Towards a Coalition Focused Neural-Symbolic Generative Policy Model

Daniel Cunnington
IBM Research
Winchester, UK
dancunnington@uk.ibm.com

Alessandra Russo
Imperial College
London, UK
a.russo@imperial.ac.uk

Elisa Bertino
Purdue University
West Lafayette, IN, USA
bertino@purdue.edu

Seraphin Calo
IBM Research
Yorktown Heights, NY, USA
scalo@us.ibm.com

Abstract

To facilitate information sharing between systems and devices in a coalition operation, unstructured data from various sensors must be analysed accordingly. Recent work has developed the notion of a context-dependant generative policy framework capable of learning generative policy models from strings and text-based data in a tabular format. However, in order to fully utilise generative policy models in coalition environments, it is vital that unstructured contextual information can be analysed alongside tabular data, potentially at the edge of the network. This paper performs a deep-dive into the emerging field of neural-symbolic machine learning with a view towards enabling future neural-symbolic generative policy models that are capable of analysing both structured and unstructured data present in a coalition operation, whilst providing full transparency and enabling edge of network reasoning capability. Specifically, a technique called DeepProbLog is investigated and applied to a coalition scenario, highlighting key research challenges and questions to be addressed in order to advance the capability of generative policy models for coalition operations.

1 Introduction

Generative policy models [1] have been proposed as a means of autonomously managing the interactions and behaviours between systems and devices within coalition operations (i.e. multi-agent environments) across varying contexts. Recent work has developed a formal framework for learning context-dependant generative policy models using Answer Set Grammars (ASGs) [2, 3] with applications in autonomous vehicles [4], logistical resupply [5], access control [6] and information sharing [7]. In all of the applications explored to-date, generative policy models have been learned from string and tabular based training examples. However, to fully utilise generative policy models in coalition environments it is crucial to also analyse and learn from unstructured data to gain a better understanding of the context in which devices operate. A hybrid neural-symbolic learning approach for learning generative policy models could match the required explainable properties of symbolic learning with the powerful performance of neural models for analysing unstructured data.

This paper performs a deep-dive into the field of neural-symbolic learning by exploring the effectiveness of an emerging technique, called *DeepProbLog* [8], for combining neural learning of contextual features from unstructured data together with symbolic-based inference of coalition policy decisions. We compare its performance to a pure end-end statistical learning approach using a logistical resupply

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

scenario, and show that the hybrid approach reaches a higher accuracy when inferring resulting policy decisions. Based on these results we argue that a future hybrid neural-symbolic learner capable of inferring, and learning generative policy models from structured and unstructured data can be deployed in real-world coalition environments due to their ability to learn over multi-modal datasets whilst retaining the required level of explainability.

In section 2 we summarise some of the key state-of-the-art work in the field of neural-symbolic learning, in section 3 we outline our experimental approach and in section 4 we present our results and discuss future work. Finally, we conclude in section 5.

2 Related Work

The field of neural-symbolic learning has made significant progress in the last decade. Among the initial contributions are the work of Garcez et al. [9] and Hammer and Hitzler [10], summarised, together with related approaches, in Besold et al. [11]. Among these, *CILP* [12] uses a neural network to approximate logical rules such that inductive learning can be used to refine logic-based background knowledge with examples. The emphasis is on using neural computation to improve symbolic knowledge. More recently, Hu et al., [13] distill logical rules into the weights of a neural network to assist the neural network in learning challenging aspects of a classification task from unstructured text, such as contrastive sentences in sentiment analysis. The emphasis here is to use symbolic knowledge to improve a neural learning task. Manhaeve et al. [8] introduce a probabilistic logic programming language, called *DeepProbLog*, that integrates neural networks with probabilistic logic programming modelling and reasoning. This extends *ProbLog* [14] to support the learning of neural objects from unstructured data such as images by guiding the neural classification process through probabilistic logic-based inference. This integrated neural-symbolic model, with given fixed probabilistic logic-based rules is trained end-end from examples of labelled consequences that the logic program is modelled to infer.

