Unsupervised Grounding of Plannable First-Order Logic Representation from Images

Masataro Asai

IBM Research

Recently, there is an increasing interest in obtaining the relational structures of the environment in the Reinforcement Learning community. However, the resulting "relations" are not the discrete, logical predicates compatible to the symbolic reasoning such as classical planning or goal recognition. Meanwhile, Latplan [Asai 18] bridged the gap between deep-learning perceptual systems and symbolic classical planners. One key component of the system is a Neural Network called State AutoEncoder (SAE), which encodes an image-based input into a propositional representation compatible to classical planning. To get the best of both worlds, we propose First-Order State AutoEncoder, an unsupervised architecture for grounding the first-order logic predicates. Each predicate models a relationship between objects by taking the interpretable arguments and returning a propositional value. In the experiment using 8-Puzzle and a photo-realistic Blocksworld environment, we show that (1) the resulting predicates capture the interpretable relations (e.g. spatial), (2) they help obtaining the compact, abstract model of the environment, and finally, (3) the resulting model is compatible to symbolic classical planning. **This paper is an extended abstract of a paper accepted in *International Conference on Automated Planning and Scheduling, Planning and Learning Track (2019)*. We cut out most of the details to meet the space requirement. For details/citations please refer to the original material.**

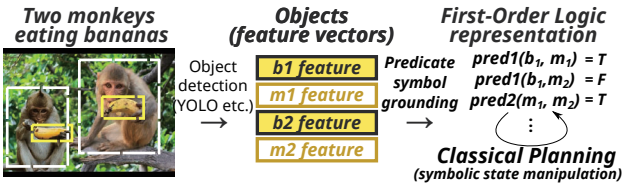

図 1: *Predicate symbol grounding* (PSG) process for identifying the predicates and obtaining the First Order Logic (FOL) representation of the environment for symbolic reasoning. In this example, an anonymous binary predicate $pred_1$ can be interpreted by humans as something like $eating(object, subject)$.

1. Introduction

Recent success in the latent space classical planning [Asai 18, Latplan] shows a promising direction for connecting the neural perceptual systems and the symbolic AI systems. Latplan is a straightforward system built upon a state-of-the-art Neural Network (NN) framework (Keras, Tensorflow) and Fast Downward classical planner [Helmert 04]. It builds a set of propositional state representation from the raw observations (e.g. images) of the environment, which can be used for classical planning as well as goal recognition [Amado 18]. However, Latplan still contains many rooms for improvements in terms of the interpretability and the scalability which are trivially available in the symbolic systems. An instance of such limitations of Latplan is that the reasoning is performed on a propositional level, missing the ontological commitment of the First-Order Logic (FOL) that *the world comprises objects and their relations* [Russell 95].

FOL is a *structured* representation, which offers some extent of interpretability compared to the *factored* representation of propositional logic formula [Russell 95]. Even if the predicate symbols discovered by a *Predicate Symbol Grounding* system (Fig. 1) are machine-generated anonymous symbols (not the human-originated symbols assigned by manual tagging), the structures help humans interpret the meaning of the relations from the several instances of the argument list (objects) that make the predicate true. For example, when two propositions *pred*(0,1) and *pred*(1,2) are true, we can guess the meaning of *pred* as `+1`, or given *pred*(*monkey*, *banana*) being true, the meaning of *pred* would be something like *eating* or *holding*. This is impossible in a propositional representation where only the variable indices and the truth values are known.

In this paper, we propose First-Order State AutoEncoder (FOSAE, Fig. 2), a NN architecture which, given the feature vectors of the objects in the environment, automatically learns to identify a set of predicates (relations) as well as to select the appropriate objects as the arguments for the predicates. The resulting representation is compatible to classical planning. We do not address the object recognition problem, whose task is to extract the object entities from a raw observation. We rather assume that they are already extracted by an external system and converted into the feature vectors, given the recent success of object detection methods like YOLO [Redmon 16] in image processing. While FOSAE is in principle data-format (e.g. images, text) independent, we focus on the image-based input in this paper.

