ICLR 2019 Notes
New Orleans, LA, USA

David Abel[*]
david_abel@brown.edu

May 2019

Contents

[*] http://david-abel.github.io

This document contains notes I took during the events I managed to make it to at (my first) ICLR, in New Orleans, LA, USA. Please feel free to distribute it and shoot me an email at david_abel@brown.edu if you find any typos or other items that need correcting.

1 Conference Highlights

Sadly, I missed more talks than normal this conference (and had to fly out a bit early so I missed a good chunk of Thursday). Some highlights:

- Lots of continued discussion following the recent wave of conversations around Rich Sutton's bitter lesson[1]. The debate held at the main conference track (see Section 4.4) and the panel at the SPiRL workshop[2] (see Section 2.2.11) featured lots of insights about the topic – highly recommended to check them out!
- The SPiRL workshop was *outstanding*. The speaker lineup, contributed talks, and panel were all exceptional (see Section 2.2). A huge thanks to the organizers for putting on such a great event.
- Hot topics: 1) Meta learning is very popular (particularly meta RL), and 2) Graph neural networks.
- Favorite talks: I loved the keynotes! I didn't make it to all of them but the ones I did catch were all fantastic. Definitely check out Prof. Zeynep Tufekci's amazing talk if you can find a video (my summary is in Section 4.3). If I happen to find a recording I'll link it here.
- Some really nice papers in learning abstraction/hierarchies in RL: 1) "Near-Optimal Representation Learning for Hierarchical Reinforcement Learning" by Nachum et al. [30]; 2) "Learning Multi-Level Hierarchies with Hindsight" by Levy et al. [26]; and 3) "Learning Finite State Representations of Recurrent Policy Networks' by Koul et al. [23].
- A small thing, but I *loved* how large the poster were allowed to be (something like 1.5 meters (5 feet) wide). Encouraged some great designs and easy viewing for the onlookers, even from a distance.

[1] http://incompleteideas.net/IncIdeas/BitterLesson.html

[2] http://spirl.info

2 Monday May 6th: Workshops

The conference begins! Today we have a few keynotes and the workshops.

2.1 Keynote: Cynthia Dwork on Recent Developments in Algorithmic Fairness

The field (of algorithmic fairness) began around 2010, but today we'll talk about brand new developments.

2.1.1 Algorithmic Fairness

Point 1: Algorithms are unfair, data are unrepresentative, labels can embody bias.

Point 2: Algorithms can have *life altering consequences.*

- Mortgage terms.
- Detention/release.
- Medical assessments and care.
- Deciding if a child is removed or not from a home.

→ Lots of papers that say: "we're shocked by these examples of algorithmic bias!". But now we're in a position to do something about it.

Algorithmic Fairness:

1. Natural desiderata of fairness conflict with each other
2. One piece of an unfair world. Deployment can be unfair, too

Goal: Develop a *theory* of algorithmic fairness. Two groups of fairness definitions:

1. Group fairness
2. Individual fairness

Definition 1 (Group Fairness): *Statistical requirements about the relative treatment of two disjoint groups.*

Example of group fairness: demographics of students accepted to a college should be equal. Or, balance for positive/negative class.

Definition 2 (Individual Fairness): *People who are similar with respect to a given classification task should be treated similarly*

→ Comes from a strong legal foundation.

Problems:

- Group notions fail under scrutiny
- Individual fairness requires a task specific metric.
  → Paucity of work on individual fairness because we need such specific metrics.

2.1.2 Approaches to Fairness

Metric Learning for Algorithmic Fairness:

- Adjudicator has an intuitive mapping from high dimensional feature vector (X) to the important aspects of the problem (Z).
- Relative queries are easy (which of A and B is closer to C?)
- Absolute queries are hard (what is $d(A, B)$?) → Idea: turn to learning theory.
- Three insights in trying to answer above absolute queries:
  1. Distance from a single representative element produce useful approximations to the true metric.
  2. Parallax can be achieved by aggregating approximations obtained from a small number of representatives.
  3. Can generalize to unseen elements.
- See also: Bridging the Group vs Individual Gap [16, 21]:

2.1.3 Hybrid Group/Individual Fairness Approaches

Consider individual probabilities: 1) what is the probability that P will repay a loan? 2) What is the probability that a tumor will metastasize?, and so on.

→ One concern: these events will just happen once. How should we think about these in terms of giving medical/legal recommendations? How can we justify the answer?

Philip Dawid wrote a recent survey of individual fairness definitions [7].

One idea: calibration. Consider forecasting the weather. When we say 30% chance of rain, we mean that 30% of the days we predict 30% rain will rain, and 70% will not.

The Tumor Example: Expectations are obtained from binary outcome data.
→ Study A says 40% chance of a tumor, and Study B says 70% (but not training data/context, just the studies output).

So, given $C = \{S_1, S_2\}$, consider the venn diagram formed by the recommendation of the two studies. We can choose values for elements $P = S_1\ S_2$, $Q = S_1 \cap S_2$, and $R = S_2\ S_1$, that retain the given expectations. This can help us clarify the appropriate decision.

But: many multi-accurate solutions. If, however, we had ensured calibration, we *can* narrow down the expectation to something accurate.

The Loan Example:

- Intersecting demographic/ethnic/age/gender/etc/ groups.
- Minimally: policies consistent with expected repayment rates for each group.

Q: Who decides which groups should be prioritized? The culturally dominant? The oppressed? How do we set our scoring function? Really hard question [19]

A: Let's turn to complexity theory!
→ All groups identifiable by small circuits acting on the given data.

Conjecture 2.1. *Captures all historically disadvantaged groups S.*

Multi-accuracy and multi-calibration: we can do it!

- Multi-Accuracy: Complexity of creating the scoring function depends on hardness of (agnostic) learning of C, but function is efficient.
- Multi-calibration: f is calibrated on each set $S \in C$ simultaneously, accurate in expectation.

Problem: The Devil is in the collection of C.

→ We hope we capture task specific semantically significant differences.

Q: What are the sources of information available to child protective services and call screening?

2.1.4 Fair Ranking

Q: Why?

A1: Ranking is crucial to many endeavors: the heart of triage, underlying impetus for scoring, rank translates to policies or to scores in clinical trials.

A2: Thinking about ranking can help us in thinking about a scoring function more generally.

Idea: Let's think about fair ranking from the cryptographic perspective.

Rank Unfairness:

- Suppose we have two groups of people: A and B.
- Suppose $\mathbb{E}[A] > \mathbb{E}[B]$.
- But! It's silly to then rank everyone in A above everyone in B.

Take a cryptographic/complexity theoretic approach to address this problem!
→ If positive and negative examples are computationally indistinguishable, the best one can do is assign to everyone a probability according to the base rate.

2.1.5 Approaches from Representation Learning

Idea: Learn a "fair" representation (in group fairness).

- Stamps out sensitive information ("censoring")
- Retains sufficient information to permit standard training.

Goal: learn a censored mapping to a lower dimensional space Z [10].

- Encoder tries to hide membership bit, permit prediction on Z.
- Decoder tries to reconstruct x from $z = Enc(x)$
- Adversary (A) tries to distinguish $Enc(x \in S)$ from $Enc(x \in S^c)$.

→ Approach by Madras et al. [28] tie this objective to scoring, show that transfer is possible.

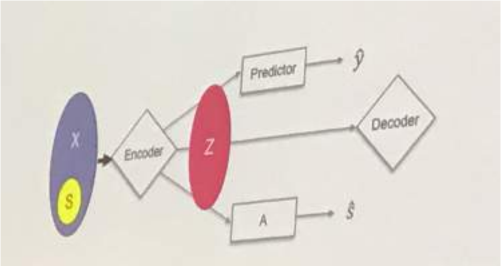


Figure 1: The cryptographic setup for learning fair representations.

The Harvard-Stanford Problem: Suppose you're at Stanford, and you build an algorithm for detecting tumors that works really well. Suppose someone else is at Harvard and does the same.

→ Claim: the algorithms wont work across the populations due to differences in the groups and in the lab settings.

Goal, then, is to find a way to identify differences/similarities across populations so that these methods *can* be transferred across populations.
Approach:

- Choose $y \sim Bernoulli(\text{base rate})$
- Choose $x \sim N(\mu, \Sigma)$
- Retain x if $Bernoulli(\sigma(f_1, s_1)) = f_2 6i(x_2) = y$.

Summary:

- A fair algorithm is only one piece of an unfair world
- Multiple kinds of fairness: group, individual, Multi-X.
- Breakthrough in metric learning for individual fairness
- Individual probabilities are hard to understand, but we can learn from fairness methods to improve their use.
- Censored representations and out-of-distribution generalization.

....

Now off to the Structure and Priors in Reinforcement Learning workshop!

2.2 SPiRL Workshop

First, Pieter Abbeel on Model-based RL!

2.2.1 Pieter Abbeel on Model-Based RL from Meta RL

Few-shot RL/Learning to RL: we have a *family* of environments, $M_1, M_2, \ldots, M_n$. Hope is that when we learn from these n environments, we can learn faster on environment M_{n+1}.

Fast Learning:

$$\max_{\theta} \mathbb{E}_M \mathbb{E}_{\tau_M} \left[\sum_{i=1}^{n} R_{\tau_M}^i \mid \text{RLAgent}_\theta \right]. \tag{1}$$

Objective is roughly the above, where we ground RLAgent_θ as an RNN (or some othe generic computational architecture).

Other ways to solve this objective like Model-Agnostic Meta Learning (MAML) [11].

Point: Family of methods that let you train quickly in new environments (new R, T), given interactions with prior environments.

Motivation for Simulation:

- Less expensive (can't break robot).
- Faster/more scalable.
- Easier to get lots of labels.

Q: How can we leverage *crude* simulation through domain randomization?

→ Think about minecraft – there's some visual structure that is useful for learning about the world, but it's not perfect. How can we train in minecraft (or similar domains) and transfer to the world?

A: Randomize aspects of simulations in order to make training of vision system easier. Can then transfer these trained perceptual classifiers to the world, and it works! (same goes for grasping)

A: Same goes for grasping – can train to grasp in simulation, then grasp real objects (with around 80% success rate).

Result: Train a hand controller (for manipulating a block in hand) in simulation, can actually deploy it on a real robot.

Definition 3 (Model-Free RL): *Interact with the world and collect data D. Then, use this data to inform π or Q, and use those to act.*

Definition 4 (Model-Based RL): *Interact with the world and collect data D. Then, use this data to inform a world simulator, $\widehat{M}$, perform simulations in order to act.*

Canonical model-based RL;

```
1. For iter $= 1, 2, \ldots$
2.     Collect data under current policy
3.     Improve learned simulator from all past data
4.     Use simulator to act and collect new data.
```

Problem: learned models are imperfect!

Fix: Learn a better simulator. But, this is insufficient. Extremely hard to learn the *right* simulation.

→ New overfitting challenge in model-based RL. Policy optimization tends to exploit regularities in simulator, leading to catastrophic failures.

