Introduction to categorical cybernetics

Jules Hedges[2,3]
joint work with
Dylan Braithwaite[1] Matteo Capucci[1] Bruno Gavranović[1]
Fabrizio Romano Genovese[2,3] Eigil Rischel[1] Riu Rodríguez Sakamoto[1]
Toby Smithe[4] André Videla[1,3] Philipp Zahn[2,3]

[1] MSP Group, University of Strathclyde
[2] 20squares
[3] Institute for Categorical Cybernetics
[4] Verses

Cybernetics??

- The meaning of “cybernetics” has drifted over time - no biotech here!
- The original 1960s meaning is roughly: control theory of complex systems
- With a strong impression of being interdisciplinary (mathematics, computer science, engineering, economics, biology, …)
- From Greek “Kubernetes” meaning “governance” or “helm of a ship”
- We’re fighting against the Cybermen and the Borg to reclaim the meaning

Categorical cybernetics

- The phrase categorical cybernetics or CyberCat has 2 levels of meaning
- Surface level meaning: using anything that looks like category theory to do anything that looks like cybernetics
- More specific meaning: a small collection of categorical tools that keep coming up unreasonably often
- {Specific examples of chain rules} + {general theory of chain rules} + {general-purpose implementation}

Lenses

- A lens $f : (X, X') \to (Y, Y')$ is given by a forwards pass $f : X \to Y$ and a backwards pass $f' : X \times Y' \to X'$

- Lenses compose by the chain rule

- $(g \circ f)'_x(z') = f'_x(g'_{f(x)}(z'))$

- This definition makes sense in any finite product category

Reverse derivatives as lenses

- Every smooth function $f : \mathbb{R}^m \to \mathbb{R}^n$ has a transpose Jacobian $J(f)^\top : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m$
- Linear in the second argument
- The ordinary chain rule says $J(g \circ f)^\top_x = J(f)^\top_x \cdot J(g)^\top_{f(x)}$
- This says exactly that $J^\top$ is a functor {smooth functions} → {lenses}
- The foundational idea of differential geometry & backprop

Bayesian lenses

- A Markov kernel is $f : X \to \Delta(Y)$ aka conditional distribution $\mathbb{P}_f[Y \mid X]$
- If we have a prior on X and make an observation of f's output we can get a posterior on X by Bayes' law
- This defines $B(f) : \Delta(X) \times Y \to \Delta(X)$
- Chain rule for Bayes: $B(g \circ f)_\pi(z) = B(f)_\pi(B(g)_{f(\pi)}(z))$
- This says that Bayes' law is a functor {Markov kernels} → {lenses}

Value iteration

- Suppose we have a Markov decision problem with transition probabilities $f : S \times A \to \Delta(S \times \mathbb{R})$
- We would like to estimate the long-run values $V : S \to \mathbb{R}$
- Value iteration says $V_{i+1}(s) = \mathbb{E}[u + \beta V_i(s') \,|\, (s', u) = f(s, a)]$
- Different ways of choosing a correspond to flavours of RL
- This converges to the infinite discounted sum $V(s) = \mathbb{E}[\sum_{i=0}^{\infty} \beta^i u_i]$

Value iteration with lenses

- We can package the value iteration step into a lens $f : (S, \mathbb{R}) \to (S, \mathbb{R})$
- Its forwards pass is $f(s) = s'$, saying how the policy changes state
- Its backwards pass is $f'_s(v) = u + \beta v$, current payoff + discounted continuation payoff
- If $V_i : (S, \mathbb{R}) \to (1,1)$ is an estimate of the value function, then $V_{i+1} = V_i \circ f$ is a better estimate
- This also works for action-value functions, aka Q-matrices

Dependent lenses

- The definition of lenses still works when the backwards pass has types indexed by the forwards pass, allowing statically-typed branching
- In pseudo-Agda: $f' : (x : X) \to Y'_{f(x)} \to X'_x$
- The category [indexed types, dependent lenses] is (non-trivially) equivalent to [polynomial functors, natural transformations]
- Put in rich structure, get even more rich structure back
- F-lenses over-generalise this to: category $\mathscr{C}$ + functor $\mathscr{C}^{\mathrm{op}} \to \mathbf{Cat}$

