The Journal of Open Source Software

pymdp: A Python library for active inference in discrete state spaces

Conor Heins[1,2,3,4¶], **Beren Millidge**[4,5], **Daphne Demekas**[6], **Brennan Klein**[4,7,8], **Karl Friston**[9], **Iain D. Couzin**[1,2,3], and **Alexander Tschantz**[4,10,11¶]

1 Department of Collective Behaviour, Max Planck Institute of Animal Behavior, 78457 Konstanz, Germany **2** Centre for the Advanced Study of Collective Behaviour, 78457 Konstanz, Germany **3** Department of Biology, University of Konstanz, 78457 Konstanz, Germany **4** VERSES Research Lab, Los Angeles, California, USA **5** MRC Brain Networks Dynamics Unit, University of Oxford, Oxford, UK **6** Department of Computing, Imperial College London, London, UK **7** Network Science Institute, Northeastern University, Boston, MA, USA **8** Laboratory for the Modeling of Biological and Socio-Technical Systems, Northeastern University, Boston, USA **9** Wellcome Centre for Human Neuroimaging, Queen Square Institute of Neurology, University College London, London WC1N 3AR, UK **10** Sussex AI Group, Department of Informatics, University of Sussex, Brighton, UK **11** Sackler Centre for Consciousness Science, University of Sussex, Brighton, UK ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Elizabeth DuPre

Reviewers:

- @seankmartin
- @patrickmineault

Submitted: 14 January 2022
Published: unpublished

Statement of Need

Active inference is an account of cognition and behavior in complex systems which brings together action, perception, and learning under the theoretical mantle of Bayesian inference (Friston et al., 2009, 2012, 2015, 2017). Active inference has seen growing applications in academic research, especially in fields that seek to model human or animal behavior (Adams et al., 2021; Holmes et al., 2021; Parr et al., 2020). The majority of applications have focused on cognitive neuroscience, with a particular focus on modelling decision-making under uncertainty (Schwartenbeck et al., 2015; Smith et al., 2020, 2021). Nonetheless, the framework has broad applicability and has recently been applied to diverse disciplines, ranging from computational models of psychopathology (Montague et al., 2012; Smith et al., 2021), control theory (Baioumy et al., 2021; Baltieri & Buckley, 2019; Millidge et al., 2020) and reinforcement learning (Fountas et al., 2020; Millidge, 2020; Sajid et al., 2021; Tschantz, Baltieri, et al., 2020; Tschantz, Millidge, et al., 2020), through to social cognition (Adams et al., 2021; Tison & Poirier, 2021; Wirkuttis & Tani, 2021) and even real-world engineering problems (Fox, 2021; Martínez et al., 2021; Moreno, 2021). While in recent years, some of the code arising from the active inference literature has been written in open source languages like Python and Julia (Çatal et al., 2020; T. W. van de Laar & Vries, 2019; Millidge, 2020; Tschantz, Seth, et al., 2020; Ueltzhöffer, 2018), to-date, the most popular software for simulating active inference agents is the `DEM` toolbox of `SPM` (Friston et al., 2008; Smith et al., 2022), a MATLAB library originally developed for the statistical analysis and modelling of neuroimaging data (Penny et al., 2007). `DEM` contains a reliable, reproducible set of functions for studying active inference, but the use of the toolbox can be restrictive for researchers in settings where purchasing a MATLAB license is financially costly. And although active inference researchers have relied heavily on `DEM` for simulating and fitting models of behavior, most of its functionality is restricted to single MATLAB scripts or functions, particularly one called `spm_MDP_VB_X.m`, that lack modularity and often must be customized for applications on a domain-specific basis. Increasing interest in active inference, manifested both in terms of sheer number of cited research papers as well as diversifying applications across disciplines, has thus created a need for generic, widely-available, and user-friendly code for simulating active inference in open-source scientific computing languages like Python. The software we present here, `pymdp`, represents a significant step in this direction: namely, we provide the first open-source package for simulating active inference with discrete state-space generative models. The name pymdp derives from the fact that the package is written in the **Py**thon programming language and concerns discrete, Markovian generative models of decision-making, which take the form of Markov Decision Processes or **MDP**s.