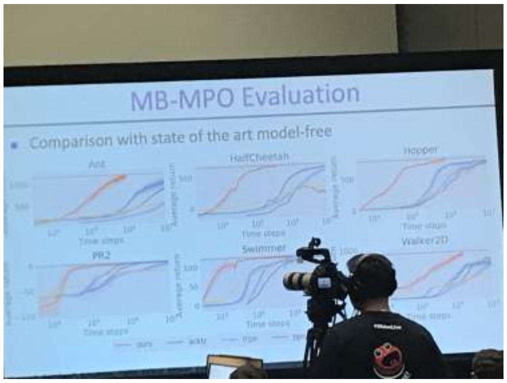
Key Idea: Don't need to learn an accurate model, just need to learn a set of models representative of the real world (and do few-shot RL):

```
1. For iter $= 1, 2, \ldots$
2.     Collect data under current adaptive policies, $\pi_1, \ldots, \pi_k$
3.     learn ENSEMBLE of $k$ simulators from past data.
4.     meta-policy optimization over ensemble
5.         new meta-policy $\pi_\theta$
6.         new adaptive policies $\pi_1, \ldots, \pi_k$.
```

Experiments:

1. *MuJoCo:* With about 45 minutes of real interaction with the MuJoCo environment, state-of-the-art model-free methods can't learn, while the meta-learning approach can.
2. *Robots:* Similarly, can train meta model-based RL in a sample efficient way to learn to perform robotic grasping tasks.

   → Between 10-100× more efficient than model-free methods *to get to the same asymptotic performance.*



MB-MPO Evaluation

- Comparison with state of the art model-free

Figure 2: Results from meta model-based RL (and a camera + person).

Challenge Question: Hierarchical RL promises to solve long horizon planning problems.

Q1: Are model-free and model-based HRL fundamentally different approaches or are they the same?

Q2: Do you think there's a way to combine these two?

Pieter A: Yeah absolutely! The methods we presented do some work to combine these methods together. For HRL it might be a bit different, though. In some sense, at the high level, humans don't do RL. We don't have enough "high level" trials to explore things like "go to conference/do a PhD". So, the higher level is probably model-based and more planning based rather than RL. Another thing that comes to mind is that HRL approaches seem to only have two levels. One interesting direction is to generalize with more depth/levels rather than two levels. Still not obvious where the separation between model-based/model-free methods takes place.

Q: Pros and Cons of looking for an end-to-end approach as opposed to a more modular approach (with more intermittent structure like state estimation).

Pieter A: There's no final answer here – state estimation sometimes involves human provided information, might lose the wrong data for doing control (what is state for a self driving car, for instance?). But, some priors in this way can probably help!

2.2.2 Contributed Talk: Kate Rakelly on Off Policy RL via Context Variables

Goal: Design agents that are skilled a variety of different tasks.

→ But, training agents in new tasks is statistically/computational infeasible, so really we'd like to exploit shared structure across tasks.

Approach (High-Level): Meta-RL to learn shared structure across related tasks.

Problems: Rapid adaptation requires efficient exploration strategies, while meta-training requires data from each task, exarcerbates sample inefficiency.

Approach (Detailed View): Meta-RL via an *off-policy* approach. But, raises a problem with exploration since we no longer control the experience distribution.

"Context": task-specific information learned from a task (z). Meta-training then has two pieces:

1. Learn to summarize the context into z.
2. Learn to take actions given s, z.

Algorithm: Pearl. Given a new task, main idea is to maintain a probability distribution over which task we're in. This lets us exploit knowledge of uncertainty to adapt efficiently to the new task.

Experiments: four different MuJoCo domains (half cheetah, humanoid, ant, walker). Rewards and dynamics change across tasks (locomotion direction, velocity, joint parameters).
Summary:

- Pearl is the first off-policy Meta-RL algorithm
- 20-100× improved sample efficiency on the domains tested
- Posterior sampling for efficient exploration during adaption.

Code: `github.com/katerakelly/oyster`

Posters now for a bit.

2.2.3 Matt Botvinick on Meta-RL: An Appreciation

Point: We need some structure to scale RL. What I have in mind is something like relational RL, objects, graph nets, and so on.

Guiding Question: What do the algorithms that meta-RL gives rise to, do? What can't they do?

Recent survey summarizing some of the ideas Botvinick et al. [4].

→ The field seems to have moved on from meta-RL, but I can't! Let's really understand these algorithms.

Tendency: let's build a faster race car. This talk: let's understand fast race cars, or why these things we've made recently are fast!

Observations in trying to understand Meta-RL:

- Consider two-armed bandits: An animal chooses between two arms, payoff determined according to some payoff schedule. Critically, sources of reward get restocked every so often.

  → Animals match their frequency in choices with the frequency with which they get rewards (probability matching).

  → Ordinary LSTM figures that out, too! Also figures out the Gittins optimal thing in the regular $\beta - Bernoulli$ bandit.

- Consider this new bandit task: probabilities of payoff keep changing, but volatility changes, too (long intervals where payoffs flip around, etc.).

  → Smart thing is to change your learning rate. People do in fact do this (if we fit a model predicting learning rates to peoples' decisions).

  → Meta-learned LSTM can do the same thing!

- Monkey sees two colored face down cups, one of which hides a raisin. Learn to pickup the cup hiding the raisin in general.

  → Then the monkey has to transfer to a new task with new objects. Turns out monkey learns to transfer as well; when no info can be used, monkey explores uniformly, when info can be exploited, monkey learns very quickly.

  → LSTM can do this, too!

Clear illustration of a meta model-free algorithm giving rise to a model-based RL, based on model-based tests designed for people/animals from Daw and Dayan [6].

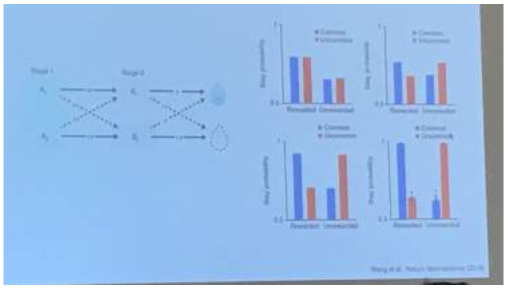


Figure 3: A task for determining whether a decision making is using model-based techniques or not (left) and results from different approaches on the task (right).

Point 1: Meta-RL can also give rise to algorithms that can solve temporal credit assignment problems, too.

Point 2: For LSTM to be solving certain kinds of tasks, they also must be retaining some sufficient statistics in their state (that roughly match what the observer would track, too).

Lastly: a crazy idea (Dave: Matt's words, not mine :)). We have some evidence that with ordinary LSTMs, model-free RL can give rise to model-based RL (in the right situation).

→ Maybe, if we set of the environments in the right way, maybe LSTMs can work out algorithms for doing (amortized?) Bayesian inference.

Take Home Message: Meta-RL is really exciting, let's keep coming up with faster algorithms but also try to understand what they're doing.

2.2.4 Katja Hoffman on Challenges & Directions in Multi-Task RL

Q: Why do we focus on structure and priors in RL?

Katja A: Three things!

1. Improve sample efficiency.
2. Improve sample efficiency.
3. Improve sample efficiency.

Q: When you think about structure and priors in RL, what kinds of challenges become solvable?

A: Maybe in games, science, transportation, medicine?

→ For these (and probably other) domains, structure is crucial.

Kinds of Structure:

- Assume multiple related tasks and useful relationship between tasks can be learned from data
- Q: What types of models allow learning and use of related task structure¿?
- Q: What trade-offs can we achieve between prior assumptions, flexibility, and sample efficiency?

First Approach: Meta-RL with Latent Variable Gaussian Processes [33]. Idea:

- Problem: assume p task with related dynamics:

$$y_t^p - f(x_i^p, h_p) + \varepsilon, \tag{2}$$

- Observe data from training tasks
- Goal: accurately predict held out test dynamics with minimal additional data
- Approach: Model-Based RL via latent variable Gaussian processes. Place a GP prior on global function.

- Experiments: 1) toy tasks, multo-task prediction. Approach is able to disentangle unseen tasks; 2) multi-task cart pole. System can vary in mass and pendulum length, many held out settings of these parameters.

Second Approach: (CAVIA) Fast Context Adaptation via Meta-Learning [37].

- Problem: distribution over training and test task p_{train}, p_{test}.
- During meta-training, sample tasks from p_{train}, get train/test data for that task
- Learn how to adapt quickly on the task by splitting up network into: 1) task specific context parameters ϕ, and 2) shared parameters θ.
- Experiments: 1) supervised learning task; 2) multi-task half cheetah.

  → CAVIA learns an interpretable task embedding, captured in context parameters → Adapts to test tasks with updates to only context parameters – sheds new light on meta-learning benchmarks. → Very flexible

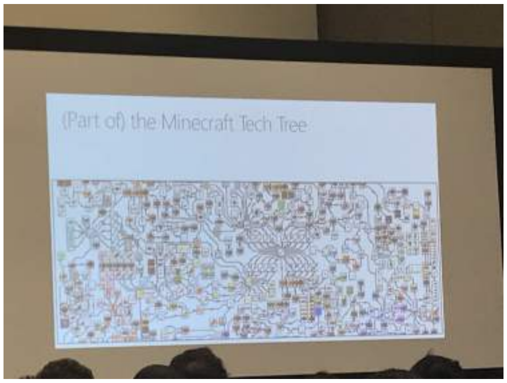
Follow up: Variational Task Embeddings for Fast Adaptation in Deep RL. Learns to trade of exploration and exploitation online, while interacting with the environment. VATE can deduce information about the task before seeing any reward.

Point: As we push forward in RL to harder domains, there might be certain generic structure that tends to work well across many domains. Can we find this unifying structure?

→ One thing that might be needed is a dataset that might rive rise to such structure. To this end:

→ MineRL: competition (`minerl.io/competition`) on sample efficient RL using human priors (upcoming at NeurIPS this year), built on top of MALMO [18].

Minecraft is massively complex, so offers a great platform for exploring the use of priors in RL. See Figure 4 for a sense of the size of the tech tree.



(Part of) the Minecraft Tech Tree

Figure 4: Part of the tech tree in Minecraft

2.2.5 Tejas Kulkarni on Object-Centric Representations RL

Point: The hard problems of RL:

1. State estimation

   RL does not prescribe a detailed recipe to represent state. Hand specify or learn a state representation.

2. Exploration

   And: how you explore depends on how you represent the world.

This work: let's use self-supervised deep learning to learn object structure.

Example: an individual blind from birth can still draw rough object structure, including *perspective* [20] – see Figure 5.

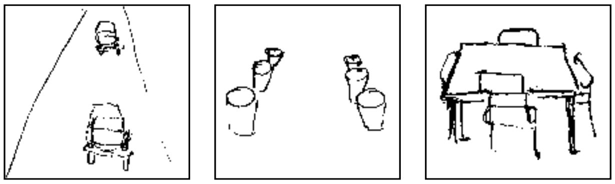


Figure 1. Drawings by E.A. using convergence in the *z* dimension: a road with two cars, rows of glasses, and a table and chairs.

Figure 5: Drawings from Kleinberg et al. [22]

Objects in particular are a fundamental and important abstraction.

Q: So, can we learn them?