FOSAE provides a higher-level generalization and the more compact model by adding a constraint that the extracted relations are common to multiple tuples of objects. Ideally, predicates model the commonalities between the multiple instantiations of its arguments, rather than rote learning some unrelated combinations. In order to discover such predicates, our framework ensures that a single predicate is applied to the different arguments *within* the same observation. Otherwise, the network may choose to apply them to the same or the very narrow combinations of arguments in every observations, resulting in an inflexible predicate that just remembers some combinations. Since the weights used to model the predicates are utilized multiple times, this also reduces the number of weight parameters required to model the environment.

2. Related Work

Recently, there are increasing interest in the effectiveness of finding "relations" in Deep Reinforcement Learning [Mnih 15, Zambaldi 18, Battaglia 18, DRL] community. In this paper, we address the following issues in these work:

Human Supervision. Providing a relational dataset as an input (as in [Battaglia 18] and neural theorem proving), or a probabilistic

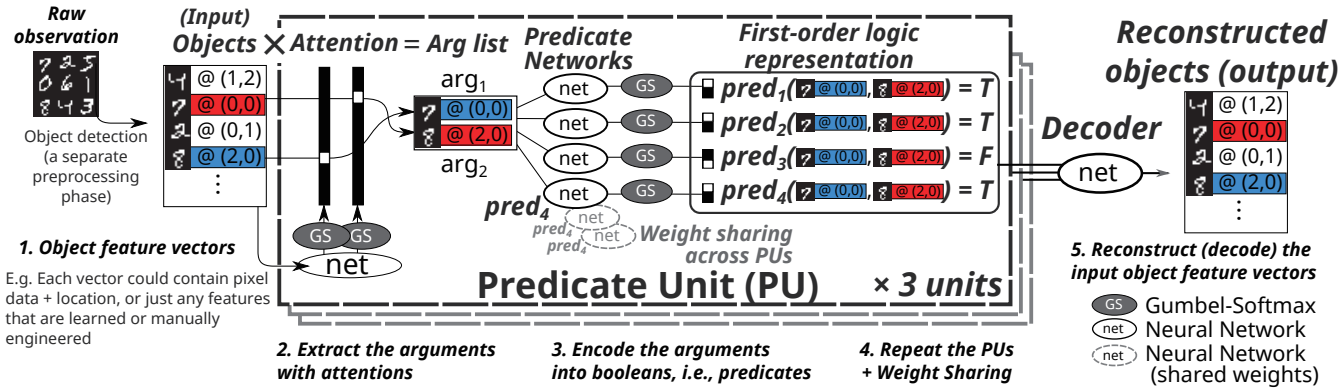

図 2: A First-Order State AutoEncoder (FOSAE) with $P = 4$ predicates, arity $A = 2$, and $U = 3$ Predicate Units. In this example, a feature vector consists of the pixel values and the (x, y) location of an 8-Puzzle tile.

logic program containing predicate symbols which defines a network, exhibits the knowledge acquisition bottleneck as the predicates are grounded by humans and thus the system relies on human knowledge.

Compatibility to the symbolic systems. Relational structures in existing work do not return explicit boolean values even when the environment is deterministic, fully observable and discrete in nature. This makes them incompatible to symbolic systems such as classical planners or goal recognition. Ideally, systems should guarantee that a discrete environment is represented in a discrete form, and numeric variables (such as those handled by numeric planner) should be introduced only when necessary.

Interpretability. Some networks use real-valued soft attentions (probability) to model the objects that take part in a relation, which are similar to the predicate arguments. However, the relations resulted from soft attentions are hard to interpret due to the ambiguity, e.g. "Bob has-a '50% dog and 50% cat"' in a "has-a" relation. Continuous outputs of the relational structures are also difficult to interpret.

Scalability for higher arities. Some work assumes the binary relations and enumerates $O(N^2)$ pairs for N objects. The explicit structure is impractical for larger arities A because the network size $O(N^A)$ increases exponentially.

3. High-Level Overview

In order to find a first-order logic representation of the environment from raw data, we perform the following processes (Fig. 1): (1) *Object detection* identifies and extracts a set of regions from the raw data that contain objects. (2) *Predicate symbol grounding* (PSG) finds the boolean functions that take several object feature vectors as the arguments.

While both processes are nontrivial, there are significant advances in (1) recently. Object recognition in computer vision e.g. [Redmon 16, YOLO], or named entity (noun / "objects") recognition [Nadeau 07, Mohit 14] in Natural Language Processing, are both becoming increasingly successful. In this paper, therefore, we do not address (1) and use a dataset that is already segmented into image patches and bounding boxes. In principle, we could extract the object vectors with these external systems.