pymdp is a Python package that is directly inspired by the active inference routines contained in DEM. However, pymdp is has a modular, flexible structure that allows researchers to build and simulate active inference agents quickly and with a high degree of customization. We developed pymdp in the hopes that it will increase the accessibility and exposure of the active inference framework to researchers, engineers, and developers with diverse disciplinary backgrounds. In the spirit of open-source software, we also hope that it spurs new innovation, development, and collaboration in the growing active inference and wider Bayesian modelling communities.

Summary

pymdp offers a suite of robust, tested, and modular routines for simulating active inference agents equipped with *partially-observable Markov Decision Process* (POMDP) generative models. Mathematically, a POMDP comprises a joint distribution over observations o, hidden states s, control states u and hyperparameters ϕ: $P(o, s, u, \phi)$. This joint distribution further factorizes into a set of categorical and Dirichlet distributions: the likelihoods and priors of the generative model. With pymdp, one can build a generative model using a set of prior and likelihood distributions, initialize an agent, and then link it to an external environment to run active inference processes - all in a few lines of code. The Agent and Env (environment) APIs of pymdp are built according to the standardized framework of OpenAIGym commonly used in reinforcement learning, where an agent and environment object recursively exchange observations and actions over time (Brockman et al., 2016).

Introduction

Simulations of active inference are commonly performed in discrete time and space (Da Costa et al., 2020; Friston et al., 2015). This is partially motivated by the mathematical tractability of performing inference with discrete probability distributions, but also by the intuition of modelling choice behavior as a sequence of discrete, mutually-exclusive choices, in e.g. psychophysics or decision-making experiments. The most popular generative models – used to realize active inference in this context – are partially-observable Markov Decision Processes or *POMDPs* (Kaelbling et al., 1998). POMDPs are state-space models that model the environment in terms of hidden states that stochastically change over time, as a function of both the current state of the environment as well as the behavioral output of an agent (control states or actions). Crucially, the environment is *partially-observable*, i.e. the hidden states are not directly observed by the agent, but can only be inferred through observations that relate to hidden states in a probabilistic manner, such that observations are modelled as being generated stochastically from the current hidden state. This necessitates both "perceptual" inference of hidden states as well as control.

As such, in most POMDP problems, an agent is tasked with inferring the hidden state of its environment and then choosing a sequence of control states or actions to change hidden states in a way that leads to desired outcomes (maximizing reward, or occupancy within some preferred set of states).

Usage

In order to enhance the user-friendliness of pymdp without sacrificing flexibility, we have built the library to be highly modular and customizable, such that agents in pymdp can be specified at a variety of levels of abstraction with desired parameterisations. The methods of the `Agent` class can thus be called in any particular order, depending on the application, and furthermore they can be specified with various keyword arguments that entail choices of implementation details at lower levels.

By retaining a modular structure throughout the package's dependency hierarchy, `pymdp` also affords the ability to flexibly compose different low level functions. This allows users to customize and integrate their active inference loops with desired inference algorithms and policy selection routines. For instance, one could sub-class the `Agent` class and write a customized `step()` function, that combines whichever components of active inference one is interested in.

Related software packages

The `DEM` toolbox within `SPM` in MATLAB is the current gold-standard in active inference modelling. In particular, simulating an active inference process in `DEM` consists of defining the generative model in terms of a fixed set of matrices and vectors, and then calling the `spm_MDP_VB_X.m` function to simulate a sequence of trials. `pymdp`, by contrast, provides a user-friendly and modular development experience, with core functionality split up into different libraries that separately perform the computations of active inference in a standalone fashion. Moreover, `pymdp` provides the user the ability to write an active inference process at different levels of abstraction depending on the user's level of expertise or skill with the package – ranging from the high level `Agent` functionality, which allows the user to define and simulate an active inference agent in just a few lines of code, all the way to specifying a particular variational inference algorithm (e.g. marginal-message passing (Parr et al., 2019)) for the agent to use during state estimation. In the `DEM` toolbox of `SPM`, this would require setting undocumented flags or else manually editing the routines in `spm_MDP_VB_X.m` to enable or disable bespoke functionality. There has been one recent attempt at creating a comprehensive user-guide for building active inference agents in `DEM` (Smith et al., 2022), though to our knowledge there has not been a package devoted to the open source development of these powerful software tools.

