MACQ: A Unified Library for Action Model Acquisition

Ethan Callanan[*1] **Rebecca De Venezia**[*1] **Victoria Armstrong**[1] **Alison Paredes**[1]
Jungkoo Kang[2] **Tathagata Chakraborti**[2] **Christian Muise**[1]

[1] Queen's University, [2] IBM Research AI
Contact: christian.muise@queensu.ca

Abstract

For over three decades, the planning community has explored countless methods for data-driven model acquisition. These range in *sophistication* (e.g., simple set operations to full-blown reformulations), *methodology* (e.g., logic-based -vs- planning-based), and *assumptions* (e.g., fully -vs- partially observable). With no fewer than 43 publications in the space, it can be overwhelming to understand what approach could or should be applied in a new setting. We present a holistic characterization of the action model acquisition space and further introduce a unifying framework for automated action model acquisition. We have re-implemented some of the landmark approaches in the area, and our characterization of all the techniques offers deep insight into the research opportunities that remain; i.e., those settings where no technique is capable of solving.

GitHub: https://github.com/QuMuLab/macq
Visualize: ibm.biz/macqviz
Demo: mulab.ai/demo/macq

1 Introduction

Within the planning community, the sub-field of automated action model acquisition has a rich history dating back to the late 80's (Shen and Simon 1989). Since then, several approaches to action model acquisition for planning have been introduced. The MACQ project seeks to (1) characterize this (still active) space of research; and (2) provide a unified framework/library for researchers and practitioners alike to use these technologies. The high-level philosophy of the MACQ project breaks the setting down into three distinct components:

1. **Trace Generation**: This includes various ways to parse and produce traces for action model extraction.
2. **Tokenization**: This allows us to morph traces into different formats for exploration (e.g., by adding noise to the fluent values or making states partially observable).
3. **Model Extraction**: This corresponds to the specific approaches introduced in the literature.

This demo is geared towards demonstrating the current capability of the MACQ library, and we detail some of the highlights in this extended abstract.

[*]These authors contributed equally.

```
from macq import generate, extract
from macq.trace import PlanningObject, Fluent, TraceList
from macq.observation import PartialObservation

def get_fluent(name: str, objs: list[str]):
    objects = [PlanningObject(o.split0[0], o.split()[1])
                for o in objs]
    return Fluent(name, objects)

traces = TraceList()
generator = generate.pddl.TraceFromGoal(problem_id=1801)
for _ in range(100):
    traces.append(generator.generate_trace())

observations = traces.tokenize(
    PartialObservation,
    percent_missing = 0.60
)
model = extract.Extract(
    observations,
    extract.modes.ARMS,
    upper_bound = 2,
    min_support = 2,
    action_weight = 110,
    info_weight = 100,
    threshold = 0.6,
    info3_default = 30,
    plan_default = 30,
)
print(model.details())
```

Figure 1: MACQ usage

2 MACQ in Action

Here, we briefly showcase some of the interface for the toolkit, as well as the summary view of the work.

2.1 Library Usage

Corresponding to the three main components detailed in Section 1 – trace generation, observation tokenization, and model extraction – the MACQ library offers a range of functionality for each. Usage of MACQ also follows this natural order of first generating or loading traces, applying tokenization, and then doing the model extraction.

Implemented Techniques	Learning Parameters											
	Model Features					Data Features						
	Uncertainty	Actions		Predicates		Fluent Observability		Predicate Information		Action Information		Trace
	Deterministic	Parameterized	Typing	Parameterized	Parameters Typed	Fully Observable	Noise	Parameterized	Typing	Parameterized	Typing	Partial
(Wang 1994)	1	1	1	1	1	0	0	0	0	0	0	1
(Yang, Wu, and Jiang 2005)	1	1	1	1	1	1	0	1	0	1	0	1
(Amir and Chang 2008)	1	1	0	1	0	1	0	1	0	0	0	0
(Zhuo, Peng, and Kambhampati 2019)	1	1	0	1	0	1	1	1	0	1	0	0

Table 1: MACQ already features a burgeoning set of acquisition techniques with varied characteristics. This table only refers to a small subset of features and only 4 of the several papers. For a full analysis of features, please refer to ibm.biz/macqviz.

```
Actions:
(communicate_soil_data waypoint
                       lander
                       rover
                       waypoint):
  precond:
    at rover waypoint
  add:
    at_rock_sample waypoint
    have_rock_analysis rover waypoint
    communicated_soil_data waypoint
    channel_free lander
    at_soil_sample waypoint
  delete:

(drop store rover):
  precond:
    have_image rover objective mode
    have_soil_analysis rover waypoint
    available rover
    calibrated camera rover
    at rover waypoint
  add:
    have_image rover objective mode
    calibrated camera rover
  delete:

 ...
```

Figure 2: Sample of the output from the ARMS extraction.