In this paper, we apply the DeepProbLog approach to a more complex unstructured dataset and combine the training and classification of two neural models together with a probabilistic logic-based program that encodes a given set of policies about decisions to make in response to the classification, using a Convoloutional Neural Network (CNN) model, of particular objects in CCTV camera images. The motivation behind this is twofold: (i) evaluate the effectiveness of ProbLog when dealing with raw, multi-modal data and (ii) provide a first step of a wider program of research aimed at learning generative policy models in an end-to-end fashion from multi-modal contextual datasets including imagery, video and audio.

3 Experimental Approach

Firstly, we describe our application scenario. Consider a logistical resupply mission where a resupply convoy must reach a particular destination, navigating through areas of adversarial activity. Let us assume the resupply convoy does not have communication to high performance back-end computing facilities although does have access to local sensing infrastructure within the coalition environment. In our experiment, we assume the resupply convoy has access to a CCTV camera image, as shown in Figure 1a, that is capable of observing enemy vehicles. We also assume that the resupply convoy has access to a team of ground troops that may be available or unavailable at a given time. The convoy must autonomously decide various actions to take within the mission, in order to mitigate the risk of adversarial compromise. In particular, it operates using the following policies: *IF enemy_vehicle_present AND ground_troops_available THEN deploy_ground_troops*. *IF* (*enemy_vehicle_present AND ground_troops_unavailable*) *OR unsure THEN re_route_convoy*. *IF enemy_vehicle_not_present THEN proceed_as_normal*.

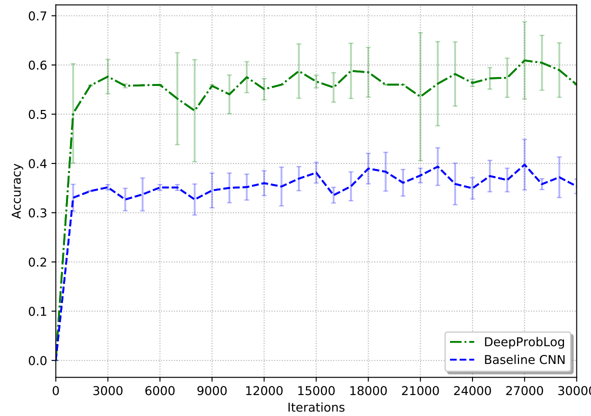
We utilise the Transport for London (TfL) CCTV dataset[1] and consider red buses to represent enemy vehicles. We use the MNIST[2] dataset to simulate the prediction of ground troop availability, assuming for our experiment that the MNIST digit 0 represents *ground_troops_available* and the MNIST digit 1 represents *ground_troops_not_available*. We formalise the policies using ProbLog and generate training and test examples of the form $mission\_step(cctv\_image, mnist\_image, policy\_action)$.

[1] `https://data.london.gov.uk/dataset/tfl-live-traffic-cameras`

[2] `http://yann.lecun.com/exdb/mnist/`

The full listing for the DeepProbLog program that represents the logistical resupply scenario is given in Appendix A.

(a) Sample CCTV image



(b) Experimental results

Figure 1: Sample CCTV image indicating the presence of an enemy vehicle represented as a red London bus and the experimental results of comparing the accuracy of DeepProbLog to a baseline CNN for predicting policy outcomes on the test set.

To analyse the CCTV contextual data, we utilise a CNN pre-trained on the ResNet18 architecture[3] and adopt a transfer learning approach, freezing the weights of all layers except the final fully connected layer which remains trainable. The CNN outputs 3 classes, *enemy_vehicle_present*, *enemy_vehicle_not_present*, and *unsure* – which occurs if TfL redact a particular CCTV image. This is analagous to a coalition environment as a coalition partner may wish to dynamically restrict access to a given sensor. As discussed, to simulate the decision as to whether or not ground troops are available, which in reality may be due to a complex set of factors such as the severity of enemy activity, weather conditions and resource availability, we use a binary classifier in the form of a CNN to detect MNIST[4] digits 0 or 1 and adopt the same CNN architecture that is used in the DeepProbLog [8] MNIST experiments. We implement our experiment in PyTorch, extending the DeepProbLog framework[5].