A recent related, but largely non-overlapping project is ForneyLab, which provides a set of Julia libraries for performing approximate Bayesian inference via message passing on Forney Factor Graphs (Cox et al., 2019). Notably, this package has also seen several applications in simulating active inference processes, using ForneyLab as the backend for the inference algorithms employed by an active inference agent (Ergul et al., 2020; T. van de Laar et al., 2021; T. W. van de Laar & Vries, 2019; Vanderbroeck et al., 2019). While ForneyLab focuses on including a rigorous set of message passing routines that can be used to simulate active inference agents, `pymdp` is specifically designed to help users quickly build agents (regardless of their underlying inference routines) and plug them into arbitrary environments to run active inference in a few easy steps.

Funding Statement

CH and IDC acknowledge support from the Office of Naval Research grant (ONR, N00014-64019-1-2556), with IDC further acknowledging support from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement (ID: 860949), the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy-EXC 2117- 422037984, and the Max Planck Society. KF is supported by funding for the Wellcome Centre for Human Neuroimaging (Ref: 205103/Z/16/Z) and the Canada-UK Artificial Intelligence Initiative (Ref: ES/T01279X/1). CH, DD, and BK acknowledge the support of a grant from the John Templeton Foundation (61780). The opinions expressed in this publication are those of the author(s) and do not necessarily reflect the views of the John Templeton Foundation.

Acknowledgements

The authors would like to thank Dimitrije Markovic, Arun Niranjan, Sivan Altinakar, Mahault Albarracin, Alex Kiefer, Magnus Koudahl, Ryan Smith, Casper Hesp, and Maxwell Ramstead for discussions and feedback that contributed to development of pymdp. We would also like to thank Thomas Parr for pointing out a technical error in an earlier version of the arXiv preprint for this work. Finally, we are grateful to the many users of pymdp whose feedback and usage of the package have contributed to its continued improvement and development.

References

Adams, R. A., Vincent, P., Benrimoh, D., Friston, K. J., & Parr, T. (2021). Everything is connected: Inference and attractors in delusions. *Schizophrenia Research*. https://doi.org/10.1016/j.schres.2021.07.032

Baioumy, M., Pezzato, C., Corbato, C. H., Hawes, N., & Ferrari, R. (2021). Towards stochastic fault-tolerant control using precision learning and active inference. *arXiv Preprint arXiv:2109.05870*. https://doi.org/10.1007/978-3-030-93736-2_48

Baltieri, M., & Buckley, C. L. (2019). PID control as a process of active inference with linear generative models. *Entropy*, *21*(3), 257. https://doi.org/10.3390/e21030257

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). Openai gym. *arXiv Preprint arXiv:1606.01540*.

Çatal, O., Verbelen, T., Nauta, J., De Boom, C., & Dhoedt, B. (2020). Learning perception and planning with deep active inference. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3952–3956. https://doi.org/10.1109/ICASSP40776.2020.9054364

Cox, M., Laar, T. van de, & Vries, B. de. (2019). A factor graph approach to automated design of Bayesian signal processing algorithms. *International Journal of Approximate Reasoning*, *104*, 185–204. https://doi.org/10.1016/j.ijar.2018.11.002

Da Costa, L., Parr, T., Sajid, N., Veselic, S., Neacsu, V., & Friston, K. J. (2020). Active inference on discrete state-spaces: A synthesis. *Journal of Mathematical Psychology*, *99*, 102447. https://doi.org/10.1016/j.jmp.2020.102447

Ergul, B., Laar, T. van de, Koudahl, M., Roa-Villescas, M., & Vries, B. de. (2020). Learning where to park. *International Workshop on Active Inference*, 125–132.