Next, PSG identifies a finite set of boolean functions (predicates) from the feature input, by learning to select the argument list from the input and detecting the common patterns between the objects that define a relation. As a result, we obtain the first-order logic representation of the input as a list of FOL statements such as $pred_2(obj_1, obj_2)$=*true*, where the system automatically learns to extract the arguments from the inputs, and also decides the semantics of the predicates by itself, in an unsupervised manner.

We now introduce the core contribution of this paper, First-Order State AutoEncoder (FOSAE, Fig. 2), a neural architecture which performs PSG and obtains a representation compatible to symbolic reasoning systems such as classical planners.

(Fig. 2, 1) Overall, the system follows the autoencoder architecture that takes feature vectors of multiple objects in the environment as the input and reconstructs them as the output. The form of the feature vector for each object is entirely problem/environment dependent: It could be a hand-crafted feature vector, a flattened vector of the raw pixel values for the object, or a latent space vector automatically generated from the image array by an additional feature learning system (such as an autoencoder).

FOSAE consists of multiple instances of *Predicate Unit*, a unit that (1) learns to extract an argument list from the input and (2) computes the boolean values of the predicates given the extracted argument list. The number of units U, the arity of predicates A and the number of predicates P are hyperparameters which should be sufficiently large so that the network can encode enough information into a boolean vector and then reconstruct the input. If the network does not converge into a sufficiently low reconstruction loss, we can increase these parameters until it does. How to run this iteration efficiently is a hyperparameter tuning problem which is out of the scope of this paper.

(Fig. 2, 2) In order to extract the arguments of the predicates, we use multiple attention networks. The use of attention avoids enumerating $O(N^A)$ object tuples for N objects as was done in the previous work. There are A attentions in each PU, thus each PU extracts A objects from the N objects in the input. With U PUs, there are $U \times A$ attentions.

An attention network is implemented as a 2 fully-connected networks ending with a Gumbel-Softmax activation. Unlike previous work which uses a Softmax in the output, where the attention vectors take the continuous probability values produced by Softmax, we instead use Gumbel-Softmax which converges to a discrete one-hot vector so that the meaning of the extracted objects are

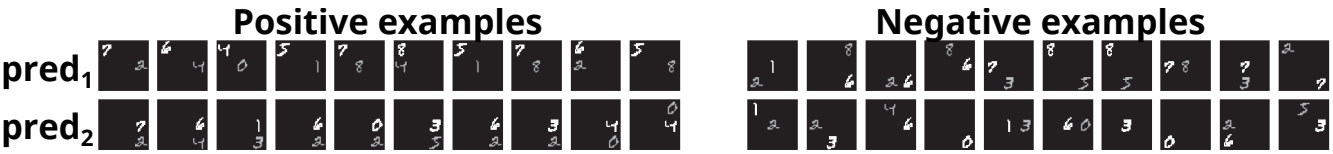

図 3: The positive/negative examples of the arguments for the first 6 predicates of $(U, A, P) = (25, 2, 50)$. The first/second argument is visualized in white / gray.

clear. For example, if an attention vector for an argument takes a value $(0, 1, 0)$, it is clearly extracting the 2nd object in 3 objects, while if it were $(0, 0.5, 0.5)$, it is unclear what was selected.

(Fig. 2, 3) Next, in each u-th PU, a set of NNs called *Predicate Network* (PN) using Gumbel-Softmax takes the arguments $\boldsymbol{g}_u = (\boldsymbol{g}_{u1} \dots \boldsymbol{g}_{uA})$ and outputs a discrete 1-hot vector of 2 categories, which means true if the first cell is 1, and false otherwise. There are P PNs where each PN $pred_p$ $(1 \leq p \leq P)$ returns a single boolean value and models a first order predicate $pred_p(\boldsymbol{g}_{u1} \dots \boldsymbol{g}_{uA}) \in \{0, 1\}$. The boolean values have the same role as the representation discovered by the propositional SAE.