We train our DeepProbLog model end-end with respect to the final policy decision, represented as a potential action within the logistical resupply mission—i.e. *deploy_ground_troops*, *re_route_convoy* or *proceed_as_normal*. The DeepProbLog model learns from example policy decisions and updates the weights of the CCTV and Ground Troop CNNs through back-propagation. Note that the CCTV and Ground Troop CNNs are not trained independently, they learn to classify enemy vehicle status and ground troop availability respectively as part of the integrated learning and policy inferencing process DeepProbLog provides.

4 Results

Figure 1b details the results of our experiment, comparing the accuracy of the fully integrated DeepProbLog approach with a baseline neural approach over $30,000$ training iterations. We use the same dataset for both approaches which consists of $10,000$ train and 500 test examples, where an example consists of a CCTV image and an MNIST image annotated with a resulting policy decision. Also, we repeat our experiment 5 times and take the average accuracy at each number of iterations to account for random weight initialisation in the neural components. Error bars indicating the standard deviation at each number of iterations are shown in Figure 1b. The goal of the DeepProbLog approach is to train the neural CNN components whilst inferring the `mission_step/3` predicate as listed in Appendix A. For the baseline, we adopt a similar approach to the DeepProbLog experiments [8]

[3] `https://www.kaggle.com/pytorch/resnet18`

[4] `http://yann.lecun.com/exdb/mnist/`

[5] `https://bitbucket.org/problog/deepproblog/src`

and concatenate the CCTV images with MNIST images to train a CNN to classify the resulting *mission_step* policy action from the concatenation of the two images.

With the addition of the logical rules, the DeepProbLog approach significantly outperforms the baseline after $30,000$ iterations. Nevertheless, the accuracy of this policy learning task is far below acceptable levels for real-world application. This is due to the challenging nature of identifying red buses in TfLs CCTV cameras. We substantiated this by running a second experiment in which the CCTV CNN was replaced with a 3-digit MNIST CNN. After only $1,000$ iterations the trained model achieved near 100% accuracy. Future work should be performed to improve the CCTV CNN and investigate object detection models using Faster-RCNN [15].

Analysing the results in Figure 1b, it is apparent that the DeepProbLog approach exhibits more variance in it's performance when compared to the baseline CNN, especially at $21,000$ iterations. We hypothesise that this is due to the different random weight initialisations in the neural components in comparison to the baseline. In the DeepProbLog approach both the MNIST CNN *and* the CCTV CNN are subject to random weight initialisation, whereas in the baseline approach there is only one random weight initialisation. This highlights a potential disadvantage for the DeepProbLog approach in coalition operations, where consistency is important as it may not be possible in terms of time to conduct many repeated learning tasks. Also, given the vast number of potential contextual features, the number of different neural components could be large. Future work should investigate increasing the number of neural components to observe the effect on performance consistency. Also, if further training time or additional computational resources are available, the number of experimental repeats could be increased to further evaluate this hypothesis.

5 Discussion and Conclusion

We have explored the use of a neural-symbolic learning technique—*DeepProbLog* for the task of inferring policy decisions from multi-modal data in the context of a logistical resupply operation. Compared to a baseline CNN, DeepProbLog results in significantly stronger performance when inferring policy decisions although exhibits greater variance between experimental repeats.

Another key limitation of the current technique is that the logical rules are specified by humans. Our future work will investigate creating a neural-symbolic generative policy model that learns the weights of neural components aimed at predicting contextual features from raw, unstructured data whilst also learning symbolic policy rules by means of our ASG-based policy learner [2, 3]. We plan to first consider the case where the weights of neural components trained to extract contextual features remain fixed and their output is used by a symbolic policy learner to learn a context-dependant generative policy model. We will then investigate a full *end-end* neural-symbolic generative policy model architecture, which back-propagates outcomes of learned rules through the neural components to help improve the contextual feature classification and consequently the learned rules. We will evaluate the two approaches with respect to accuracy, training time, explainability and consistency so to satisfy the requirements for operability in coalition environments.

Acknowledgments

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] D. Verma, S. Calo, S. Chakraborty, E. Bertino, C. Williams, J. Tucker, and B. Rivera, "Generative policy model for autonomic management," in *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Aug 2017, pp. 1–6.

[2] M. Law, A. Russo, B. Elisa, B. Krysia, and L. Jorge, “Representing and learning grammars in answer set programming.” in *AAAI*, 2019.