Three properties from object-centric representation in physical domains:

1. Capture spatio-temporal features at degrees of freedom.

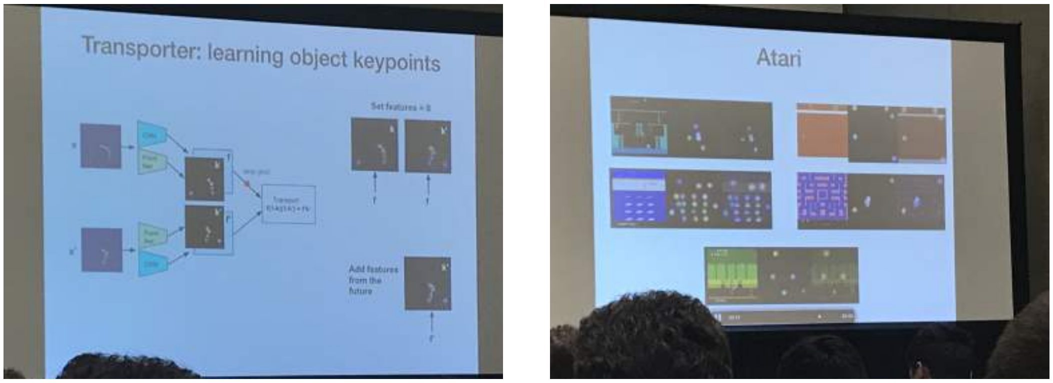
2. Long-term temporal consistency.

3. Captures basic geometry of the environment.

A: Yes! The "Transporter" network (see Figure 6).
→ Transporter network does a good job for capturing salient objects in image based problems like Montezuma. Trained on uniform random policy, and the result is a spatial map of where different objects area.

But: the above approach only tracks moving objects, not stationary ones. So, we can pair it with instance-based segmentation for finding stationary objects.

Next Stage: Object-Centric RL and Planning in high dimensional domains, building on the earlier work by Diuk et al. [9].


Transporter: learning object keypoints
Add features from the future
Atari

(a) Transformer Network (b) Object Classification

Figure 6: Transporter Network (left) maps source to target image via compressed geometric representation, and some candidate objects found (right) in different Atari games.

→ Key Problem 1: Structured exploration in the space of objects and relations.

→ Key Problem 2: Generalization in the form of self generated tasks in the space.
Object-centric HRL architectures [24, 8].

Major frontier: data efficient RL in hard exploration settings like Montezuma. Idea: systematically explore in the space of objects and relations.
Challenge Question: In supervised learning, progress in representation learning and transfer learning for vision has been largely driven by ImageNet. NLP had its “ImageNet” moment with GPT-2 and BERT. So, will there be a analogous “ImageNet” moment in RL that allows us to learn general purpose data-driven priors?

A: Yeah, absolutely! I think we’re almost there. Lots of folks working in this direction, I think we are right around the corner.

2.2.6 Tim Lillicrap on Learning Models for Representations and Planning

Current State and Limitations of Deep RL:

1. We can now solve virtually any single task/problem for which:
   (a) Formally specify and query the reward function
   (b) Explore sufficiently and collect lots of data
2. What remains challenging:
   - Learning when a reward function is difficulty to specify
   - Data efficiency and multi-task transfer learning

We measure outcomes via: $R(\tau) = \sum_{t=0}^{T} \gamma^t r_t$, with the objective function:

$$J(\theta) = \int_{\mathbb{T}} p_\theta(\tau) R(\tau) d\tau. \tag{3}$$

But: In model-free RL we tend to throw away what we know about the task to solve it.

Clear structure to introduce: plan with a model.

→ Tricky! Getting this model is really hard. If we can get it, we know it can be really powerful (see AlphaZero [35]).

Problem: Planning with learned models is really hard. (Tim said it became a running joke to started up a model-based RL project at DeepMind in the last few years: no one expected it to work).

Idea: Hybrid model-free and model-based approaches. By augmenting previous algorithms with a learned model did in fact help on a goal-finding task.

Planning with Learned Models: PETS [5], Deep Planning Network (Planet) [12].

Experiments: continuous control from image observations (finger, cheetah, cup, etc).

→ Some versions of this end up working well! Around 1k-2k episodes it can solve image-based MuJoCo problems.

Conclusions:

- Model-based algorithms hold promise of addressing data efficiency and transfer learning limitations.
- Beginning to develop working recipes that allow planning with models in unknown environments.
- Necessary and sufficient conditions for planning with learned models are unclear.
- Much work remains!

Challenge Question: What are the trade-offs in rolling value estimation and perception into the same architecture?

Tim A: I don't know anyone that's systematically studied this kind of thing, but it's definitely important to study it more. Some insights can be gathered from AlphaZero, ELO rating analysis. Lots more to do!

2.2.7 Karthik Narasimhan on Task-agnostic Priors for RL

Current State of RL: success of model-free RL approaches (see: Go, DOTA).

→ All of these feats require huge amounts of time and samples (like 45,000 years of game play for DOTA).

→ Little to no transfer of knowledge.

Recent Approaches:

- Multi-task policy learning
- Meta-learning
- Bayesian RL
- Successor Representations

Observation: all approaches tend to learn policies, which are rigid and hard to transfer to other tasks.

Solution: model-based RL.

→ Approach: bootstrap model learning with task agnostic priors. The model is 1) more transferable, 2) expensive to learn but can be made easier with priors.

Q: Can there be a universal set of priors for RL?

A: Look to how humans learn new tasks. These priors seem to come from 1) a notion of intuitive physics and 2) language.

Project 1: Can we learn physics in a task agnostic way? Moreover, can this physics prior help sample efficiency of RL?

Lots of prior work in this area, but they are task-specific.

→ This work: learn physics prior from task-independent data, decouple the model and policy.

Overview of approach:

- Pre-train a frame predictor on physics videos
- Initialize dynamics models and use it to learn policy that makes use of future state predictions.
- Simultaneously fine-tune dynamics model on target environments.

Two Key Operations: 1) isolation of dynamics of each entity in the world, 2) accurate modeling of local spaces around each entity.

Experiments: PhysWorld and Atari – in both cases, use videos containing demonstrations of the physics of the environment to do some pre-training of the dynamics model (in Atari, pre-training is still done in PhysWorld). Results show the approach works very well, both at 10-step prediction and at helping sample efficiency in RL.

Project 2: Can we use language as a bridge to connect information we have about one domain to another, new domain?

Overview of approach:

- Treat language as task-invariant and accessible medium.
- Goal: transfer a model of the environment using text descriptions.

  Example: "Scorpions can chase you". Might be able to learn a model that places high probability on the scorpion moving closer to the agent location.
- Main technique: transfer knowledge acquired from language to inform a prior on the dynamics model in a new environment.

Conclusions:

1. Model-based RL is sample efficient but learning a model is expensive
2. Task agnostic priors over models provide a solution for both sample efficiency and generalization
3. Two common priors applicable to a variety of tasks: classical mechanics and language.

Challenge Question: Lots of success in Deep RL. New push into hybrid approaches like cross-domain reasoning, using knowledge from different tasks to aid learning, and so on. What are the greatest obstacles on the path to mid-level intelligence?

Karthik A: I would emphasize the need for distributional robustness and transfer – need to look at agents that can transfer across similar domains. Some obstacles involve

2.2.8 Contributed Talk: Ben E., Lisa L., Jacob T. on Priors for Exploration

Challenges in RL today:

1. Exploration
2. Reward function design
3. Generalization
4. Safety

→ Priors are a powerful tool for handling these challenges.

Q: Can we learn useful priors?

A: Yes! This work is about a general algorithm for learning priors. Idea is to frame RL as a two player game, with one player being an adversary choosing a reward function.

Project 1: State marginal matching.
→ Idea is to try to maximize policy state distribution to some objective distribution. Minimize KL between π^* and π.

Experiments: test for exploration and meta-learning capacity of the algorithm. Tested with locomotion and manipulation tasks. Their approach works quite well.

Project 2: Priors for Robust Adaptation.

→ RL with unknown rewards: assume we're given a distribution over reward functions. Then, sample a new reward function and optimize with respect to it.

Main approach: compute the Bayes-optimal policy, and then perform regular RL.

2.2.9 Doina Precup on Temporal Abstraction

Guiding Q: How can we inject temporal abstraction into options?

→ Where do options come from? Often, from people (as in robots).

→ But what constitutes a good set of options? This is a *representation* discovery problem.

Earlier approach: options should be good at optimizing returns, as in the Option-Critic [1]. Option-critic learns option representations that yield fast in-task learning but also effective transfer across tasks.

Point: Length collapse occurs – options "dissolve" into primitive actions over time.

Assumption: executing a policy is cheap, deciding what to do is expensive. So, can use options with an explicit *deliberation cost* in mind [13].

That is, can define a new value function based on the deliberation cost:

$$Q(s, o) = c(s, o) + \sum_{s'} P(s' \mid s, o) \sum_{o'} \mu(o' \mid s') Q(s', o'),$$

with $c(s, o)$ some cost of deliberation.

Experiments: on Atari, with and without deliberation cost (as a regularizer). Indeed find that options take longer before terminating (which was the intended goal).

Q: Should all option components optimize the same thing? (Should $\mathcal{I}, \beta, \pi$ all be geared toward maximizing rewards?)

A: Based on the deliberation cost work, one might think that aspects of the option should take these regularizers into account. See, for instance, the recent work by Harutyunyan et al. [14], or the termination critic [15].

Idea: Bottleneck states – we might want options that take us to these bottlenecks.

→ Drawback: expensive both in terms of sample size and computation.

Discussion:

- Priors can be built into option construction via optimization criteria
- Termination and internal policies of options could accomplish different goals
- *****Biggest Open Question:*** how should we empirically evaluate lifelong learning AI systems?

How do we assess the capability of a lifelong agent?

1. No longer a single task!
2. Returns are important but too simple.
3. How well is the agent preserving and enhancing its knowledge?

→ Proposal: hypothesis-driven evaluation of continual systems. That is, take inspiration from other fields (psychological, for instance).
Challenge Question: Lots of recent work applies deep RL to existing algorithms in HRL and option discovery. What has deep RL brought to the table? Do they fix all of the problems or do we need some new paradigm shift?

Doina A: Neural nets brought to HRL roughly what they brought to regular RL – aspects of the feature discovery problems have effectively been solved. On one hand that's great, because we have a solution. On the other hand, we are still lacking in methods that truly do knowledge discovery. Deep nets are not really an answer to that process. There's a temptation to take a deep net throw it at a problem and put some HRL objectives on the top. Yes, that might work, but it doesn't lead to decomposable or modular knowledge. We've mode lots of progress but perhaps it is a good time for us to take a step back and do fancier things in terms of state abstraction and options.

2.2.10 Jane Wang on Learning of Structured, Causal Priors

Point: Structured priors enable faster learning.

→ *Causal* priors in particular can enable faster learning (by improving exploration, generalization, credit assignment, and so on).

Causal reasoning is a rich field, so, some background:

Definition 5 (Bayes Net): *A probabilistic graphical model that represents a set of variables and their conditional probability distribution in the form of a directed acyclic graph (DAG)*

Definition 6 (Causal Bayes Net): *Bayes net where arrows represent causal semantics*

Definition 7 (Intervention): *Fixing a value of a variable, disconnecting it from its parents.*

Definition 8 (Do-calculus): *A set of tools for making causal inferences given observational data*

Can ask three levels of questions:

1. Association: are drinking wine and me having a headache related?
2. Intervention: If I go drink wine, will I have a headache?
3. Counter-factuals: go back in time and ask, what if I had drank wine? Would I have a headache?