Figure 1 shows the code required to (1) generate traces for a problem found in the online repository at api.planning.domains, (2) tokenize by removing 60% of the fluents seen, and (3) apply the ARMS algorithm to extract potential actions (Yang, Wu, and Jiang 2005). A portion of the output is shown in Figure 2. All of the identified works in the space have been classified on the assumptions they make about the trace generation and action extraction process. A limited excerpt of this collection of meta-data is shown in Table 1.

As long as the type of tokens in a trace allows for it, various extraction techniques can be substituted and compared. Similarly, various data sources (from generative to pre-existing) can be used to seed the entire approach.

The MACQ library was built from the ground up to be (1) extensible and generalizable to all of the common model acquisition techniques; (2) serve as a rich resource for practitioners looking to apply model acquisition; and (3) provide a foundation for new research in the area of model acquisition.

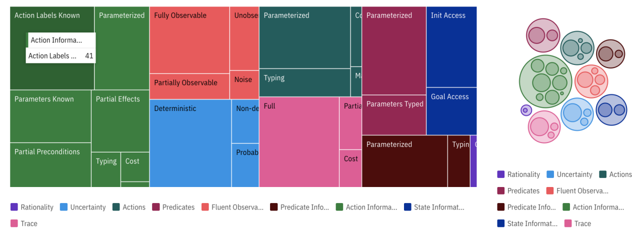

Figure 3: MACQ treemap view.

3 Visual Interface

The MACQ library also comes with a visual interface for users to explore the available papers on the topic through various lenses. The primary view provides a taxonomic account of the various topics identified in the field and how papers are classified along with those topics – for us, these topics correspond to the features of the model acquisition techniques considered. While a full discussion on the interface is beyond the scope of this demo, we highlight one aspect in Figure 3 – the treemap view of action model acquisition techniques characterized by their features.

4 Demo Logistics

An ICAPS demo of the MACQ initiative will involve mainly two key aspects: (1) a live demonstration of the library's functionality, and (2) an interactive visualization of the field.

Library Usage This aspect of the demo will include an interactive demonstration of the various trace generation, object tokenization, and model extraction techniques that have already been implemented in the MACQ library. Modifications of an existing Python script will be used to dynamically demonstrate MACQ's capabilities to interested demo goers.

Interactive Visualization The interactive visualization characterizes the space of action model acquisition research and includes four key elements: (1) a taxonomy of the features that capture each unique approach; (2) an "affinity" interaction to demonstrate how the papers cluster together (based on neural embeddings of the paper text); (3) the citation network of the relevant literature; and (4) automatically generated insights that dynamically suggest gaps in the literature. The final aspect was developed uniquely for this demonstration and serves as a guided recommendation on research yet to be explored.

References

Amir, E.; and Chang, A. 2008. Learning Partially Observable Deterministic Action Models. *Journal of Artificial Intelligence Research*, 33: 349–402.

Shen, W.; and Simon, H. A. 1989. Rule Creation and Rule Learning Through Environmental Exploration. In Sridharan, N. S., ed., *Proceedings of the 11th International Joint Conference on Artificial Intelligence. Detroit, MI, USA, August 1989*, 675–680. Morgan Kaufmann.

Wang, X. 1994. Learning Planning Operators by Observation and Practice. In Hammond, K. J., ed., *AIPS*.

Yang, Q.; Wu, K.; and Jiang, Y. 2005. Learning Actions Models from Plan Examples with Incomplete Knowledge. In Biundo, S.; Myers, K. L.; and Rajan, K., eds., *ICAPS*.

Zhuo, H. H.; Peng, J.; and Kambhampati, S. 2019. Learning Action Models from Disordered and Noisy Plan Traces. *arXiv:1908.09800*.