HUZAR: Predicting Useful Actions with Graph Neural Networks

Piotr Rafał Gzubicki, Bartosz Piotr Lachowicz, Álvaro Torralba

Aalborg University, Aalborg, Denmark
piotr.gzubicki97@gmail.com, jendrekpb@gmail.com, alto@cs.aau.dk

Abstract

We present HUZAR, a participant of the learning track of the International Planning Competition 2023. HUZAR focuses on simplifying the planning task by predicting a set of useful actions with Graph Neural Networks before the search starts.

Introduction

An important challenge for solving planning tasks is that the number of possible actions grows with the number of objects that are being dealt with, even though most of those actions will not be used in a plan. Therefore, planning can be more efficient if one focuses on a sub-set of relevant actions that are sufficient to find a plan. While most approaches that remove irrelevant actions (Haslum 2007; Haslum, Helmert, and Jonssson 2013; Alcázar and Torralba 2015; Torralba and Kissmann 2015; Fišer, Torralba, and Shleyfman 2019) focus on identifying a set of actions that can be removed without compromising plan solvability and optimallity, one can also discard actions more greedily (Nebel, Dimopoulos, and Koehler 1997; Heusner et al. 2014). If a plan exists with the remaining actions, it can be found more efficiently.

HUZAR is inspired on partial grounding approaches (Gnad et al. 2019), which used Machine Learning methods to predict what actions to ground. However, HUZAR relies on fully grounding the task, and analyzing the entire grounded task in order to decide what actions are truly relevant. HUZAR aims to leverage the capabilities of Graph Neural Networks (GNNs) (Scarselli et al. 2009) to predict the set of useful actions within a given task. Due to their ability to process data represented as a graph, GNNs have found diverse applications, including the modelling of social networks (Wu et al. 2020), physical systems in natural science (Sanchez-Gonzalez et al. 2018), and protein-protein interface networks (Fout et al. 2017).

This work is an outcome of the project undertaken by our study group with the guidance of the supervisor during the 10th semester of the Computer Science program at Aalborg University. The primary objective of this project was to develop an advanced tool capable of improving any automated planner by using a preprocessor that leverages the cutting-edge technology of GNNs.

Approach

Graph Representation

We consider planning tasks in the SAS$^+$ formalism (Bäckström and Nebel 1995), which can be automatically obtained from the PDDL representation (Helmert 2009). We represent each planning task as a Problem Description Graph (PDG) (Pochter, Zohar, and Rosenschein 2011). The PDG has three types of nodes: *variable*, *value*, and *action* nodes.

The graph contains edges connecting each variable to all its values, as well as each value to each action where the value appears in the precondition or the effect. Unlike Pochter, Zohar, and Rosenschein, who considered two separate nodes for the preconditions and effects of each action, we have a single node per action and distinguish preconditions and effects by using different types of edges (Shleyfman et al. 2015).

Value nodes contain two Boolean features that specify if the corresponding atom belongs to the initial state and/or the goal, respectively. The framework is easily extensible with other types of features, but this is left as future work.

The goal is to predict what actions are useful. Therefore, each action node contains a target label, specifying whether it is a useful action for achieving the goal from the initial state or not. The GNN will obtain a representation for each node, that will be used to predict each target class.

Learning Phase

In the learning phase, HUZAR, takes as input a set of planning tasks for training. The output knowledge file consists of a GNN classifier that predicts whether a grounded action is useful for solving a given planning task.

First, we obtain training data by solving the planning tasks provided as input. For that, we use Downward Lab (Seipp et al. 2017) to run the LAMA (Richter and Westphal 2010) planner to obtain a plan for each task, as well as a symbolic search planner (Torralba et al. 2017) that computes the set of actions in all optimal plans. With this, we can create a labelled dataset consisting of the PDG of all the training instances and labelling nodes corresponding to "useful" actions.