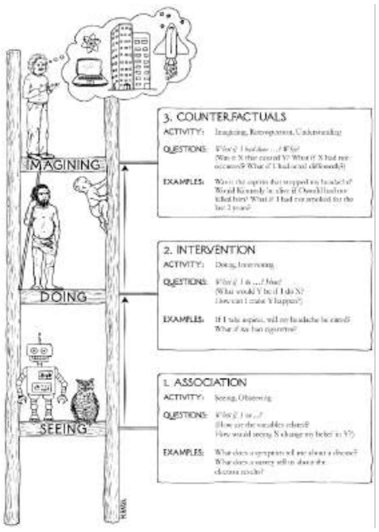


Figure 7: Pearl's ladder of causality

Q: How does causal reasoning manifest in humans?

A (babies): Babies less than a year old do not demonstrate causal knowledge, but do have a sense of physical continuity.

A (2 year olds): Can learn predictive relationships between events, but can't sponetaneously make interventions based on causal understanding.

A (3-4 year olds): Can infer causal maps from observing conditional dependencies.

A (4-5 year olds): Can make informative targeted interventions based on causal knowledge.

A (adolescence): Strategies for causal learning continue to improve.

A (adults): Evidence of an associative bias, a "rich-get-richer" principle: one variable is more likely to be present if others in the causal model are also present.

Overall: evidence suggests that children display ability to perform causal inference from observation (roughly consistent with Bayesian inference). More sophisticated forms of reasoning (performing novel informative interventions) come online later as a result of experience.

But: major deviations from rationality/good inference.

→ Reasons for deviations:

1. Formal models of causal reasoning optimize difference cost functions.
2. Humans do not optimize for a specific causal graph, but a flexible one.

Takeaway: A structured universe of tasks $\implies$ we should use structured priors.

Idea: Meta Learning of causally inspired priors. Similarly to previous talks, assume a distribution of tasks, want to learn some priors that lead to sample efficient learning in new tasks.

Experiments: 1) learning from observations on causal networks (can the agent learn some causal structure from a DAG?); 2) learning from interventions; 3) Learning from instance specific info.

Challenge Question: Why does deep Rl seem to struggle with out-of-sample generalization compared to other domains? ?
Jane A: In RL, lots of ways to be out of sample (in contrast to supervised learning), so it's much harder. Generalization is just harder because lots of things can change: state, action, policy, transition function, reward, function, and so on. RL also requires an ongoing interaction with the environment – input is really dependent on policy, so input will change as you update policy.

2.2.11 Panel: Matt, Jane, Doina, Sergey, Karthik, Tejas, Tim

Q: What is the role of structure vs. data?

Tim: Question is loaded. Depends on what you want to do: if you want to get good at a niche, specify more. If you want a general learning algorithm: specify less.

Tejas: Yes! The article talks about search, but to me the big question is making domains "computable" (simulatable). The article is misguided: where do the primitives come from? Can't rely on the data to give you primitives. We should radically add structure. There are certain truths we can and should exploit to search effectively (objects, agents, physics).

Karthik: Humans have evolved many things over time that are key to our intelligence. Definitely having the right kind of inductive biases and structured priors to get things to work.

Matt: I want to push back on that, because I hear it from psychologists. The assertion is often made that because babies have strong inductive biases, our agents have inductive biases. But its not obviously a constraint we need to fight in designing agents. I don't buy the argument that babies tell us that much. I love Rich Sutton but I think we have to start with structure (like CNNs). Also potentially a formal consideration; the abstractions we need to learn will require an arbitrarily small.

Doina: There's one thing to say we need only data, and there's another thing to say that we always have to learn from scratch. Right now we don't incorporate the right kind of structures so that we don't have to learn from scratch. We use some ideas (CNNs, gradient descent), but we want to avoid adding too much structure, too.

Jane: One thing in Rich's essay (which I agree with about 80%) is that he said our brains/the environment are irredeemably complex. I don't agree with that. Neuroscience and cognitive science have been making great strides in understanding our brains.

Sergey: This question is different because of a methodological issue. ML is a mathematical/philosophical/scientific field. So, we're good at some things, but not at all. We've been great at making computer vision systems, language systems, and so on. In RL we've mostly been focused on solving some problems but that's been a proxy to a much grander vision that we hope to work. That's where the methodological flaw catches us. It's very easy to get improvement from bias in small problems.

Doina: I think that's true in the rest of ML too (Sergey: I didn't want to offend everyone! Dave: tongue in cheek :)). In NLP, yes, we can do some tasks but can't really do all tasks. In all of ML, we make tasks that are examples that help us improve algorithms/models, but ultimately we need to go to a more complex setting.

Matt: At the risk of engaging too much in the philosophical side of ML. We don't want to build in inductive biases that are too concentrated on one domain. In ML, we do tend to have a sense of the general domain we anticipate our agents being deployed in. We just have to make our choices about what ontology you want to buy into.

....

Q: We don't really know what tasks to train on in RL, especially in lifelong RL/multitask RL/meta RL. Any thoughts on defining the problem in a more precise way?

Doina: Simulation is obviously necessary. Think about human learning: babies have parents, who have an important role. How to build interesting and rich simulations that can be used for complex RL evaluation tasks? Well, one thing we can do is look more seriously at multi-modal data.

Sergey: Important to think about what RL would look like as a data driven field instead of a simulation driven field. If we don't we might be stuck in a regime where we don't think much about generalization and other core problems. We could think about RL tasks as being started with *data*, which is closer to the kinds of RL algorithms we might want that live in the real world. Kind of crazy to imagine an algorithm could learn a complex task *tabula rasa*.

Tejas: I agree with everything that was said. Generalization only matters when your data is limited. One way to think about that is when agents can generate their own tasks. We should think of metrics and measurements which incentivize researchers and platforms where agent can create lots of tasks, play in those tasks to learn more complex behaviors.

Tim: Question for Sergey: approaching RL in a data driven mode, how much of that needs to be done with real world data vs. simulation?

Sergey: I'll answer that with a question: what if we wanted to make a better image classifier? We do it with real data in vision because it's easier to make progress. So, in RL, it's easier too, because of the inherent complexity/diversity in the real data.

Doina: Bit of a different problem b/c we need trajectories. Lots of trajectories. Trajectories generated by people might be very different too. Might be really hard to understand/learn from whole trajectories.

Sergey: Might not be that hard. Can collect lots of self driving car data, grasping data, and so on, relatively easily.

Jane: Tend to agree with you (Sergey). One question about real data: can we make guarantees about moving beyond that laboratory data?

Sergey: That's where you need to be really careful.

Tejas: from first principles, no reason we can't make a simulator that's indistinguishable in terms of graphics and physics. Just a matter of time before we have a simulator that can replace real world data.

Sergey: Sure! But might be really slow. Why wait?

....

Q: What's the main reason to used model-based vs. model-free learning?

Karthik: Learning a model of the world can give you far more flexibility about what you can accomplish. In model free learning, you tend to just have on policy/value function, it wont generalize well/transfer across tasks. Can cover roughly 90% of things that can happen with a (good) model.

Doina: Not so sure the distinction is salient. Models can be thought of as generalized value functions, where things become much more blurry. Might end up with model-based RL that is much less efficient because learning the mapping from observation to observation is very hard. To do this, need to understand the effects of your actions vs. understand what other variables are important for policy/value function. This latter component might be much simpler than the former. Might need to rethink the nature of our models. We should build models that are small bricks.

Matt: That resonates! Fascinated by grey the distinction is between model-free and model-based. For a number of years in neuroscience this distinction was treated as very categorical. But, we've realized it's much messier. We're focused on models that can maximize value, but the situation changes when we instead focus on agents that can transmit knowledge to other agents via teaching or communication. We likely need very different models for these kinds of tasks.

Sergey: I think the two approaches are the same. They're similar, at least. RL is about making predictions (usually), but it's important to realize there's a lot of knowledge contained in some of the things agents predict like the value function (Dave: very clever dollar bill example Sergey used: imagine you wanted to predict the number of dollar bills in front of you, and maximize this value. To do this you basically need to model everything in the world.)

Tejas: Lots of domains that have nothing to do with rewards or reward maximization, like coming up with number theory or computability.

Sergey: It's not the only way to develop knowledge about the world. But if there's something that has a meaningful effect on your environment, then you'll need to predict them in order to estimate/maximize your value function.

Jane: One thing about learning these models just from interaction – in the real world, things are not necessarily disentangled in the way you might like. Dave: I missed the rest :(

....

Q: Is there a general recipe for extracting understanding from human intelligence for RL?

Matt: Meta learning (mic drop). Look at the history of AI: AI winter was partially caused by the failure of trying to encode all knowledge into our agents. We definitely want to avoid that with inserting the right structures.

Tejas: Good inductive biases are ones that will never go away. Desiderata for those are that they are truths. How do we find truths? A few scientific ways to do that: notions of agency are true, objects are true. Put these kinds of truths into the agents' head, I think we'll have agents that do the right thing.

Q: Is anything really "True"?

Tejas: Yes, otherwise it's just soup. There's no existence without it. An agent is an object with goals, and objects are true. We can try to learn these invariant truths.

Doina: With the risk of circling back: one of the lessons that we have been learning over and over again is that putting too much stuff into the agent's head can be detrimental. See: AlphaGo. Put in a bit of knowledge, let it find its own knowledge from its own data. Lots of the structure we have in deep RL are useful, but some are probably not useful.

Tejas: if you don't assume anything then what? Can we assume a model of an agent?

Sergey: Lots of things that are true and obvious. It's okay to learn that from data. Maybe it's better, because then it knows how to learn those true things to other ones. Better for my students to figure some things on their own, for instance.

Tim: Tend on the side of fewer inductive biases, and we'll tend to keep coming back to that. As we get more data, it might be easy to discovery the essential inductive biases from the data we have around. Humans/animals genetic code don't contain that many bits that are sitting in our DNA that produces our brains/bodies. Rediscovering those inductive biases might not be that hard, so we should tend to only lightly add inductive biases.

Dave: Running out of battery: missed the last question!

3 Tuesday May 7th: Main Conference

Onto day two! Today is entirely the main conference. I have a lot of meetings today so I will only be around for a few talks, sadly.

3.1 Keynote: Emily Shuckburgh on ML Conducting a Planetary Healthcheck

Remember: NOLA in 2005 post Katrina. We thought this would be a wake up call.

CO_2 in the atmosphere in 2005: 378 parts per million, in 2019; CO_2 415 parts per million. Hurricanes and cyclones in Mumbai, rising seas, wetter skies $\Longrightarrow$ devastation caused by these events is that much worse.

Note: One *million* species at risk of extinction in the next decades (from a recent study on biodiversity).

→ We are having a huge impact on our planet.

Guiding Question: How can we get a sense of the health of our planet, and turn it around? Can we use Machine Learning to make that happen?

3.1.1 Challenges for ML in Climate Science

Key questions, observations, and action items:

1. Urgently need actionable information on climate risk

   Need to understand potential risk and outcomes from

   (a) Flooding, heat waves, and other disasters.

   (b) Effects of changes in biodiversity.

   (c) Impact on supply chains (food, water, and beyond), and 4) effects on the natural world (coral reefs, forests, arctic sea ice, permafrost).

2. We have vast data-sets describing how the planet is changing.

   Includes data from satellites, robotic instruments under water, networked sensors, massive computer simulations, crowd sourcing.