Optics

- Optics generalise lenses to any monoidal category (and to actegories)
- Key example: categories of Markov kernels - Bayesian open games
- Require almost no structure as input, get back as much as lenses
- Even in cartesian settings, optics are operationally better - they memoise instead of recompute the forwards pass
- tl;dr Optics are better than lenses in every possible way

Dependent optics

- Question: how to get the best of both worlds between optics (non-cartesian, better operationally) and dependent lenses (richer structure, especially branching)

- Partial answer: indexed optics - compute the coproduct completion of optics

- The full answer:

The Para construction

- For any monoidal category, a parametrised morphism $X \to Y$ is P + $P \otimes X \to Y$
- Para + lenses synergise very well: $(P \otimes X, P' \otimes X') \to (Y, Y')$
- A general theory of controlled processes - central to cybernetics
- P is the control, P' is the feedback to the controller
- Compose horizontally (along processes) + vertically (along controllers)

Variational inference

- Bayesian inverses are hard to compute
- So we make the backwards pass of Bayesian lenses parametrised
- Loss functions, eg. KL divergence, variational free energy measure the failure to be the exact Bayesian inverse
- Variational inference for composite processes can be done “locally” exactly like backprop + gradient descent
- Working towards a fully compositional account of active inference

Open games

- Open games = parametrised optics + counterfactual optimisation
- Forward pass: actions in a game
- Backward pass: (counterfactual) payoffs

The open game engine

- Open-source implementation of Bayesian open games in Haskell
- A domain specific language for specifying them, with variable binding syntax
- Behaves like a model checker for Bayesian Nash equilibria
- Used for real-world modelling at 20squares
- Suffers from drawbacks coming from Haskell + the DSL design
- Programming with an explicit backwards pass is a huge barrier

RL in the open game engine

- Open games but runs multi-agent Q-learning instead of equilibrium checking
- Frontend in Haskell, backend in Python rllib
- Currently closed source
- Used for simulations to study algorithmic collusion

“Diegetic open games”

- A deep categorical account of where the backwards pass comes from
- Key idea: there is a lens from pairs of gradients to gradients of pairs, $(X \times Y, T(X) \times T(Y)) \to (X \times Y, T(X \times Y))$
- Leads to a completely functorial account of parametrised optics
- In game theory, “tangent vector” = payoff matrix
- This lens contains the essence of Nash equilibrium

The next implementation

- Learning from the mistakes of the open game engine
- Use theory of diegetic open games to make the backwards pass implicit
- Deep rather than shallow embedding - no maintaining a parser
- No longer specialised to game theory
- Current status: prototyping in Haskell, only partially working yet

The big picture

- The €1,000,000 question: so what?
- Categorical methods can’t do anything genuinely new
- Distinction between practical compositionality and its mathematical theory
- We are pinning down folklore (“x looks kinda like y”) in a precise way
- One hope: that a general-purpose implementation will be genuinely useful

The CyberCat Institute

- A scheme of Philipp Zahn & me
- Idea: try to organise researchers better than we can in a university
- Idea: put researchers and software engineers on equal footing
- Neither completely blue-sky nor profit-driven R&D
- Neither academia nor business but taking the most useful parts from both
- Specific example: try funding devops work on an academic research grant
- Current status: actively looking for funding

Some references

- Cruttwell, Gavranović, Ghani, Wilson, Zanasi - Categorical foundations of gradient-based learning (arXiv:2103.01931)
- Smithe - Mathematical foundations for a compositional account of the Bayesian brain (arXiv:2212.12538)
- Braithwaite, Hedges - Dependent Bayesian lenses (arXiv:2209.14728)
- Hedges, Sakamoto - Value iteration is optic composition (arXiv:2206.04547)

Some more references

- Capucci, Gavranović, Hedges, Rischel - Towards foundations of categorical cybernetics (arXiv:2105.06332)

- Capucci - Diegetic representation of feedback in open games (arXiv:2206.12338)

- https://github.com/CyberCat-Institute/open-game-engine

- Genovese - Experiments in modelling a disaster - https://hackmd.io/@3hIMf3V6RQWVUTzjzKEQuw/SyM8N6jSo

- https://cybercat.institute/ , https://cybercat-institute.github.io/