For training and running GNNs, we use the Pytorch-geometric library (Fey and Lenssen 2019), using message-

passing GNNs (Bronstein et al. 2021). Our training method has several hyperparameters that can be tuned. First of all, the architecture of the neural network, such as number of layers, neurons per layer, and learning rate. Second, the definition of what actions are labelled as "useful". Our default configuration considers an action useful iff it belongs to some optimal plan of the task. However, we also considered an alternative, where useful actions are those that are used in the plan found by LAMA.

The decision of what NN architecture has better performance, as well as well as what is the best criteria to decide whether an action is useful, may differ for different domains. Therefore, we set the value of these parameters by an hyperparameter optimization tool, SMAC (Lindauer et al. 2022). SMAC is a versatile tool, allowing to automatically search for parameter configurations that maximize/minimize any optimization criteria. One could optimize for training loss, or classification performance. However, that ignores the impact that the reduction has on the planner's performance. Instead, we optimize for reducing the number of actions that are kept after our preprocessing, giving a very high penalty if the task is not solved.

The learning phase concludes with the generation of the knowledge file, which contains the best classifier found by the SMAC optimization.

Planning Phase

In the planning phase, HUZAR leverages the knowledge file in order to simplify the planning task, by removing actions that are classified as not useful. HUZAR's planning engine is implemented on top of Scorpion (Seipp 2018), a variant of the Fast Downward Planning System (Helmert 2006) that implements, which had already integrated the h2-preprocessor (Alcázar and Torralba 2015).

Then HUZAR's preprocessor removes actions deemed not useful. First, it grounds the planning task into a SAS^+ task (Helmert 2009), and generates the corresponding PDG. Then, we run the GNN classifier to obtain a value between 0 and 1 for each action. We choose a threshold T and discard all actions with $GNN(a) < T$. The preprocessor can also be configured to choose T such that at least X% of the actions are not removed. We start with greedier variants that may remove up to 90% of the actions. If, after removing the actions the planner fails to find a plan (e.g. because no plan is possible with the remaining actions), then we repeat the process, increasing the percentage of actions that should be kept.

After applying the filter of the GNN, we run the h^2 preprocessor (Alcázar and Torralba 2015), which further removes actions from the planning task. Specifically, it removes actions that are only applicable on unreachable states and/or result in dead-end states. This also removes unreachable facts and irrelevant variables so that the task only contains variables that can still be modified by the remaining actions.

After the task has been grounded and simplified, we solve it using LAMA (Richter and Westphal 2010).

Conclusion

In conclusion, HUZAR is a new planner that simplifies the planning task by learning to remove unnecessary actions. We show how, by encoding the task as a PDG, one can leverage Graph Neural Networks to classify actions as useful or unnecessary.

References

Alcázar, V.; and Torralba, Á. 2015. A Reminder about the Importance of Computing and Exploiting Invariants in Planning. In Brafman, R.; Domshlak, C.; Haslum, P.; and Zilberstein, S., eds., *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling (ICAPS 2015)*, 2–6. AAAI Press.

Bäckström, C.; and Nebel, B. 1995. Complexity Results for SAS^+ Planning. *Computational Intelligence*, 11(4): 625–655.

Bronstein, M. M.; Bruna, J.; Cohen, T.; and Velickovic, P. 2021. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. *CoRR*, abs/2104.13478.

Fey, M.; and Lenssen, J. E. 2019. Fast Graph Representation Learning with PyTorch Geometric. *CoRR*, abs/1903.02428.

Fišer, D.; Torralba, Á.; and Shleyfman, A. 2019. Operator Mutexes and Symmetries for Simplifying Planning Tasks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI 2019)*, 7586–7593. AAAI Press.

Fout, A.; Byrd, J.; Shariat, B.; and Ben-Hur, A. 2017. Protein Interface Prediction using Graph Convolutional Networks. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Gnad, D.; Torralba, Á.; Domínguez, M. A.; Areces, C.; and Bustos, F. 2019. Learning How to Ground a Plan – Partial Grounding in Classical Planning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI 2019)*, 7602–7609. AAAI Press.