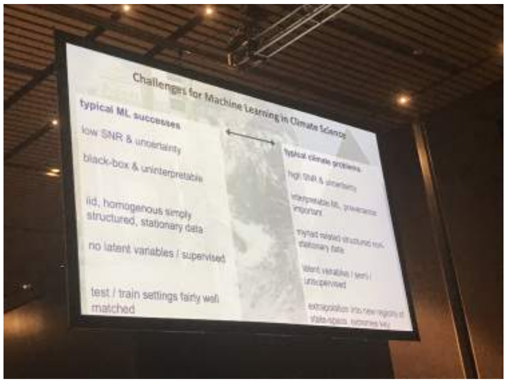
   → We have more data than we know what to do with.

3. **Main Point:** Can we employ advances in data science and machine learning to harness this data (from 2.) to help address the challenges in (1.)?

Q: In spite of the challenges (see Figure 8), what can we do?

A: Three steps to conduct a planetary health check:

1. Monitoring the planet



Challenges for Machine Learning in Climate Science

typical ML successes

low SNR & uncertainty

black-box & uninterpretable

iid, homogenous simply structured, stationary data

no latent variables / supervised

test / train settings fairly well matched

typical climate problems

high SNR & uncertainty

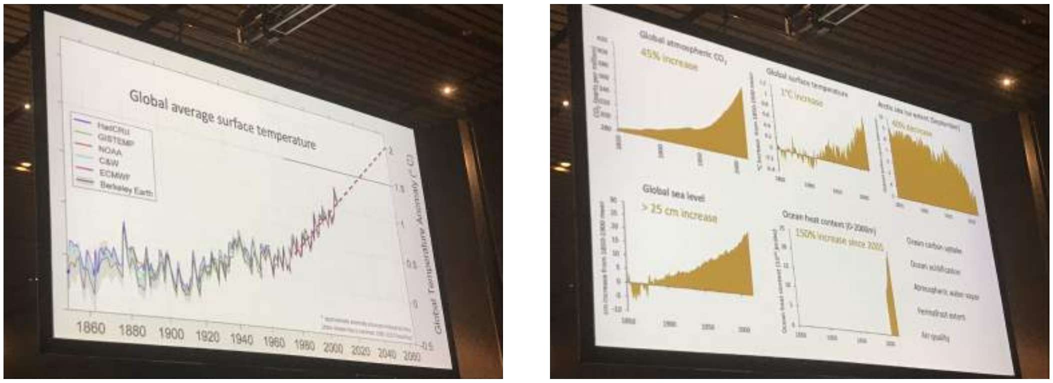
interpretable ML, provenance important

myriad related structured non-stationary data

latent variables / semi / unsupervised

extrapolation into new regions of state-space, extremes key

Figure 8: Challenges in bringing tools from ML to bear on problems in Climate Science.



(a) Surface temperature over time

(b) Other climate-relevant data over time

Figure 9: Changes in properties of the earth's health over time.

2. Treating the symptoms
3. Curing the disease

3.1.2 Step One: Monitoring The Planet

Q: How can we appropriately monitor the health of the planet? It's a huge challenge! Lots of important data is sparse, while less important (or low signal-noise ratio data) is abundant.

A: More comprehensive testing – not just temperature, but lots of other properties, too.

3.1.3 Step Two: Treating the Symptoms

Standard tools: coordinated international climate modeling project (CMIP6): $\tilde{4}0$ Petabytes. Around a million lines of code, used to run simulations of surface radiation, changes in solar radiation, and so on.

Q: What do these models do for us?

A: They make predictions about critical properties in the future, like emissions due to greenhouse gases with and without different interventions, and so on.

→ we can actually predict global average surface temperature extremely well.

Q: What will future conditions be like in the world's cities and megacities? How can we predict these things?

A: clime models can project these changes many years into the future!

But: 1) have coarse resolution, 2) have systematic biases at a local level, and 3) different clime models do better/worse at representing different aspects of the climate system.

Example: Consider a climate model making predictions about temperatures in London.

→ Sometimes, the model is systematically wrong (biased). It's too high for long periods, then too low, and so on. So how can we remedy this?

Approach: Apply probabilistic machine learning to build a new predictive model from actual observed weather data. That is, learn $f : \mathcal{X} \to \mathcal{Y}$, given lots of weather data.

Q: Can we go further? Can we extend this model to account for correlated risks and map to data on impacts?

→ Really we'd like to regulate sustainable urban drainage, thermal comfort in buildings, and address questions like how vulnerable a particular country/region is to clime disruption?

Similar approach—consider a task where:
input: time, space, climate model outputs, meteorological data **output**: future risk of specific impact occurring, with the **task:** of synthesizing and interpolating different datasets, learn mappings between different variables, may need to find novel sources of data.

3.1.4 Step Three: Cure the Disease

Key Takeaway: many opportunities for improve future projections of climate change to inform policy making. Here are a few:

1. Blend data-driven and physics based approaches

   → Can combine physics models of ice melt and machine learning models (with our large dataset) to make more accurate predictions of ice melt.

2. Develop data-based simulators of key processes

(a) (b) (c)

Figure 10: Change in glacial structure over time.

→ Given massive datasets of key processes, such as cloud formations, we can help to build more accurate models. Current climate models don't scale well, so we need to find new ways to model climate change.

3. Use ML to better understand the *physical processes* involved in key shifts, as in glacier shifts (see Figure 10.

Summary:

1. Climate change is perhaps the defining issue of our time
2. To assess risks posed to society and the natural world, we need more information and tools.
3. Vast datasets cover every aspect of the planet's health but we lack some of the *tools* to process them to generate that information.
4. **Takeaway Question:** Can we establish benchmark tasks that drive climate research forward much like ImageNet has done for vision?

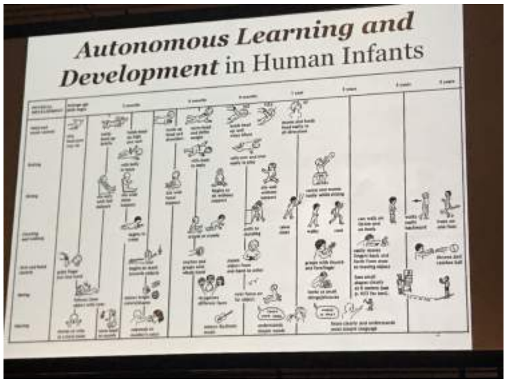
Dave: Stepping out for meetings the rest of the day, back at it tomorrow!

4 Wednesday May 8th: Main Conference

Today I should be around for more talks. The day begins with a keynote!

4.1 Keynote: Pierre-Yves Oudeyer on AI and Education

Note: Children are extraordinary learners! And typically do so without an engineer following them hand tuning every aspect of their learning algorithm and environment.



Autonomous Learning and Development in Human Infants

Figure 11: Learning and development in human infants.

Guiding Fields:

1. *Cognitive Science:* Understanding human development and learning.
2. *Robotics:* New theory for lifelong and autonomous learning.
3. *Applications* in education technology.

Example 1: study of *morphology*, body growth, and maturation in designing motor and perceptual primitives in a robot.

Example 2: consider language acquisition. Children learn new language *very* quickly.

Example 3: intrinsic motivation, play, and curiosity.

Q: How can we understand these practices, and harness them in AI tools, and build new educational tools around them?

4.1.1 Intrinsic Motivation and Curiosity

Consider *active exploration*: video of a baby playing with a variety of toys in a room over time (reminds me of the playroom domain from RL).

→ Similarly, give a baby a few toys, and a hollow cylinder suspended off the ground with a toy car inside of it. The baby over time tends to put the toy into the cylinder which knocks the car out of the tube (at which point the parent is very happy!).

→ But! When the car pops out of the tube, the baby also tends to pick up the car and put it back in the tube.

Other children experiment in very *different* ways; one kid picked up the block and hit the cylinder to make noises, and seemed very pleased by the noises. This was considered a "failure" in the study, but was pretty sophisticated exploration!

Note: Theories of intrinsic motivation, curiosity, and active learning drive to reduce uncertainty, experience novelty, surprise, or challenge. See Berlyne [2] and Berlyne [3].

Perspective: The child is a sense making organism: explore to make good predictive models of the world and control it!

Q: Based on this perspective, what kind of modeling/algorithms are needed in order to explain these behaviors?

A: We use robotic playgrounds – place robots in a playroom like environment, and encourage them to play to learn object models and affordances. Also place another robot in the playroom that gives feedback in the form of contingent reactions (like making sounds or movements based on the behavior of the learning robot). This in turn lets the guide robot play the role of a parent encouraging/discouraging the baby.

Essential ingredients in these robots:

- Dynamic movement primitives
- Object-based perceptual primitives (like infants, build on prior perceptual learning)
- Self supervised learning forward/inverse models with hindsight learning
- Curiosity-driven, self-motivated play and exploration.

4.1.2 The Learning Progress Hypothesis

Q: What is an *interesting* learning experiment for a robot/baby to conduct (to learn)?

Lots of answers in the literature: high predictability, high novelty, high uncertainty, knowledge gap, novelty, challenge, surprise, free energy, and so on.

This Work: The Learning Progress Hypothesis [31]:

Definition 9 (Learning Progress Hypothesis): *The "interestingness" of an experiment is directly proportional to empirical learning progress (absolute value of derivative of the errors)*

→ Few assumptions on underlying learning machinery and on match between biases and real world.

Framework: suppose we have some robots with motion primitives. Takes some sequence of actions to yield a trajectory:

$$\tau = (s_t, a_t, s_{t+1}, \ldots).$$

From this trajectory, the robot should learn, assuming some behavioral abstraction ϕ:

1. Forward model: $F_i : s, \theta \to \phi_i$, with θ the parameters of the behavioral policy, $\pi_t heta$.
2. Inverse model: $I_i : s, \phi_i \to \arg\min_\theta ||\phi_i - F_i(s, \theta)||$

We can use these models to measure learning progress:

1. Measure changes in errors of the *forward model*.

   → Help conduct prediction experiments where the agent samples parameters θ based on this learning progress.

2. Measure changes in errors (derivative) of the inverse model/goal achievement (called "competence progress").

   → Can yield goal achievement experiments in which the agent 1) samples goals where they *expect* high competence progress, and 2) use the model to infer a θ and roll it out.[3]

Algorithm used for generating the automated curriculum learning: hierarchical multi-armed bandits. The idea is to split a space into sub-regions, where an agent monitors the errors of each sub-region. Use these errors to measure the learning progress over time. Then, in the bandit setting, can explore based on the ratio of these errors over time.

→ Note: This algorithm will be a core of the curriculum learning for the rest of the experiments described.

→ Example: explore omni-directional locomotion. Look at diversity (in terms of spread of states reached in some space) of outcomes by different exploration policies on a robot. Finding: curiosity-driven exploration is less-efficient than goal exploration.

Finding: Two kinds of curiosity-driven exploration 1) of the forward model, and 2) goal exploration (using the inverse model). Turns out the second kind is more efficient to learn a diverse set of skills/controllable effects.[4]

Example: curiosity-driven discovery of tool use. Videos of a robot playing with different tools (a robot gripper learns to interact with a joystick that moves a separate arm with a cup as its

[3] Many thanks to Pierre-Yves Oudeyer for the clarification on this point!

[4] Thanks to Pierre-Yves Oudeyer for the helpful clarification!

end-effector).