[3] S. Calo, I. Manotas, G. de Mel, D. Cunnington, M. Law, D. Verma, A. Russo, and E. Bertino, “AGENP: An ASGrammar-based GENerative policy framework,” in *Policy-Based Autonomic Data Governance*. Springer, Sep. 2019, pp. 3–20. [Online]. Available: https://doi.org/10.1007/978-3-030-17277-0_1

[4] D. Cunnington, I. Manotas, M. Law, G. de Mel, S. Calo, E. Bertino, and A. Russo, “A generative policy model for connected and autonomous vehicles,” *2019 IEEE Intelligent Transportation Systems Conference*, Oct. 2019, due to appear.

[5] G. White, J. Ingham, M. Law, and A. Russo, “Using an asg based generative policy to model human rules,” *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, Jun. 2019.

[6] S. Calo, D. Verma, S. Chakraborty, E. Bertino, E. Lupu, and G. Cirincione, “Self-generation of access control policies,” in *Proceedings of the 23Nd ACM on Symposium on Access Control Models and Technologies*, ser. SACMAT ’18. New York, NY, USA: ACM, 2018, pp. 39–47. [Online]. Available: http://doi.acm.org/10.1145/3205977.3205995

[7] D. Cunnington, G. White, M. Law, and G. de Mel, “A demonstration of generative policy models in coalition environments,” in *Advances in Practical Applications of Survivable Agents and Multi-Agent Systems: The PAAMS Collection*. Springer International Publishing, 2019, pp. 242–245.

[8] R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, and L. De Raedt, “Deepproblog: Neural probabilistic logic programming,” in *Advances in Neural Information Processing Systems*, 2018, pp. 3749–3759.

[9] A. S. Garcez, L. C. Lamb, and D. M. Gabbay, *Neural-symbolic cognitive reasoning*. Springer Science & Business Media, 2008.

[10] B. Hammer and P. Hitzler, *Perspectives of neural-symbolic integration*. Springer, 2007, vol. 77.

[11] T. R. Besold, A. d. Garcez, S. Bader, H. Bowman, P. Domingos, P. Hitzler, K.-U. Kühnberger, L. C. Lamb, D. Lowd, P. M. V. Lima *et al.*, “Neural-symbolic learning and reasoning: A survey and interpretation,” *arXiv preprint arXiv:1711.03902*, 2017.

[12] A. S. Avila Garcez and G. Zaverucha, “The connectionist inductive learning and logic programming system,” *Applied Intelligence*, vol. 11, no. 1, pp. 59–77, Jul 1999. [Online]. Available: https://doi.org/10.1023/A:1008328630915

[13] Z. Hu, X. Ma, Z. Liu, E. Hovy, and E. Xing, “Harnessing deep neural networks with logic rules,” *arXiv preprint arXiv:1603.06318*, 2016.

[14] L. De Raedt, A. Kimmig, and H. Toivonen, “Problog: A probabilistic prolog and its application in link discovery.” in *IJCAI*, vol. 7. Hyderabad, 2007, pp. 2462–2467.

[15] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in neural information processing systems*, 2015, pp. 91–99.

A DeepProbLog Logistical Resupply Program

```
nn(red_bus_net,[X],Y, [present,not_present,unsure])
     :: neural_red_bus(X,Y).
nn(ground_troops_net,[X],Y, [available,unavailable])
     :: neural_ground_troops(X,Y).

outcome(present,available,deploy_ground_troops).
outcome(present,unavailable,re_route_convoy).
outcome(not_present,_,proceed_as_normal).
outcome(unsure,_,re_route_convoy).

mission_step(CCTV,GT_Info,Result) :-
          red_bus(CCTV,RB_Status),
          ground_troops(GT_Info,GT_Status),
          outcome(RB_Status, GT_Status, Result).

red_bus(CCTV,present) :-
     neural_red_bus(CCTV,present).
red_bus(CCTV,not_present) :-
     neural_red_bus(CCTV,not_present).
red_bus(CCTV,unsure) :- neural_red_bus(CCTV,unsure).
ground_troops(GT_Info,available) :-
     neural_ground_troops(GT_Info,available).
ground_troops(GT_Info,unavailable) :-
     neural_ground_troops(GT_Info,unavailable).
```