Haslum, P. 2007. Reducing Accidental Complexity in Planning Problems. In Veloso, M. M., ed., *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, 1898–1903.

Haslum, P.; Helmert, M.; and Jonssson, A. 2013. Safe, Strong and Tractable Relevance Analysis for Planning. In Borrajo, D.; Kambhampati, S.; Oddi, A.; and Fratini, S., eds., *Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling (ICAPS 2013)*, 317–321. AAAI Press.

Helmert, M. 2006. The Fast Downward Planning System. *Journal of Artificial Intelligence Research*, 26: 191–246.

Helmert, M. 2009. Concise Finite-Domain Representations for PDDL Planning Tasks. *Artificial Intelligence*, 173: 503–535.

Heusner, M.; Wehrle, M.; Pommerening, F.; and Helmert, M. 2014. Under-Approximation Refinement for Classical Planning. In Chien, S.; Fern, A.; Ruml, W.; and Do, M., eds., *Proceedings of the Twenty-Fourth International Conference*

on Automated Planning and Scheduling (ICAPS 2014), 365–369. AAAI Press.

Lindauer, M.; Eggensperger, K.; Feurer, M.; Biedenkapp, A.; Deng, D.; Benjamins, C.; Ruhkopf, T.; Sass, R.; and Hutter, F. 2022. SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization. *Journal of Machine Learning Research*, 23(54): 1–9.

Nebel, B.; Dimopoulos, Y.; and Koehler, J. 1997. Ignoring Irrelevant Facts and Operators in Plan Generation. In Steel, S.; and Alami, R., eds., *Recent Advances in AI Planning. 4th European Conference on Planning (ECP 1997)*, volume 1348 of *Lecture Notes in Artificial Intelligence*, 338–350. Springer-Verlag.

Pochter, N.; Zohar, A.; and Rosenschein, J. S. 2011. Exploiting Problem Symmetries in State-Based Planners. In Burgard, W.; and Roth, D., eds., *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2011)*, 1004–1009. AAAI Press.

Richter, S.; and Westphal, M. 2010. The LAMA Planner: Guiding Cost-Based Anytime Planning with Landmarks. *Journal of Artificial Intelligence Research*, 39: 127–177.

Sanchez-Gonzalez, A.; Heess, N.; Springenberg, J. T.; Merel, J.; Riedmiller, M.; Hadsell, R.; and Battaglia, P. 2018. Graph Networks as Learnable Physics Engines for Inference and Control. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 4470–4479.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1): 61–80.

Seipp, J. 2018. Fast Downward Scorpion. In *Ninth International Planning Competition (IPC-9): Planner Abstracts*, 77–79.

Seipp, J.; Pommerening, F.; Sievers, S.; and Helmert, M. 2017. Downward Lab. https://doi.org/10.5281/zenodo.790461.

Shleyfman, A.; Katz, M.; Helmert, M.; Sievers, S.; and Wehrle, M. 2015. Heuristics and Symmetries in Classical Planning. In Bonet, B.; and Koenig, S., eds., *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2015)*, 3371–3377. AAAI Press.

Torralba, Á.; Alcázar, V.; Kissmann, P.; and Edelkamp, S. 2017. Efficient Symbolic Search for Cost-optimal Planning. *Artificial Intelligence*, 242: 52–79.

Torralba, Á.; and Kissmann, P. 2015. Focusing on What Really Matters: Irrelevance Pruning in Merge-and-Shrink. In Lelis, L.; and Stern, R., eds., *Proceedings of the Eighth Annual Symposium on Combinatorial Search (SoCS 2015)*, 122–130. AAAI Press.

Wu, Y.; Lian, D.; Xu, Y.; Wu, L.; and Chen, E. 2020. Graph Convolutional Networks with Markov Random Field Reasoning for Social Spammer Detection. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 1054–1061. AAAI Press.