→ Point: focus on playing with and manipulating objects in the world. The gripper robot learns to manipulate the joysticks, which moves the robot that can pickup the ball. Torso eventually learns to make a light, play a sound, and hide the ball in the cup.

Project: "MUGL: exploring learned modular goal spaces" [25]. Main idea is to extend these exploration techniques to high dimensional input (the robot examples above used a feature vector, not images).

→ MUGL can be used to discovery independently controllable features (learn to control a ball, and so on).

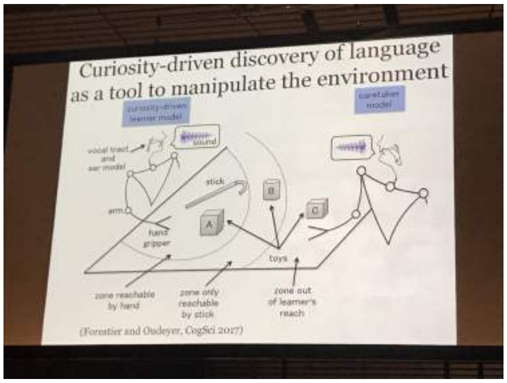
4.1.3 Models of Child Development Data

Experiment: modeling vocal development. Use exact same algorithms from before.

→ Goal: make experiments for the infant using the learning progress idea from before.

Finding: Some self-organization of developmental structure in infants. First vocal track is learned (unarticulated sounds) and then learns articulated sounds.

→ Observe: regularities that tend to occur at the same time across different individuals, but some things change dramatically. Interactions between learning system and body morphology is stochastic, contingency in exploration, surprising that many things remain constant.


Curiosity-driven discovery of language as a tool to manipulate the environment

curiosity-driven learner model
caretaker model
sound
vocal tract and ear model
stick
arm
hand gripper
A
B
C
toys
zone reachable by hand
zone only reachable by stick
zone out of learner's reach

(Forestier and Oudeyer, CogSci 2017)

Figure 12: Curiosity-driven discovery of language

The "Ergo-Robots" (with Mikhail Gromov and David Lynch, I think?) Dave: Lynch! :o. Surreal video of robots learning to speak and interact with their environment and companions, see a sample video here: `https://www.youtube.com/watch?v=J0gM5i091JQ`. Robots learn to use language in a meaningful way through exploration.

→ Use similar ideas to come up with a realistic models of learning to use a vocal track. See: *Self-Organization in the Evolution of Speech* [32].

Finding: The distributions of vowels we find in the world languages matches those of the systems that emerge in these curiosity-driven learning systems. This might explain some regularities of language structure.

Q: How is spontaneous exploration structured during free play?

A: Experiment! Let subjects play a bunch of games/tasks, with no guidelines. Just do whatever you want (play games like guitar hero, free to pick any level/song).

→ People tend to focus on levels of intermediate complexity; exploration follows a controlled growth in complexity, actively controlled by individuals' predictive models.

4.1.4 Applications in Educational Technologies

Goal: Develop technologies for fostering efficient learning and intrinsic motivation.

→ Project: KidLearn – allows personalization of intelligent tutoring systems, based on experiments with $>$ 1000 children in 30+ schools.

Principle: graph (usually a DAG) defines difficulty of task/exercise type. This allows the system to sample exercises in some sequence (but still give the kids some choice among nodes in the graph).

Main study:

- Examine learning impact based on these interventions.
- Compare to typical pedagogical expert (vs. their system).
- Find that students tend to achieve higher success rate with certain variations of the algorithm.

Takeaways: Fundamental role of spontanous developmental exploration, can be harnessed to develop human-like robots and empower the learning process.

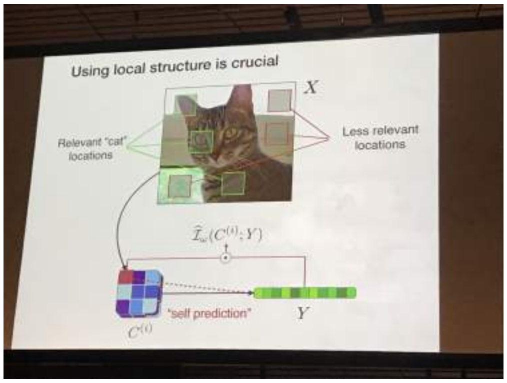
4.2 Contributed Talks

Next up some contributed talks.

4.2.1 Devon Hjelm on Deep InfoMax [17]

Broad Goal: Learn unsupervised image representations.

Example: video of a dog catching a snowball. What annotations make sense? ("Cute dog!", "Good boy!").



Using local structure is crucial

Figure 13: Extracting local feature maps via local mutual info estimation and maximization

→ Not clear these are the right/useful annotations.

Point: Don't always want supervised learning of representations. Annotations rarely tell the whole story, real world doesn't come with labels, and really want to find the underlying structure (annotations might not enhance this part).

Preliminaries:

- Encoder: $E_\psi : \mathcal{X} \rightarrow Y$, with Y a representation.
- Mutual info; $I(X;Y) = D_{KL}(P(X,Y) \mid\mid P(x)p(y)$

→ Introduce a mutual information estimator: encode an image into a representation. Take pairs of representations from images that aren't associated with each other, and treat these as *negative* samples,.

Approach:

1. Encode input X input Y via E_ψ.
2. Use output to estimate $\hat{I}(X;Y)$, and maximize this estimate.
3. Just doing this alone isn't quite enough.

   Intuition: you might not pick up on the relevant locations of an image. Consider a picture of a cat, the background isn't as crucial as the information in the front.
4. So: instead of maximizing the mutual info globally, instead maximize it *locally.* Perform this estimation/maximization across all locations simultaneously. See Figure 13
5. This yields local feature vectors for different regions of the image, which can then be stitched together into a global feature vector.

Evaluation: depends heavily on downstream task. Can measure mutual info, down stream performance on classification/regression tasks, and so on.

→ Deep Info Max performs very well when the learned representation is used in downstream tasks like classification.

Other tasks investigated: prior matching, coordinate prediction, relative coordinate prediction

Dave: Off to more meetings.

4.3 Keynote: Zeynep Tufekci on Dangers if ML Works

Grew up wanting to be a physicist! But: eventually, many physicist encounter the *nuclear* problem: massive ethical issues facing the advancement of a scientific field.

→ At CERN recently: giant projects! 700 people on the Higgs Boson paper. They were concerned about how to divide up the Nobel prize, meanwhile in CS we're worrying about the impact our tools have on society, security, labor, climate, social infrastructure, and beyond. So: lots of ethical issues in CS, too!

Back in Turkey: no internet, limited access to culture/TV from the outside world (grew up watching little house in the prairie). But, got internet connection through working at IBM.

→ Wonderful to have unimpeded access to lots of information and connections to people! Lots of optimism about the power for this tool to do good in the world

This Talk: big dangers that we're sleepwalking into. Going to end up in a scary and unhappy place, we will end up having built and contributed to part of it.

- Seven years ago: the "cat" paper, that learned to recognize cats on YouTube. Was hard to even imagine the current state of the field back in 2012.
- What we need to worry about: what if it works? What are we really introducing into the world?

Themes:

1. You are part of this world, and your tools will be too.
2. These tools won't be used the way you think they will be.
3. Alternative paths are possible.

4.3.1 Examples Of Ways Things Can Go Wrong

Example 1: social movements in the public sphere. Specifically, the Facebook news feed:

- Optimized to keep people engaged on the website

- Recall: Ferguson, MZ, shooting of an African American teen by a police officer. Similarly: in a McDonalds, some customers were thrown into a van without any chance to say anything (by some police officers).

  → Huge discussions about these events on Twitter, but *not* Facebook.

- Point: Friends *were* talking about it, but these issues weren't "engagement friendly" for Facebook (it was trying to populate news feed with things that would be liked/shared).

  → At the time, the ALS ice bucket challenge was going as well. Very likeable/shareworthy, so that dominated most Facebook news feed.

  → Chilling moment: icebucket challenge dominated the media/Facebook as a result.

Difference for social movements isn't whether you hold a protest. The thing the movements manage to do to be most effective is get attention.

→ But our algorithms are optimized to keep people engaged, so it tends to not spread certain kinds of information. A new form of censorship.

Classic finding: People that tend to be into one conspiracy theory tend to be into others (from social science).

→ Instagram would recommend similar conspiracy theories, not because designers recommend them, but just because the algorithm learned this social science fact (someone likes a "faked moon landing" post will also be presented with tons of other conspiracy theories).

in 2016: these phenomena were *widespread.* Watching one certain kind of video would immediately bias future videos (if you watch a jogging video, it then leads you to hardcore marathon training, if you watch a vegetarian video, it then leads you to vegan videos).

Point: YouTube algorithms tends to *lead individuals down more polarizing and extreme routes.* Not by hand-design, but it does it *to increase engagement.*

→ Hate speech, extreme views, and so on, are very "shareworthy" and capable of improving engagement.

4.3.2 ML and its Challenges

Some core challenges of implementing ML systems in the world:

1. Bias
2. Interpretability
3. Potency-at-scale Restructuring power
4. Surveillance and Privacy
5. Transition and Upheaval

Example: suppose you're hiring, and suppose you use ML to hire.

(rhetorical) Q: We have methods for predicting depression rates; what if your ML system in using this information for hiring outcomes?

Problem: You won't even know that your system is using this information!

Similar problem: can use a "mirror population" system (identify similar groups) to target individuals that are fear-prone with well designed ads to encourage voting for authoritarians? (again coming from findings in social science).

→ The way these systems are built *incentivizes* surveillance, since data about individuals/groups is always important. Data *will* get out, and will be used.

Path Forward: build things that can do what we want to do, but not let it control us and be used on things we don't want it to be.

**We are in a very particular historic moment: all the companies here are trying to recruit you. If these companies that are trying to recruit you, but can't get you to do the things they want you to do, but you *insist* on using privacy-preserving methods.

Takeaway: Lots of good the talent in this room can do.

Some thoughts:

1. Google gave away money for ML projects. One of them was on detecting societal ideation, someone at risk, and do intervention.

   Seems like a great

2. What will happen: university will expel kids with mental health issues. They don't want that happening in their dorms.

3. Look at the database of people killed by police officers. Many people are in a mental health crisis.

   → Suicide detection program can be used in a very bad way (like laws that prevent at risk individuals from basic needs/goods).

4. But! Imagine a privacy preserving system designed to help individuals instead.

First earth day: pictures were blurry because of smog. Now that's not the case! So, a final note: genuinely consider unionizing.

Union: you get a huge number of legal protections. You get some freedom of speech. Organize your own industry ML group.

No one here is trying to explicitly go down this dark path. So many good people in this community! But the business model plus the world leads us to the wrong kinds of systems.

Final thoughts:

1. Organize
2. Build alternatives
3. Insist on having a voice

*What we want, when someone wants to have a phone/app, we're just doing a bad job of what China is doing (in terms of monitoring/surveillance) for the purpose of selling more stuff.

→ We need a real alternative. We need the conveniences and empowerment, we need an option that respects us, that gives us early warning and intervention but not at the expensive of subjecting us to mass surveillance.

Final Example: Before Diffie-Helman (public key crypto), we didn't have a way of sending a message without exchanging secret keys.

- They thought this was a terrible problem. Needed to give people a way to communicate and exchange info without having to exchange keys beforehand.
- All of public key crytography comes from that. Really important and individual-empowering tool that has dramatically reshaped the state of the world.
- So: stock options are cool, but we should be thinking about developing tools like this for the next generation of technology.

