Interpretable Robust Decision Making

Extended Abstract

Hal James Cooper
Columbia University
New York, NY
hal.cooper@columbia.edu

Garud Iyengar
Columbia University
New York, NY
garud@ieor.columbia.edu

Ching-Yung Lin
Graphen, Inc.
New York, NY
cylin@graphen.ai

ABSTRACT

Interpretable decision making frameworks allow us to easily endow agents with specific goals, risk tolerances, and understanding. Existing decision making systems either forgo interpretability, or pay for it with severely reduced efficiency and large memory requirements. In this paper, we outline DeepID, a neural network approximation of Influence Diagrams, that avoids both pitfalls.

KEYWORDS

Interpretable agent modelling; deep learning; robustness

ACM Reference Format:
Hal James Cooper, Garud Iyengar, and Ching-Yung Lin. 2018. Interpretable Robust Decision Making. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

In most decision making frameworks, there is a trade-off between interpretability, i.e. the ability to clearly identify the relationships between the key elements, and computational efficiency. There are many decision making settings, in particular, those involving financial or regulatory decisions, where interpretability often takes precedence over efficiency.

Influence Diagrams (IDs) [4] were one of the earliest quantitative approaches for decision making with a single agent. An ID consists of sets of chance nodes C, decision nodes D, and utility nodes U, connected by directed arcs representing the conditional independence relations. The ID reduces to a Bayesian network once a strategy is chosen at each of the decision nodes, thus inheriting the conditional dependence structure of the Bayesian network. This allows for the relationships between key elements impacting the decision problem to be defined in a clearly interpretable manner. Although IDs facilitate interpretable decision analysis, IDs are not able to efficiently represent and integrate over distributions associated with the chance and decision nodes.

2 DEEPID: DEEP INFLUENCE DIAGRAMS

In the DeepID framework, each chance node, decision and utility node of an ID is replaced by a type of Differentiable Generator Network (DGN), i.e. a function $g_{\bar{\theta}_c}(\pi(c), \boldsymbol{\epsilon})$, where $\pi(c)$ denotes the outputs at the direct parents of the node c in the ID representation, the fixed parameter $\bar{\theta}_c$ is learned by matching conditional distributions, and $\boldsymbol{\epsilon}$ are samples from a given fixed distribution. In contrast, at a decision d, the strategy is represented by the DGN $g_{\theta_d}(\pi(d), \boldsymbol{\epsilon})$ where the parameter θ_d is chosen to maximize the average utility.

The DeepID is thus formed by connecting (according to a topological sorting of the nodes of the ID) the DGNs that correspond to each ID node to form a neural network. In the resulting network, the chance and utility nodes can be trained apriori as GANs [2, 9] since they approximate conditional distributions; whereas the decision nodes are collectively trained as feedforward neural networks to optimize the sample mean using gradient descent methods.

Some distributions (such as those in the location-scale family) can be easily implemented using DGNs. For example, suppose that the distribution $f(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu} + \mathbf{C}\mathbf{y}, \mathbf{R}\mathbf{R}^\top)$ is a multivariate Gaussian distribution with mean $\boldsymbol{\mu} + \mathbf{C}\mathbf{y}$ and covariance $\mathbf{R}\mathbf{R}^\top$. This distribution can be generated by the DGN $g_\theta(\mathbf{y}, \boldsymbol{\epsilon}) = \boldsymbol{\mu} + \mathbf{C}\mathbf{y} + \mathbf{R}\boldsymbol{\epsilon}$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, which is a simple linear layer with an offset. We can use this "reparameterization trick" [6] to create simple but exact chance (where parameters are trained or set apriori) or decision (where parameters are free to be trained during joint learning of the full DeepID DGN) nodes. The Concrete distribution [5, 7] can be similarly employed to facilitate gradient based training of discrete nodes.

