

Reviewer #1

I suggest to describe the overall concept of the learning approach first (see above for my hopefully correct summary) and to give the technical details later, rather than describing the technical details of compilation only without providing the overall picture. Going top down would make reading much easier. The second reason for revision is the comparison to ARMS. I suggest to compare with LOUGA that is better than ARMS in all aspects (runtime and quality of obtained models) so ARMS is no more state-of-the-art.

Page 3, Section 2.1: The definition of effects is strange. The authors define positive and negative effects as literals. What if a negated predicate (which is a literal) is among positive effects? Does it mean the same as having the positive version of the predicate among negative effects? That is very confusing.

Also, there are additional assumptions about the model, namely positive effects cannot be among preconditions while negative effects must be among preconditions. It would be useful to discuss these assumptions more as there are domain models that do not satisfy these assumptions. Do we really need them?

Page 4: There is a nice discussion about partial traces, but I have found it incomplete because the relation between actions and state partial observability is not fully clear. For example, if we have FO for actions, does it mean FO or PO* for states? Also PO* for states looks like a special case of PO based on the definition. I would suggest to include some discussion there.

Page 5, Table 2: I suggest including LOUGA there too.

Page 10, Section 4.2: $\text{on}(v1, v1)$ looks a bit strange. Do we really need such predicate with repeated variables?

Page 11, Section 4.3: Maybe this section fits better in the section Background. Do you use the assumption about having negative effects among preconditions and not having positive effects among preconditions like before? This is not clear from the definition of triggered.

Page 11, Section 4.4. This section is hard to understand, at least at the beginning. One needs to read it completely before understanding the details. I suggest to describe the concept at high level first and going to details then.

Page 11, Section 4.4. This section is hard to understand, at least at the beginning. One needs to read it completely before understanding the details. I suggest to describe the concept at high level first and going to details then.

Page 12: As I understand, the programming actions are building the model by adding preconditions and effects to the model. This immediately brings the question about symmetries as these programming actions can be used in any order (any permutation of them is fine). It would be useful to discuss this now. There are a few words about using SAT-based planner later, where parallel actions can be used. I think that this is critical for efficiency of the approach. Note also that the proposed method looks like a generate-and-test approach - we first generate a candidate model and then we verify that the model complies with plan traces. This does not seem particularly efficient.

Page 13: There is implication used in the precondition. It was never mentioned before that a general formula can be used.

My major problem here is that it is not clear how the actions validating states interleave with actions applying the learned actions. This must be clarified as the states are generated by actions.

Page 20: In the graphs, it is not clear how the values are obtained. Are these average values between the domains or mean values or is it just for one selected domain (which?). The graph with runtimes show what I was afraid about - the efficiency degrades fast with larger example plans. This raises the question if longer plans can be used if needed for some domains.

Reviewer #2

It took me a while to figure out that the authors focus on a "consistency" kind of problem, in which the task is to find a model that is consistent with the observations. This is given in section 4.1, but I would liked an organized problem definition. This is important because throughout the paper you talk about soundness, completeness and computational complexity of the problem, but this only makes sense w.r.t a clear problem definition.

For example, the authors discuss in section 3 the complexity of the

learning task with different assumptions, but this is done before the problem is defined. Also, all the complexity of the results mentioned in page 8 are given without a proof, and the only reference in this context is to the Russell and Norvig book. The authors write in page 8 that FAMA solves the learning problem by compiling it to a planning problem, and consequently the planning problem is P-SPACE complete, but this is not a proof. In fact, it seems like this is an incorrect reduction: if planning is P-SPACE complete and learning can be solved by planning, it does not mean that learning is P-SPACE complete, as it may be easier.

There are several cases where authors repeat themselves several times. Here are a few examples:

- (page 2) "While current learning systems . none of them allow partial observability in the sequence of executed actions "

Two paragraphs later:

- " unlike most relevant action-model learning algorithms, FAMA does not require the traces to contain any observed action "

(Page 8) "FAMA represents one step ahead without assuming observed actions."

- A STRIPS action model is defined twice: once in section 2 and another time in section 4.

- The syntax-based notions of false positives and false negatives are defined twice: once in the second paragraph of section 5, and twice in Def. 4-5. In fact, most of page 17 seems redundant to me, and the text should be merged with that in the previous page.

Surprisingly, adding more traces causes the precision and recall of FAMA to drop for the NO action NO states setting. The authors say this is because more timeouts were observed due to the complexity of the model learning planning problem. To support this, the authors should report the number of timeouts for the different number of traces. In particular, since we do not see this drop in precision and recall in the other settings, I expect a comparison of the number of timeouts as a function of the number of traces for both settings. If we see that the number of timeouts for this setting is larger than the simpler setting, that would support the authors conclusion.