Fountas, Z., Sajid, N., Mediano, P. A. M., & Friston, K. J. (2020). Deep active inference agents using monte-carlo methods. *Advances in Neural Information Processing Systems*. https://proceedings.neurips.cc/paper/2020/hash/865dfbde8a344b44095495f3591f7407-Abstract.html

Fox, S. (2021). Active inference: Applicability to different types of social organization explained through reference to industrial engineering and quality management. *Entropy*, *23*(2), 198. https://doi.org/10.3390/e23020198

Friston, K. J., Daunizeau, J., & Kiebel, S. J. (2009). Reinforcement learning or active inference? *PLoS ONE*, *4*(7), e6421. https://doi.org/10.1371/journal.pone.0006421

Friston, K. J., FitzGerald, T., Rigoli, F., Schwartenbeck, P., & Pezzulo, G. (2017). Active inference: A process theory. *Neural Computation*, *29*(1), 1–49. https://doi.org/10.1162/NECO_a_00912

Friston, K. J., Rigoli, F., Ognibene, D., Mathys, C., Fitzgerald, T., & Pezzulo, G. (2015). Active inference and epistemic value. *Cognitive Neuroscience*, *6*(4), 187–214. https://doi.org/10.1080/17588928.2015.1020053

Friston, K. J., Samothrakis, S., & Montague, R. (2012). Active inference and agency: Optimal control without cost functions. *Biological Cybernetics*, *106*(8-9), 523–541. https://doi.org/10.1007/s00422-012-0512-8

Friston, K. J., Trujillo-Barreto, N., & Daunizeau, J. (2008). DEM: A variational treatment of dynamic systems. *NeuroImage*, *41*(3), 849–885. https://doi.org/10.1016/j.neuroimage.2008.02.054

Holmes, E., Parr, T., Griffiths, T. D., & Friston, K. J. (2021). Active inference, selective attention, and the cocktail party problem. *Neuroscience & Biobehavioral Reviews*, *131*, 1288–1304. https://doi.org/10.1016/j.neubiorev.2021.09.038

Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*(1-2), 99–134. https://doi.org/10.1016/S0004-3702(98)00023-X

Laar, T. van de, Senoz, I., Özçelikkale, A., & Wymeersch, H. (2021). Chance-constrained active inference. *arXiv Preprint arXiv:2102.08792*.

Laar, T. W. van de, & Vries, B. de. (2019). Simulating active inference processes by message passing. *Frontiers in Robotics and AI*, *6*, 20. https://doi.org/10.3389/frobt.2019.00020

Martínez, E. C., Kim, J. W., Barz, T., & Bournazou, M. N. C. (2021). Probabilistic modeling for optimization of bioreactors using reinforcement learning with active inference. *Computer Aided Chemical Engineering*, *50*, 419–424. https://doi.org/10.1016/B978-0-323-88506-5.50066-8

Millidge, B. (2020). Deep active inference as variational policy gradients. *Journal of Mathematical Psychology*, *96*, 102348. https://doi.org/10.1016/j.jmp.2020.102348

Millidge, B., Tschantz, A., Seth, A. K., & Buckley, C. L. (2020). On the relationship between active inference and control as inference. *International Workshop on Active Inference*, 3–11. https://doi.org/10.1007/978-3-030-64919-7_1

Montague, P. R., Dolan, R. J., Friston, K. J., & Dayan, P. (2012). Computational psychiatry. *Trends in Cognitive Sciences*, *16*(1), 72–80. https://doi.org/10.1016/j.tics.2011.11.018

Moreno, A. R. (2021). *PID control as a process of active inference applied to a refrigeration system*. https://projekter.aau.dk/projekter/files/415131289/1034_PID_Control_as_Active_Inference.pdf

Parr, T., Markovic, D., Kiebel, S. J., & Friston, K. J. (2019). Neuronal message passing using mean-field, bethe, and marginal approximations. *Scientific Reports*, *9*(1), 1–18. https://doi.org/10.1038/s41598-018-38246-3

Parr, T., Rikhye, R. V., Halassa, M. M., & Friston, K. J. (2020). Prefrontal computation as active inference. *Cerebral Cortex*, *30*(2), 682–695.