Clearly, the DeepID framework is of interest only if one can guarantee that the optimal strategy can be represented by the associated network of DGNs. It is easy to show that a large class of IDs can be arbitrarily closely approximated by DeepIDs, with optimal solutions that correspond to one another. Consider an ID with chance nodes C, decision nodes D and utility nodes U. Suppose the outputs of all nodes $i \in C \cup D$ take values in a compact set, the inverse conditional CDF $F_i^{-1}(x_i|\pi(i)\}$ is continuous for all $i \in C \cup D$, and the utility functions are differentiable and bounded. Then we have the following:

(a) Let $\bar{\sigma} = \{\bar{\sigma}_d\}_{d \in D}$ denote any strategy profile across all decision nodes D in the ID, and let $\mathbb{P}_{\bar{\sigma}}$ denote the corresponding joint distribution over actions, chance and utility node outcomes. Then there exists a sequence of DGNs $g^{(n)}$ and parameter vectors $\theta^{(n)}$ such that the corresponding joint distribution over actions, chance, and utility outcomes $\mathbb{P}_{g^n_{\theta^{(n)}}} \xrightarrow{D} \mathbb{P}_{\bar{\sigma}}$.

(b) Let σ^* denote the optimal strategy for the ID, with expected utility $\mathbb{E}[u(\sigma^*)]$. Then there exists a sequence of DGNs g^n with input size m such that $\mathbb{E}[u(g^n_{\theta^{(n)}_{\max}}(X))] \to \mathbb{E}[u(\sigma^*)]$, for $X \sim \text{Uniform}[0, 1]^m$, where the components of the parameter vector $\theta^{\max}$ corresponding to the chance and utility nodes are defined by matching conditional distributions apriori, and the parameters corresponding the decision nodes are computed by the maximization of the expected utility.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden. © 2018 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

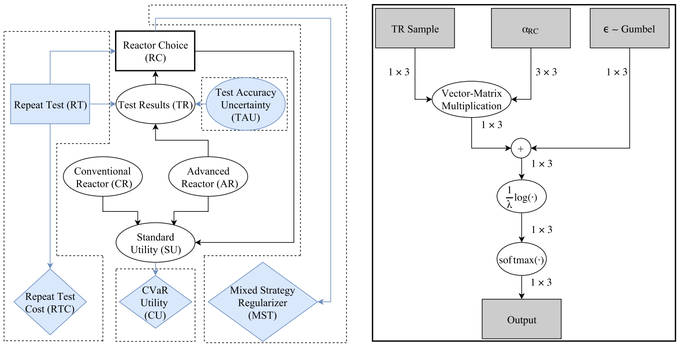

Figure 1: Reactor problem DeepID and modifications (LHS) and internal DGN of RC (RHS).

3 THE ROBUST REACTOR

To demonstrate the concepts of this paper, we introduce (see Figure 1) a variation of the Reactor problem [11] that involves an agent conducting a test of an advanced reactor design, and then choosing between a conventional or the advanced reactor depending upon the test results. The performance of both the conventional and advanced reactors are uncertain; however, the performance of the advanced reactor is less certain (but with higher payoff if successful). The LHS of Figure 1 shows the macro-level of the DeepID, which corresponds exactly to the original standard ID of the problem (the shaded and boxed sub-graphs will be addressed in Section 4). On the RHS of Figure 1, we show the internal DGN representation of the Reactor Choice node (where α_{RC} are the parameters to be learned and λ is the temperature of the Concrete approximation).

4 INTERPRETABLE ROBUSTNESS

Decision problems typically involve a complex set of interacting elements, some more uncertain than others. Thus, it is beneficial for decision making frameworks to be flexible enough to allow for targeted introduction of robustness. In this section, we show DeepIDs allow for targeted robustness to be introduced in an interpretable manner. In a DeepID, the "purpose" of each component DGN (to approximate the corresponding ID node) is retained; therefore, one can separately control the robustness of each component network. Introducing such targeted robustness is nearly impossible in a standard deep network where we have relatively limited understanding of the function of particular nodes or subnetworks. MDPs [1, 3] also do not easily accommodate targeted robustness since the definition of state, and corresponding transitions, often obscure the underlying independence relations [8, 10]. We use the DeepID Reactor problem to demonstrate some of the interpretably robust adjustments that can be made to DeepIDs. Note that all these modifications to a DeepID are equivalent to adding new chance, decision or utility nodes, changing the objective of the training algorithm. Consequently, such modifications do not increase the complexity class of the problem.