(Fig. 2, 4) Attentions and PNs form a single PU. We repeat such PUs U times, which results in $U \times P$ total propositions. While the weights in the attention functions (att_{ua}) are specific to each PU, the PN weights for $pred_p$ are shared across PUs (hence it lacks the subscript u here). This makes the boolean function $pred_p$ in different PUs identical to each other, and force them to learn a common relations among the different arguments because PNs take different arguments in each PU.

(Fig. 2, 5) Finally, the input object vectors are reconstructed from the propositional representation by concatenating the boolean outputs from all PUs and feeding them to the decoder.

4. Modeling 8-Puzzle Instances

In order to evaluate FOSAE, we created a toy environment of 8-puzzle states using the feature vectors shown in Fig. 5. Each feature vector as an object consists of 15 features, 9 of which represent the tile number (object ID) and the remaining 6 represent the coordinates. Each data point has 9 such vectors, corresponding to the 9 objects in a single tile configuration. We generated 20000 transition inputs (state pairs) which are divided into 18000 (training set) and 2000 transitions (test set).

Previous work on relational structures have not yet provided evidence that they actually help modeling the environment and extract the abstract knowledge. For example, it is possible that even if a relational structure like RN [Santoro 17] extracts multiple arguments, the succeeding layers may ignore some arguments by assigning zero weights, essentially modeling just unary predicates (i.e. attributes) rather than the structural relationships.

We made the contour plots (Fig. 6) of the reconstruction errors for the test set with various U, P, A, and compared their Pareto fronts. For the same (U, P) pair, the size of the bottleneck layer (propositional vector) is $U \times P$ regardless of A, which makes the direct comparison between different A feasible. We see that the arity plays a critical role in finding the more compact information, demonstrating that structural relations contribute to building an abstract representation.

We also compared the number of trainable parameters (weights) in the network because for the same (U, P), the larger arity means the larger number of parameters in the networks which may help the training. Table 1 shows the models with the fewest parameters among those achieved the reconstruction error ≤ 0.1 for each A.

A	U	P	Propositions	Trainable parameters
1	18	5	90	287343
2	9	6	**54**	**268273**
3	9	7	63	303302
9	1	171	171	811828
SAE (Asai 2018)			18	3404467

表 1: Configurations $(U, P) \in [1, 20]^2$ for each A that achieved the reconstruction error ≤ 0.1 with the smallest trainable parameters.

Next, we show how the hard attentions make the predicates interpretable through visualization. In principle, we can visualize the objects in the images selected by the attentions (e.g. monkeys, bananas in Fig. 1) using a decoder function that reconstructs the regions from feature vectors. Fig. 3 shows the visualizations of the arguments given to the predicate networks. Each subfigure is a visualization of an argument list vector $\boldsymbol{g}_u = (\boldsymbol{g}_{u1}, \boldsymbol{g}_{u2})$ randomly sampled from the dataset. We humans could recognize the patterns that are shared on the left hand side (positive examples) of each row, which is not available in the propositional representation.

5. Evaluating Classical Planning Capability

We show that the FOL representation generated by FOSAE is a feasible and sound representation for classical planning.

We tested the FOSAE-generated representation with AMA_1 PDDL generator [Asai 18] and the Fast Downward [Helmert 04] classical planner.

5.1 8 Puzzle

Omitted due to space.

5.2 Photo-Realistic Blocksworld

The dataset generator produces a 300x200 RGB image and a state description which contains the bounding boxes (bbox) of the objects. Extracting these bboxes is an object recognition task we do not address in this paper, and ideally, should be performed by a system like YOLO [Redmon 16]. We resized the extracted image patches in the bboxes to 32x32 RGB, flattened it into a 3072-D vector, and concatenated it with the bbox vector. The bbox vector is 200-dimensional and is generated by discretizing (x_1, y_1, x_2, y_2) by 5 pixels and encoding it as a 1-hot vector (60/40 categories for each x/y-axis), resulting in 3072+200=3272 features per object.

We then solved 30 planning instances with 3 blocks, generated by taking a random initial state and choosing the goal states by the

図 4: (**middle**) The initial/goal state of a Blocksworld instance. (**right**) The solution to this problem reconstructed from the latent vector. It unpolishes the red cube, then moves the cylinder, the red cube, the yellow cube and then polishes the yellow cube.