4.3.3 Q & A

Q: Could you comment on AI in the military?

A: War in 2019: any form of it is going to be terrible. Anything that makes it more convenient will make it even worse. Governments should be accountable to us.

Q: Feels a bit US centric – any thoughts on the broader international perspective?

A: Remember the paper that can detect (claims to) detect sexual orientation. Ugandan government has outlawed homosexuality, so we need to keep these things in mind. If we have better health care some of these issues will be less damaging. Fixing the politics makes some of these dangers less potent. Important for silicon valley not to be in this bubble. Every time I go to SanFran now, every other table around me is talking about when stock options vest. Even if you're making great money but health care isn't there and your technology is used to keep people from getting hired, it's problematic.

Q: Question about the "solutions" part, and refusing to build unethical software. Often we build small parts of a big thing, how do you predict what's going to be harmful?

A: First, that's why we should organize and unionize. Second, refusal is powerful but will only last so long. Real path forward is the alternative path: can't get out of oil without solar/wind energy.

Surveillance and data economy is our oil economy, and it's cheap. But we can develop the solar equivalent here, and insist on these alternatives that we can embrace. Third, you all might be on the product side: I encourage you to find people on your security team and talk to them. They see the worst. They can warn you how these things might be used in the worst possible way.

Dave: They did the last three questions together
Q: Lots of surveillance in the US, and strong culture against whistleblowers, too. Should people at ICLR take a bigger stand?

A: Yes absolutely speak out against that. Encourage and support whistleblowers in your own company.

Q: What are the real problems, and what specific things should we be avoiding? We've seen engineers stand up to people working on the army. We've talked about companies that sell ads at any cost.

A: If the data exists, it will be sold and used. Great challenge: we need to figure out how to do some of this without surveillance. That includes, expiration dates on data, operating on encrypted data (insights on aggregate, not individualized). Key thing is to kill the surveillance economy. We can't kill phones, since those empower us, but we need an alternative.
Q: Any comments on health care solutions?

A: Lots of things! Skin care diagnosis and other tools that are empowering. Need to figure out a new full stack trusted method that doesn't feed into the surveillance economy. No way we can collect all this data and work on this data and not have the powerful corporations and governments come for it. We need to stop the surveillance economy.

4.4 Debate led by Leslie Kaelbling

The debaters are: Josh Tenenbaum (JT), Doina Precup (DP), Suchi Sara (SS), Jeff Clun (JC), with Leslie's statementsas LPK.

LPK: We are interested in questions from the audience! The topic is on structure in learning. Each panelist will do some intros:

DP: I drew the short straw. I'm going to argue that a lot of what we need can and should be learned from data rather than come from priors. Especially in the context of generally purpose AI, rather than special purpose applications. In a special purpose application we should definitely use structure/priors, but in making general AI, we want our systems to learn from data. See: AlphaGo. First we used the specialized human version with some expert knowledge, but then the approach that ended up dominating was the one that used entirely data.

JT: Other extreme! I want to emphasize what you can build in. I, like Doina, am interested in general AI. I'm not against learning, but I'm really struck by how much systems like AlphaGo *don't* do. We have a general toolkit for building a specific kind of intelligence, but not general intelligence. Transferring to a different board size, for instance, would require retraining from nearly scratch.

I'm interested in taking inspiration from the ways humans come into knowledge, like learning from the stage of a baby. In the universe the only case we have of learning from an early/scratch stage are human infants. We've learned so much, a bit contrary to some of the (bitter) lessons that some people have been drawing. The findings in cognitive science we find tell a different story: lots of inductive biases in human children. Also: the gazelle, learns to walk in the savanna very quickly or it will be eaten. Or think about a bird: the first time it falls out of a nest it has to fly. Human babies don't start off being able to walk, but before they can do those things, they develop objects, intuitive physics, schemes for defining goals, some notion of space, and so on. Exciting opportunity to take those kinds of core systems and learn how to devise and go beyond them. I would like to think about how we can take the right kind of machinery that include what we know how to do in RL and deep learning, but also what we know about symbolic reasoning and classical methods, so they know how to live in a world.

JC: This debate is usually framed as two extremes: 1) build in the right stuff, and 2) learn everything from scratch. I think there's a third alternative here, which is what I call AI generating algorithms: algorithms that can search for an AI agent that on the inner loop is very sample efficient (which come from the outer loop doing the right thing). It's a nice marriage between the two things. When you have a new problem you don't learn from scratch, you deploy this new agent. We know that can work (existence proof: Earth, this happened!). The algorithm is evolution, the inner loop is the human brain. This research direction isn't to say that the outer loop has to be evolution, could be something else like meta learning. If we want to make progress here: three pillars:

1. Meta learn the architectures
2. Meta learn the learning algorithms
3. Automate effective learning environments (scaffolding)

Clear trend in ML that we're aware of: hand designed systems that work okay will be surpassed by approaches that are wholly data driven/rely on big compute. If you believe in this trend, you should also believe that it can apply to the discovery of this machinery itself. So: what are the structures that allow for sample efficient general purpose learning? This may be the fastest path for us to pursue these big goals.

SS: I'm going to oversimplify a bit. Each of the three other panelists proposed a solution path, so I want to highlight some observations, first:

1. Observation 1: Josh and Jeff suggested that if we want to build human-like intelligence, and we know we can get there during learning. Proof for this is evolution. Evolution is very slow. In the process of getting to us lots of calamities. So, a big issue: we can't afford to make our civilization to go extinct or to make societal level mistakes. What does it mean for us to come up with the right paradigm since we can't afford to make mistakes?

2. Observation 2; What are the levers we have in ML? Algorithms learn by interacting or from past data. So the question is: can we conclude that if we have data and interactions alone, we can learn anything we want? Unclear!

3. Observation 3: As a field we focus very hard on incrementing on a solution set. I'm curious if we have answers to these kinds of questions: what can we learn easily, starting from a superintelligent being. Do we have a taxonomy for what's easy to learn what's hard to learn? Can we define where the gaps are? Might be good to define the strategy here!

So, do we really want to be as slow as evolution or take into account what we know? Definitely the latter

....

LPK: Really important for us to each define our own objectives. We each have different big and small scale objectives. Big scale: understand math of learning, biology of the brain, making things work in 10 years or 100 years. So one reason to make this not a knockdown debate is that probably no answer will be true for every objective. So can you each say something more about your objective?

DP: I have multiple objectives. Want to understand how to build general AI agents. Some of the things we do now don't align with that goal: we often train from scratch, but don't necessarily need to do that! We could load a vision system in and learn downstream aspects. Would be great to emphasize this kind of continual learning a bit more. Other chunk of my time I think about medical applications: it's a space where we don't have a lot of data and we can't always intervene because of ethical considerations. Here we often use structure/priors. Conceptual thing: what are the principles of intelligence? Focusing on learning from data can get us to that. People are wonderful but there are other intelligent things in nature too. Working on these algorithmic questions can help us understand much of nature.

SS: To me, the biggest challenge is how do we proceed when our objective is broken? (see previous keynote!) Lots of research is asking the right question. Who is going to define the question? How should we think about defining the right objective?

DP: That's true! We can have different objectives and that's okay.

JT: I'll say quickly: we're all pretty broad minded people but we have our strong deep views about things. I love the evolutionary view. We look at the learning algorithms we've seen, Evolution is the master algorithm (Pedro Domingos). Gazelles/animals are deeply fascinating. Also cultural evolution, especially once you have language, opens up whole new possibilities. The success of AI tools has in large part occurred due in part to the cultural evolution that lets us build collective knowledge and share goals and ideas.

JC: I'm interested in how to build intelligence piece-by-piece. But also interested in building a process that can build intelligent systems. Also fascinating to think about the set/space of possible intelligent entities. In part we're motivated by what looks like us, but we might miss out on huge regions of what intelligent/sentient systems look like. Also motivated by applications in health care, and beyond. Great opportunity for all of us!

LPK: Okay now we'll go to audience questions. This one is anonymous: "What kind of results would cause you to change your opinion on how much structure vs learning is necessary?" — How much structure is necessary is the wrong question? You can contextualize this with respect to a particular objective and get a really fascinating question. Hard to prove a negative. We're in the midst of a methodological crisis because we're all putting points in a space. Colleagues?

JT: I know one case study I find interesting about intuitive physics: being able to predict/imagine/plan to causally intervene on objects. All about predicting "what happens if I hit this thing with that thing?". We built a system that didn't do any learning, just did probabilistic inference using a game engine. Other people around the same time tried tackling same problems but in an end-to-end learning method in some impressive ways. Still required huge amount of data and didn't seem to generalize to well. Others have been trying different systems. We've had some workshops on this topic, and reached broader argument: not an argument for necessity, but an empirical result. Building in some models of objects that might interact via contact/having close spatial locality is extremely powerful. We can't prove they're necessary but they're massively effective. Doesn't change any one person's mind but it's a case where we've learned some important empirical lessons that turn out to be valid.

DP: Can be very valuable to build these things in, but I'm not convinced that we can't just learn these things. The main complaint is usually sample complexity: if we had the right kind of approach for the learning algorithm it could learn these things. Typical example is causality. If we had the right learning algorithm we could learn a causal model not a predictive model. Another short thing about methodology, also very important for us as a field. Difficult to make progress when we care a lot about numbers without also understanding what our systems do. So yes it's great when our algorithms do better, but we need to also understand why. I want to argue that we need to probe our systems not just measure our systems quantitatively/qualitatively, but really probe our systems via hypothesis testing.

JC: I agree with Doina! Just wanted to ask Josh: might give us a big lift by building these things in, but perhaps we can learn even better lifts? How do you see things like HOG and SIFT where we eventually learned something better?

JT: Yeah, HOG and SIFT is a very instructive example. We see things in intuitive physics with a similar story: right now the best models we have are physics engines (for playing video games, for instance). But we know there's lots of things they don't capture. Recent work starting to learn physics simulators that can improve on these classical notions of physics simulators. We don't know where that story will end. If we look at vision, not just HOG and SIFT, you see the community over a few generations of research, we see the same motifs/ideas repeated (non-linearity, hierarchy, etc). Many versions of the same idea. HOG and SIFT were one idea, AlexNet was another. We'll see similar ideas recur and build and call back to the same motifs.

SS: Josh, question for you: do you think the reason we need more than learning is that you think these systems cant be learned from scratch? Can we learn physics just by learning?

JT: That's what happened on Earth via evolution. Compute for simulating evolution is huge and we're just barely start to making progress. So the current route of modern deep RL is not necessarily going to deliver on this promise to learn an entire physical model. Human children are proof that some other route works.

JC: I think lots of routes can get to general AI, but it's partially a question of which is the fastest route. It's an open question as to whether the manual path will get there faster than the meta learning or evolutionary approach.

SS: Except you missed this along the way: what about everything that happens along the way?

JC: Fascinating question! General question of whether we should be building a general AI? Also, what should we be doing along the way to building that? Very real considerations. What's the fastest way to get there: I say the meta-learning approach above. Regarding ethics? Lots of open questions, too.

LPK: One thing Jeff didn't mention was the physical infrastructure required to make this work. Robots need to interact with the world unless we rely on a perfect simulator. I'm not sure we can get that taking off that we might suggest.