It was a bit weird to me that the authors chose to compile to a planning and not to a SAT problem like many prior work on learning action models.

One reason to compile to planning is when the length of the solution plan is unknown, but the authors say that this is known in this case as well (see my question on this later).

From my understanding of this work, there's a variable for every tuple of action name, fluent, and precondition/add-effect/delete-effect option.

Then, there are constraints to hold that all this fall together in a consistent way with the observed trajectories.

So, my question is: can the authors motivate the choice of planning over SAT?

Since planning problems can be solved by compilation to SAT, and SAT problems can be solved by compilation to planning, it does not really matter to which problem you compile to, but still, I wonder what motivated the authors to take this approach.

Page 1. In general, the introduction left me very confusing, and I only really understood it after reading the paper. I wish the authors could do a better job in having the introduction not rely on understanding the rest of the paper so much. Concretely, talking about FAMA as a "solving scheme" "whose solution must be compliant with the input plan traces" did not make sense to me in the first pass.

Page 1. " an abstract version of the capability model that reflects " I am not sure that "abstract" is the right term here, since the learned model is not an abstraction of the real-world but an approximation of it.

Page 1. "Two different types of data are generally identified in the generation of explanations (2) data that do not accurately represent the decision-making process because the observed external behavior of the agent responds to a black-box model." First, I think you meant "corresponds" and not "responds". Second, if the agent is treated as a black box, then this is not really a way to generate explanation. Please better explain the text after (2) .

Page 2. " the synthesis of different kinds of generative models with classical planning " I do not understand this sentence. What do you mean by generative models in this context?

Page 2. "FAMA is a solving scheme " FAMA is a learning scheme, not a solving scheme. I agree that it is a learning scheme that learns by solving a planning problem.

Page 2. " by the input observed actions, " What is "input observed actions"? please rephrase.

Page 3. "the open world assumption" Either explain what this means or provide a reference.

Page 3. "the minimal action sequence to transit from state s_i to s_{i+1} is composed of a single action" This can be stated in a simpler way: "there exists a single action that transitions s_i to s_{i+1} "

Page 4. "A plan trace . For a planning frame . holds that for every action in and that " this sentence is not clear. Please rephrase.

Page 4. The term action header is not defined.

Page 4. " of the evaluation method to validate " missing "used" before "to"

Page 4."Exceptionally, a NO state sequence in LOCM is a fully-empty trajectory, with neither initial or final state. " I don't understand this sentence. Isn't a NO state sequence a fully empty trajectory for all algorithms? Also, it should be "nor" and not "or".

Page 4."Most approaches assume that a set of predicates and a set of action headers are provided alongside the input traces" What about FAMA? I think it also requires this.

Pages 5-7. When describing prior work, it is sometimes not clear when you describe the algorithm and when you describe their specific implementation and experiments. Specifically:

1. When describing SLAF, you write that 1,000 action-observation sequences and 10 fluents are selected. Is this part of the algorithm or just how they specifically implemented it in their experiments?
2. When talking about LAMP, you write that it allows trajectories up to a "minimum percentage of 1/5 " again, is this requirement mandatory in their algorithm, or is it just what they happened to get in their experiments? Also, if you write 1/5 then you should write "ratio" and not "percentage".
3. The same question applies to your description of CAMA, when talking about 80% empty states.

Pages 5-7. " the state of a sort" what is a "sort"?

Pages 5-7. "As many of the approaches in section 2 assume, there may be an " this text is redundant. It just repeats again what was said twice already.

Page 8." an unbound number of missing actions " should be "unbounded'

Page 8.I disagree that having FO implies having human observers annotating the traces. For example, if one monitors a robot performing actions by monitoring the communications between the robot and its controller.

Page 8."When the plan traces is fully observed, learning STRIPS action models is straightforward" The reference is for work done in 2018, while learning an action model in this setting has been discussed in IJCAI 2017 by Stern and Juba (Safe, model-free learning of action models").

Page 8.The use of syntax and semantics in this stage was confusing to me. I am not sure how to improve this, though.

Pages 9-13

Pages 9-13

Pages 9-13

Pages 9-13

Reviewer #3

First sentence of the Introduction: there is no satisfactory or generally accepted definition of complete domain model, given that a domain model is an abstraction of the domain being modelled. If the authors use such a term they should make clear what their definition is. Adequate is a better word to use - in the sense of being adequate enough for use with a planner to solve a set of planning tasks.