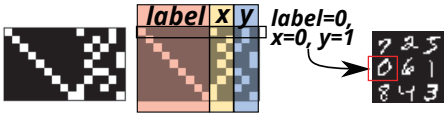

図 5: A single 8-puzzle state as a 9x15 matrix, representing 9 objects of 15 features. The first 9 features are the tile numbers and the other 6 features are the 1-hot x/y-coordinates.

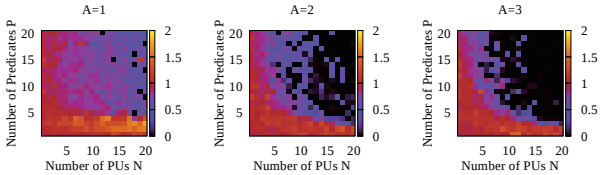

図 6: Contour plots of the reconstruction error of the test set for A=1,2,3, $(U, P) \in [1..20]^2$. It shows that the larger arity helps learning the compact representation.

3, 7, or 14 steps random walks (10 instances each). The system correctly solved all instances, where the correctness of the plans are checked manually. Fig. 4 shows an example solution generated from the intermediate states of the plan.

6. Discussion and Conclusion

We proposed First-Order State AutoEncoder, a neural architecture which grounds/extracts first order logical predicates from the environment without human supervision. Unlike any existing work to our knowledge, the training is fully automated (no manual tagging / no predefined reinforcement signals) and the resulting representation is interpretable, verifiable and compatible to symbolic systems such as classical planners.

We do not claim that we fully solved the FOL generalization because the learned FOL statements are quantifier-free, grounded representation that is essentially equivalent to the propositional statements. However, this work is an important step toward the full FOL generalization including quantification because quantifying a FOL formula requires a set of predicate symbols in the first place.

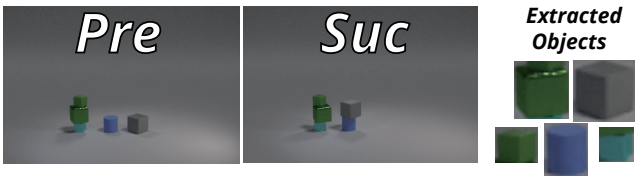

図 7: An example Blocksworld transition. Each state has a perturbation from the jitter in the light positions and the ray-tracing noise. Other objects may intrude the extracted regions. Objects have the different sizes, colors, shapes (cube or cylinder) and surface materials (metal or rubber).

参考文献

[Amado 18] Amado, L., Pereira, R. F., Aires, J., Magnaguagno, M., Granada, R., and Meneguzzi, F.: Goal Recognition in Latent Space (2018)

[Asai 18] Asai, M. and Fukunaga, A.: Classical Planning in Deep Latent Space: Bridging the Subsymbolic-Symbolic Boundary, in *Proceedings of AAAI Conference on Artificial Intelligence* (2018)

[Battaglia 18] Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al.: Relational inductive biases, deep learning, and graph networks, *arXiv preprint arXiv:1806.01261* (2018)

[Helmert 04] Helmert, M.: A Planning Heuristic Based on Causal Graph Analysis, in *Proceedings of the International Conference on Automated Planning and Scheduling(ICAPS)*, pp. 161–170 (2004)

[Mnih 15] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al.: Human-Level Control through Deep Reinforcement Learning, *Nature*, Vol. 518, No. 7540, pp. 529–533 (2015)

[Mohit 14] Mohit, B.: Named Entity Recognition, in *Natural language processing of semitic languages*, pp. 221–245, Springer (2014)

[Nadeau 07] Nadeau, D. and Sekine, S.: A Survey of Named Entity Recognition and Classification, *Lingvisticae Investigationes*, Vol. 30, No. 1, pp. 3–26 (2007)

[Redmon 16] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788 (2016)

[Russell 95] Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., and Edwards, D. D.: *Artificial Intelligence: A Modern Approach*, Vol. 2, Prentice hall Englewood Cliffs (1995)

[Santoro 17] Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., and Lillicrap, T.: A simple neural network module for relational reasoning, in *Advances in neural information processing systems*, pp. 4967–4976 (2017)

[Zambaldi 18] Zambaldi, V., Raposo, D., Santoro, A., Bapst, V., Li, Y., Babuschkin, I., Tuyls, K., Reichert, D., Lillicrap, T., Lockhart, E., et al.: Relational Deep Reinforcement Learning, *arXiv preprint arXiv:1806.01830* (2018)