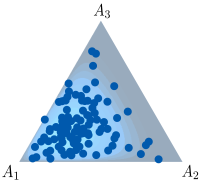
1. Sample the pre-query Dirichlet

Answer a

Model Update

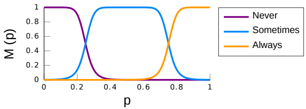
As we use a **Dirichlet-Multinomial** on the Markov Chain parameters, we want to compute the **posterior Dir**$(\theta|q, r)$.

Question q



1. Sample the pre-query Dirichlet

Answer a



$$w_s(q^*, r^*) = \begin{cases} M_f(s(a_{post})) & \text{if } r^* = \text{'yes'} \\ 1 - M_f(s(a_{post})) & \text{if } r^* = \text{'no'} \end{cases}$$

$$w_s(q^*, r^*) = M_{r^*}(s(a_1), s(a_2))$$

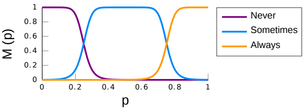
2. Filter/Weight samples based on answer

Model Update

As we use a **Dirichlet-Multinomial** on the Markov Chain parameters, we want to compute the **posterior Dir**$(\theta|q, r)$.

Problem Statement
Query Design
Query Selection
Model Update
User Study
Discussion

Answer a

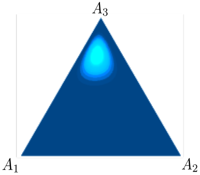


$$w_s(q^*, r^*) = \begin{cases} M_f(s(a_{post})) & \text{if } r^* = \text{`yes'} \\ 1 - M_f(s(a_{post})) & \text{if } r^* = \text{`no'} \end{cases}$$

$$w_s(q^*, r^*) = M_{r^*}(s(a_1), s(a_2))$$

2. Filter/Weight samples based on answer

Updated model

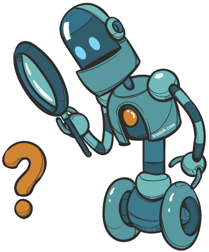


3. Fit the post-query Dirichlet

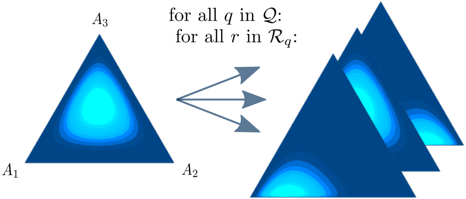
Query Selection

To select the **most informative** question, we need a measure of **information gain**.

We use the **entropy** $\mathbb{H}_q$ of the posterior distribution. For Dirichlet distributions, the entropy is always negative and decreases as the distribution becomes **more informative**.



Query Selection



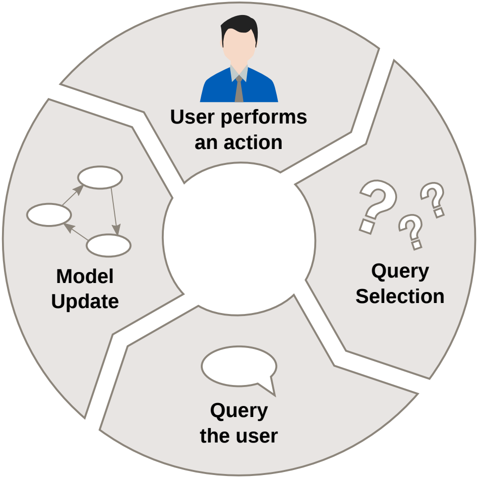
$$\Delta\mathbb{H}_q = \overbrace{\mathbb{E}_r[\mathbb{H}(\mathbf{Dir}(\cdot|q,r))]}^{\text{post query}} - \overbrace{\mathbb{H}(\mathbf{Dir}(\cdot|\boldsymbol{\alpha}))}^{\text{pre query}}$$

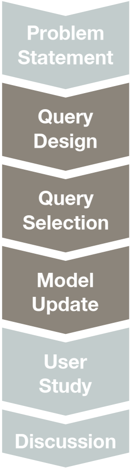
$$= \sum_r p(r|q)\mathbb{H}(\mathbf{Dir}(\cdot|q,r)) - \mathbb{H}(\mathbf{Dir}(\cdot|\boldsymbol{\alpha}))$$

$$q^* = \operatorname*{argmin}_q \Delta\mathbb{H}_q$$

Problem Statement
Query Design
Query Selection
Model Update
User Study
Discussion

Summarizing



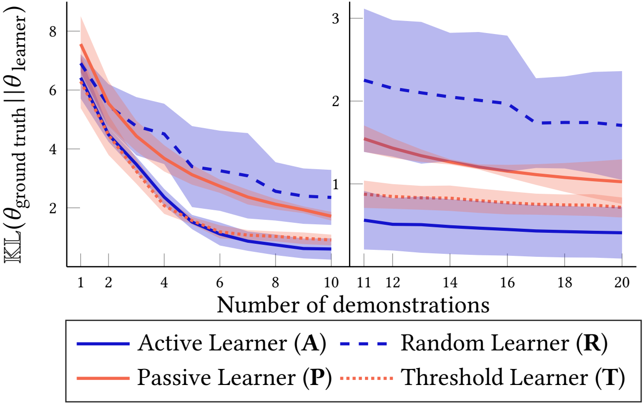


Simulations

Settings:

- Task with **9** actions available
- **4** preference patterns (ordering of actions)
- **20** demonstrations per pattern
- **4** learning strategies:
  1. Active Learner (**A**): proposed approach
  2. Passive Learner (**P**): LfD approach
  3. Random Learner (**R**): asks questions at random
  4. Threshold Learner (**T**): asks questions only if $\Delta\mathbb{H}_q < \tau$
- **72** questions to choose from at each selection step
- **No prior knowledge** before training (uninformative priors)

Simulations



Problem Statement

Query Design

Query Selection

Model Update

User Study

Discussion

Simulations

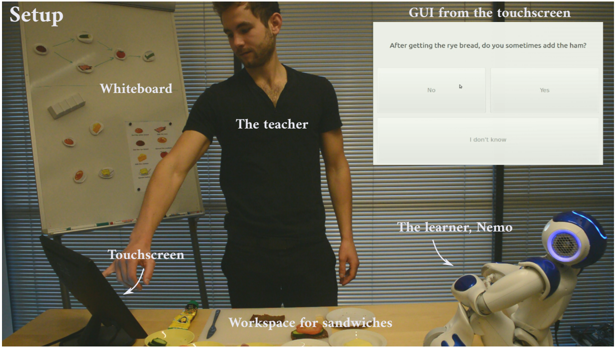
- **A** and **T** learn faster than **P**
- **T** $\approx$ **A**, while asking **59**% (first 10 demos) and **96**% (last 10 demos) fewer questions

Improvements:

- Introduce a *"I don't know"* answer!
- Provide **feedback** during the training, after each answer!

User Study

- Interactive learning of a cooking task: sandwich recipes
- Within-subject study: 3 conditions (**A**, **T** and **R**), 18 subjects

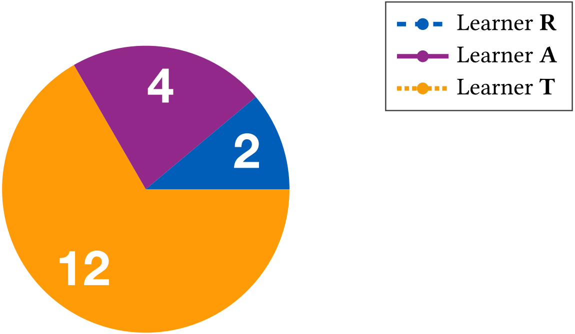


Questionnaire (1-7 Likert scale questions)

- **Perceived Performance:** How well do you think Nemo learnt the recipe (in percent)? (1 - *0%*, 4 - *50%*, 7 - *100%*)
- **Transparency:** While showing the recipe, was it clear to you if Nemo was learning the recipe? (1 - *Not clear at all*, 7 - *Extremely clear*)
- **Distraction:** Were Nemo's questions bothering or distracting you from your task? (1 - *Extremely distracting*, 7 - *Not bothering at all*)
- **Ease of Teaching:** How easy was it to teach Nemo the recipe? (1 - *Extremely difficult:*, 7 - *Extremely easy*)
- **Contextuality:** How in context were Nemo's questions with respect to your recipe steps? (1 - *Completely out of context*, 7 - *Extremely in context*)

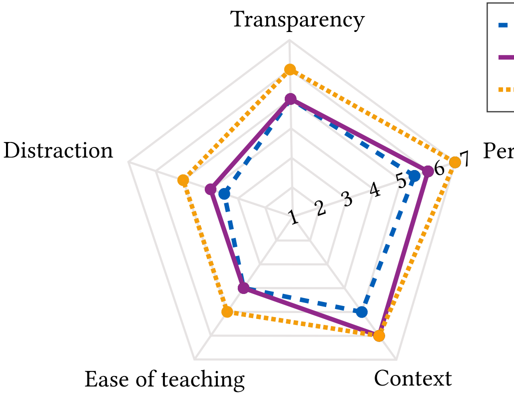
\+ **Post-experiment ranking of learning strategies**

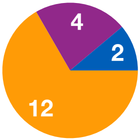
Ranking and Questionnaire scores



Problem Statement
Query Design
Query Selection
Model Update
User Study
Discussion

Ranking and Questionnaire scores





Problem Statement
Query Design
Query Selection
Model Update
User Study
Discussion

Perceived Performance and Transparency

- The **amount of questions** is the deciding factor
- Learner **T** reduced number of questions over time was perceived a sign of good learning and was used to decide when to stop the training
- Learners **A** and **R** asked too many questions

Ease of Teaching and Distraction

- Only **1.6%** of questions received the *"I don't know"* answer
- Subjects complained about Learner **A** and **T**'s tendency to pick **intricate or difficult** questions (especially questions expecting a **negative answer**)

Users' explanations of the query selection

- **R**'s questions were often perceived as *irrelevant* and *random*
- About questions targeting unseen actions
  - *"(Nemo) seemed to rule out uncommon options"*
  - *"Nemo wasted time on asking things I never did"*
- About repeated questions
  - *"(Nemo was) repeating questions and not learning much"*
  - *"(Nemo) seemed to confirm things by repeating questions instead of asking randomly"*

Conclusion

- **Active Robot Learning** can be used for learning temporal models interactively from non-expert users
- **Not only the query design but also the query selection must take into account the user**
  - Integrate user preferences regarding questions (positive answers, repeated queries) in the selection
  - Trade-off between **performances** and **quality of the interaction**

Problem Statement
Query Design
Query Selection
Model Update
User Study
Discussion

Thank you for the attention!

Mattia Racca and Ville Kyrki, *"Active Robot Learning for Temporal Task models,"* ACM/IEEE International Conference on Human Robot Interaction (HRI), 2018

Video available at `vimeo.com/mattiaracca/hri18`