Penny, W. D., Friston, K. J., Ashburner, J. T., Kiebel, S. J., & Nichols, T. E. (2007). *Statistical parametric mapping: The analysis of functional brain images*. https://doi.org/10.1016/B978-0-12-372560-8.X5000-1

Sajid, N., Ball, P. J., Parr, T., & Friston, K. J. (2021). Active inference: Demystified and compared. *Neural Computation*, *33*(3), 674–712. https://doi.org/10.1162/neco_a_01357

Schwartenbeck, P., FitzGerald, T., Mathys, C., Dolan, R., & Friston, K. J. (2015). The dopaminergic midbrain encodes the expected certainty about desired outcomes. *Cerebral Cortex*, *25*(10), 3434–3445. https://doi.org/10.1093/cercor/bhu159

Smith, R., Friston, K. J., & Whyte, C. J. (2022). A step-by-step tutorial on active inference and its application to empirical data. *Journal of Mathematical Psychology*, *107*, 102632. https://doi.org/10.1016/j.jmp.2021.102632

Smith, R., Kirlic, N., Stewart, J. L., Touthang, J., Kuplicki, R., Khalsa, S. S., Feinstein, J., Paulus, M. P., & Aupperle, R. L. (2021). Greater decision uncertainty characterizes a transdiagnostic patient sample during approach-avoidance conflict: A computational modelling approach. *Journal of Psychiatry & Neuroscience*, *46*(1), E74. https://doi.org/10.1503/jpn.200032

Smith, R., Schwartenbeck, P., Stewart, J. L., Kuplicki, R., Ekhtiari, H., Paulus, M. P., & Tulsa 1000 Investigators. (2020). Imprecise action selection in substance use disorder: Evidence for active learning impairments when solving the explore-exploit dilemma. *Drug and Alcohol Dependence*, *215*, 108208. https://doi.org/10.1016/j.drugalcdep.2020.108208

Tison, R., & Poirier, P. (2021). Communication as socially extended active inference: An ecological approach to communicative behavior. *Ecological Psychology*, *33*, 197–235. https://doi.org/10.1080/10407413.2021.1965480

Tschantz, A., Baltieri, M., Seth, A. K., & Buckley, C. L. (2020). Scaling active inference. *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/IJCNN48605.2020.9207382

Tschantz, A., Millidge, B., Seth, A. K., & Buckley, C. L. (2020). Reinforcement learning through active inference. *Bridging AI and Cognitive Science at the International Conference on Learning Representations*. https://baicsworkshop.github.io/pdf/BAICS_37.pdf

Tschantz, A., Seth, A. K., & Buckley, C. L. (2020). Learning action-oriented models through active inference. *PLoS Computational Biology*, *16*(4), e1007805. https://doi.org/10.1371/journal.pcbi.1007805

Ueltzhöffer, K. (2018). Deep active inference. *Biological Cybernetics*, *112*(6), 547–573. https://doi.org/10.1007/s00422-018-0785-7

Vanderbroeck, M., Baioumy, M., Lans, D. van der, Rooij, R. de, & Werf, T. van der. (2019). Active inference for robot control: A factor graph approach. *Student Undergraduate Research E-Journal!*, *5*, 1–5.

Wirkuttis, N., & Tani, J. (2021). Leading or following? Dyadic robot imitative interaction using the active inference framework. *IEEE Robotics and Automation Letters*, *6*(3), 6024–6031. https://doi.org/10.1109/LRA.2021.3090015