In this part and the rest of the paper I do not fully agree with the authors
hype

about FAMA with respect to the input information required. The minimal amount of

information that is needed (if I have got this right) is to

specify every predicate that will be needed in the application (this is done in the initial state)

the number of steps that are in the training plan, the set of all action headers including their

list of parameters, and a goal state which refers to the same predicates as the initial state.

Comparing this to, say, LOCM, where no state/predicate information at all

needs to be specified, I would say that in fact LOCM requires a lot less information

a priori. What FAMA does not require is the identity of all the actions at each step,

nor the instantiation of their headers, but it does require the definition of all

the predicates that may appear in any state. The truth it seems to me is that

FAMA would be good for some applications which fit this set of constraints, which is

a different set of constraints than for other domain model learners.

A way forward might be to put the claims about FAMA into a smaller set of learning systems

(certainly ARMS will be in that set) where it clearly needs less input information.

I totally agree with the authors in their assertion that the evaluation methods for

previous model learners is currently poor, and amount to how close the learned model

is to a GTM. I am not convinced, however, that the use of

the statistic methods proposed is a great improvement. Precision and Recall

are heavily used in data mining where (generally) there are huge data sets and there

is not generally a hand crafted model for comparison. In planning, however, the idea of

separating out a set of traces into training and testing appears odd, especially where

it is known that the learning will eventually create a model that explains

all the training
traces. This seems particularly odd if there are only a few traces to be
split up (and FAMA appears
only able to cope with a few traces because of complexity problems).

- The method of learning: This is to compile the learning problem into a
conditional-operator
classical planning problem. This method appears to be highly innovative
to this line of research
(stemming from an earlier ICAPS paper by the authors).
The description of the compilation method did not appear to me to be
at the level
of the detail to allow easy reading. It took me a good deal of time to
work out
what going on. This is a complex process and I recommend much more
use of examples
throughout - perhaps a running example to show the full run of a com-
pilation.
It might also improve the paper if an algorithmic description of compi-
lation
were included? One aspect that was not obvious from the description
was how a
SET of models are delivered by planning rather than just one; this needs
spelling out.

- The evaluations of the domain model using classification metrics of pre-
cision and recall:
I like the distinction of the methods labelled "syntactic" (using the
GTM) vs "semantic" here;
however I think the use of the word - "semantic a bit misleading. This
implies
taking into account the meaning of the features within the application
domain which are
left out or in erroneously compared to the GTM e.g. that a missing
"on(x,y) in a learned model
is somewhat better/worse from a missing holding(x) - because of their
meaning, rather than
counting each predicate as 1.
In the end your method reverts to counting the number of different
predicates, but in the context
of the recall/precision formula with respect to domain model(s) that are
consistent with the test
sets, not only the GTM.
The definition of the metrics was not too clear. Intuitively, I think I

understood 5.1 but it

was not well stated -the definition of some terms seemed syntactically wrong or at least sloppy.

E.g. in the definition of INS and DEL, should not the universally quantified variables be further

qualified by stating that there is a union of the already unioned sets?

Or again, should there not

be an extra "Sigma" to represent the addition of the addition of the number of precons and

effects in size(M) definition? In proposition 7 you say the closest set "is the GTM"

do you mean equal to the singleton set of the GTM?

Also - the structure of 5.1 - it would be best to clearly state that

the syntactic evaluation is going to be introduced, then as a further subsection the semantic

evaluation.

- I advise changing title Experimental Results => Experimental Evaluation

- At the beginning of the evaluation you state the obvious - that the purpose is to evaluate

the performance of FAMA and the quality of the models. At this stage to clarify the experimental

section I suggest that you indicate, before detailing the set up, what would be success, for example

how do you intend to show that your technique is better than or comparable to others? This will

no doubt mean introducing the metrics you intend to use and giving an idea of success

in terms of them. Also it may mean taking text from the start of the sections 6.2 - 6.5

into the first part of the evaluation section. This will however give a good rationale for the

'set-up, the choice of the particular experimental scenarios, and for the discussion of the

results.

- Why "14" models? please give some explanation (e.g. there are only 14 strips models there)

so we know why/if you have left any out

was the max predicate arity of "2" in all of the 14 a big issue? Given the inherent computational problems in planning I guess so?

so for the NO / NO case study, we know the full IS , GS, the number of action steps,
and the space of possible action headers for each step - is that correct?
If so please make it more explicit - it not say explicitly what it means.
This was not very explicit in the introduction/compilation process description either.

Typos spotted
around a hundred of tracesp.20
Please replace fairly sound and almost totally complete with something more precise. p.21
reason why the scores of p.26