Distributional uncertainty for specific chance and decision nodes: For a chance node $c \in C$ with the DGN $g_{\bar{\theta}_c}(\pi(c), \epsilon)$ we can modify the parameters $\bar{\theta}_c$ or the exogenous samples ϵ to model distributional uncertainty. For example, we can introduce uncertainty to the accuracy of the advanced reactor test result (TR node) by replacing constants in the TR distribution parameters with hyperparameters of some distribution. This type of change can be represented graphically by adding a new chance node c', adding a directed arc (c', c), and updating $g_{\bar{\theta}_c}(\pi(c), \epsilon)$ accordingly. Similarly, for an agents decision node d with DGN $g_{\theta_d}(\pi(d), \epsilon)$, we can add noise at any level – to θ_d, to ϵ, or the output of d – to encourage gradient descent to compute a stable decision strategy. We can interpret this as encoding that decision execution isn't exact, with the agent sometimes making mistakes.

Regularization for specific decisions: Consider an agents decision node d with the DGN $g_{\theta_d}(\pi(d), \epsilon)$, we can regularize the parameters θ_d to encourage certain properties at node d. A particularly noteworthy application of decision level regularizers is to encourage particular discrete decisions to have pure or mixed strategies. This can be achieved by adding a utility node u (MST in the Reactor problem) with $g_{\bar{\theta}_u}(g_{\theta_d}(\pi(d), \epsilon), \epsilon')$ a p-norm regularization penalty on θ_d (e.g. to encourage a pure strategy for choosing a reactor).

Custom risk-tolerances: We can modify the network $g_{\theta_u}(\pi(u), \epsilon)$ at a utility node u by adding a new layer that represents an agents' risk-reward tolerance, i.e. setting the output to $g_{\theta_{u'}}(g_{\theta_u}(\pi(u), \epsilon), \epsilon')$. This is represented by introducing a new node u' and adding an arc (u, u'). Conditional Value at Risk (CVaR) is an example of a common-risk measure used in financial decision problems; we can easily incorporate CVaR into the Reactor problem by feeding Standard Utility samples into a new utility node CU and calculating the sample-wise CVaR.

5 CONCLUSION AND FUTURE WORK

In this work, we sketched the method of the DeepID approach. We showed how it maintains the macro-level interpretability of the ID and how this facilitiates robust modifications. We argued that by approximating (arbitrarily well) each node with a DGN, we can then employ the well-developed tools of deep-learning and gradient based methods to facilitate scalability. We did not present formal guarantees of convergence or convergence rates, and so developing such formal guarantees is an important area of future research. We similarly seek to demonstrate the performance of the DeepID at very large scales.

REFERENCES

[1] Richard Bellman. 1957. *A Markovian decision process.* Technical Report. DTIC Document.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[3] Ronald A Howard. 1960. *Dynamic Programming and Markov Processes.* MIT Press.

[4] Ronald A Howard. 1983. *Readings on the principles and applications of decision analysis.* Vol. 1. Strategic Decisions Group.

[5] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical Reparameterization with Gumbel-Softmax. *arXiv preprint arXiv:1611.01144* (2016).

[6] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[7] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. 2016. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. *arXiv preprint arXiv:1611.00712* (2016).

[8] Paolo Magni, Silvana Quaglini, Monia Marchetti, and Giovanni Barosi. 2000. Deciding when to intervene: a Markov decision process approach. *International Journal of Medical Informatics* 60, 3 (2000), 237–253.

[9] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[10] Joseph A Tatman and Ross D Shachter. 1990. Dynamic programming and influence diagrams. *IEEE transactions on systems, man, and cybernetics* 20, 2 (1990), 365–379.

[11] Miguel A Virto, Jacinto Martín, David Ríos Insua, and Arminda Moreno-Díaz. 2002. Approximate Solutions of Complex Influence Diagrams through MCMC Methods. In *Probabilistic Graphical Models*.