....

LPK: Next question "Symbol manipulation has no place in deep learning. Change my mind" (Yann LeCun). Symbol means something different to embody; I'd argue that embeddings are roughly symbol manipulations, so maybe you've been doing it and you don't actually know.

JT: Certainly not my place to define what deep learning is. One reason it's interesting is that deep learning can mean a lot of things; it could mean and already sort of does mean doing SGD and symbol manipulation. Nice talk tomorrow on this. Whole community trying to explore interesting ways for symbols to live inside of systems that do deep representation learning. Multiple ways to learn a representation, might include symbols or combinations of these. We'll see systems that do things better than without these different techniques.

DP: But will we see systems that do this from scratch?

JT: Having a basic path for compositionality that gives us meaning/function that wasn't there from the beginning.

DP: Meaning as in people assign meaning? Or meaning as in what the machine defines for itself?

JT: Whatever you mean! ... If we want to build systems that we trust that we think do have the right kind of meaning, it's one reason to build AI that have these kinds of structure/symbols.

....

LPK: Q from Yoshua Bengio: "We of course need priors but to obtain the most general AI, we'd like the least amount of priors which buy us the most towards AI-tasks, don't we?"

JC: Yes!

SS: Also yes, New question for the audience: everything can be learned? (some hands) not everything can be learned (more hands) – some examples from audience poll of things that can't be learned: causality, halting problem, false things.

Dave: Need to run out for a meeting! Fantastic debate though.

5 Thursday May 9th: Main Conference

Last day! I'm flying out this afternoon so I will only make a few sessions today.

5.1 Contributed Talks

Now some contributed talks.

5.1.1 Felix Wiu on Pay Less Attention with Sequence Models [36]

Observation: Sequence models are almost everywhere in NLP (have become the CNN↔ vision).

This work:

1. Q1: Is self attention needed for good performance?
2. Q2: Can we do well on a range of NLP tasks with limited context?

Different models perform very differently on Neural machine Translation (Transformer achieve BLEU of 28, SliceNet of 25, phrase-based of 22).

→ Large performance gap of self-attention and convolutional models.

Background: three ways to encode a sequence

- RNN: recurrent neural net. can. $h_t = f(x-t, h_{t-1})$, with x_i the input at time i, h_i the hidden state at time i.
- CNN: convolutional neural net.

  $h_t = f(x_{t-k}, \ldots, x_{t+k}) \rightarrow$ look at a limited window.
- Self-attention models: Compute pairwise similarity between words and aggregate them.

  $h_t = \sum_{i,j} a_{i,j}$.

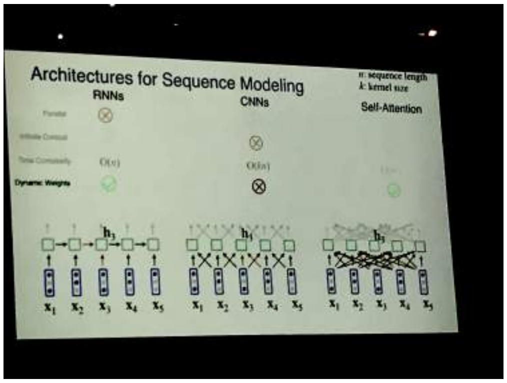
Some pros and cons of each! RNNs can't be parallelized, while CNNs and self-attention, time complexity is higher for self-attention, and so on (see Figure **??** for full comparisons).

Approach: *dynamic* convolution that addresses the main disadvantage of CNNs (lack of dynamic weighting).

But, some challenges to dynamic convolution: too many parameters to optimize!

→ Response: turn to lightweight convolution, which reduces the number of parameters.

Experiments: Explore the trade-off made between *inference speed* measured by sentences per second) vs. *BLEU score*, which is a way to measure the quality of output translations.



Architectures for Sequence Modeling

n: sequence length
k: kernel size

Figure 14: Pros and Cons of CNNs, RNNs, and self-attention for sequence modeling.

→ Main finding: dynamic convolution achieves same BLEU score as self-attention, but with a 20% speed up in inference time.

Conclusion:

1. Local information is sufficient for several NLP tasks.
2. Introduced dynamic convolution: context-specific kernels.
3. Lightweight convolution: fewer convolution weights still work well.

5.1.2 Jiyauan Mao on Neural-Symbolic Context Learner [29]

Focus: Visual concept reasoning.

→ Given an input image (of some objects), people can quickly recognize the objects, texture, surface, and so on.

Visual Question Answering: given an image and a question "What's the shape of the red object?", output an answer to the question.

→ Also, may want to do image captioning: "there is a green cube behind a red sphere", or instance retrieval (a bounding box on a particular object).

Prior Approaches: End-to-end approaches for solving these three problems. Two things to learn: 1) concepts (colors, shapes), and 2) reasoning (counts).

→ Downside to end-to-end: concept learning and reasoning are entangled. Not obvious how to transfer.

This Approach: Incorporate concepts in visual reasoning. Prior methods rely on explicit concept annotation.

The idea:

- Joint learning of concepts and *semantic parsing*.
- Given a scene parser, and a semantic parser, learn a program that understands the concepts while parsing both objects.

Example: given an image of a red sphere and green cube, first perform object detection/feature extraction to get a representation. At the same time, do semantic parsing on the text, to output a parse program that predicts the output of the question. The full overview is given in Figure 15

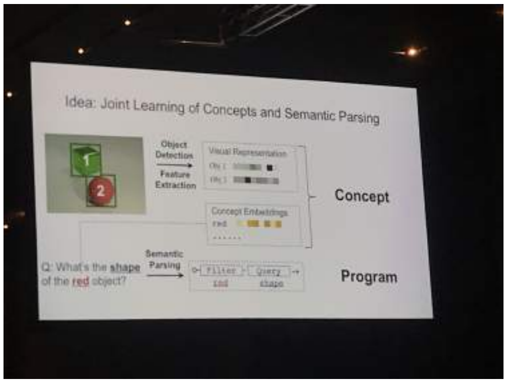

Idea: Joint Learning of Concepts and Semantic Parsing

Figure 15: Overview of the approach for joint semantic and scene parsing.

Two main methods:

- Learn a program for understanding concepts
- Learn a concepts that can help facilitate parsing new sentences

Experiments: This approach yields several advantages

- State of the art performance on the "CLEVR" data set for visual question answering.
- Extensions to natural images and natural sentences as in the VQS dataset: "what color is the fire hydrant?" given a natural seeming image of a fire hydrant (correctly guesses "yellow").
- Model also supports composition of low level concepts into high level concepts, and bounding box detection.

Limitations and Future Directions:

- Consider example of a person with an umbrella hat on, and the question "what purpose does the thing on this person's head serve"? proves extremely challenging!
- Recognition of in-the-wild images and beyond (like goals).
- Interpretation of noisy natural language
- Concept learning in a more sample efficient way.

Conclusions:

- New model: NSCL learns visual concepts from language with no annotation
- Advantages of new model: high accuracy and data efficiency, transfer concepts to other tasks.
- Principles: explicit visual grounding of concepts with neuro-symbolic reasoning.

5.1.3 Xiang Li on Smoothing Geometry of Box Embeddings [27]

Point: Learning representations is crucial in NLP! These representations are usually vectors like word2vec or BERT.

→ These vectors define semantic similarity in space (closer together words have similar meaning/use).

But, consider: Rabbit/mammal. They're close to each other in space, but don't capture the full complexity of their relationship rabbit $\subset$ mammal).

→ One idea: Gaussian representation of classes like "mammal". Advantages: 1) region, 2) asymmetry, 3) disjointness; but, one downside: not closed under intersection. Recent work extends this to a probabilistic model that gives up disjointness to achieve closure under intersection.

Their Approach: An extension of these probabilistic models using a *box representation* to account for joint concepts, thereby achieving all four of the desired properties (region, asymmetry, etc.). Box representation seen in Figure 16.

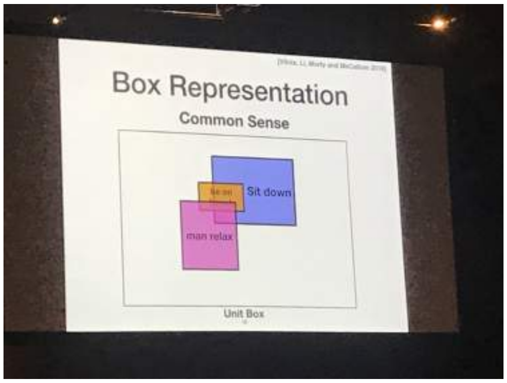


Figure 16: Idea behind the new probabilistic box representation

Learning problem; boxes represent probability mass, try to do distribution matching over concepts. Initialize random concepts ($Pr(deer), Pr(deer \mid mammal)$.

Experiments: 1) Matrix factorization in MovieLens, 2) Classification on Imbalanced Wordnet

1. MovieLens Marketbase; Movie × Movie matrix: $p(lionking \mid aladdin) = 0.73$), 286 million pairs.

   → Regression task: train/dev/test, yields a matrix factorization problem (determine which movies people will like).

   → Forrest Gump ends up being a large box, indicating that everyone likes it!

2. Imbalanced WordNet: show the models learning ability for sparse, disjoint data.

→ Binary classification task: achieve SOTA, even with sparse/disjoint data.

5.1.4 Best Paper Award Talk: Yiqang Shen on Ordered Neurons [34]

Assumption: Language has a latent tree-like structure

→ This work: focus on *constituency tree.*

Q: Why?

A1: Hierarchical representations with increasing levels of abstraction can be captured by these trees!

A2: Compositional effects of language, and long term dependency problem can be handled by these trees.

Main Question: Can we provide a new inductive bias based on this tree structure to achieve a higher down stream task performance?

Two types of models for answering this in the past:

1. Recurrent models (SPINN, RL-SPINN, RNN)
2. Recursive models (RvNN, ST-Gumbel, DIORA)

→ For most prior works: tree-structure given by external parser, or try to make hard decisions about how to design it.

This Work: Integrate a tree structure directly into an RNN.

→ Tree-structure is defined by: when a larger constituent ends, all nested smaller constituent also ends.

Effect: This yields an inductive bias of "ordered neurons", when a high ranking neuron is erased, all lower rankings neurons should also be erased.

To model this structure, introduce a new forget gate called the *cumax*:

$$cumax(x) = cumsum(softmax(x)). \tag{4}$$

Master gates for RNN:

- Master forget gate: $\tilde{f_t} = cumax(W_f x_t + \ldots)$
- Master input gate: $\tilde{i_t} = 1 - cumax(W_f x_t + \ldots)$

Experiments:

1. Language Modeling: PTB dataset to do next-word prediction. Achieve near state of the art.
2. Unsupervised Constituency Parsing: Penn TreeBank data set on language modeling task.

3. Targeted Syntactic Evaluation: Marvin and Linzen dataset on a language modeling task (given a pair of similar sentences, one ungrammatical, one grammatical, see how the model performs). ON-LSTM is able to pick up on the long-term dependencies.

Summary:

- Proposed new Ordered Neuron inductive bias:
  - High ranking neurons sotre long term info
  - Low ranking neurons store short term info
- New activation: $cumax()$ and ON-LSTM
- Inducted structure aligns with human annotated structure
- Stronger performance on a lot of experiments.

Dave: And that's a wrap! Just a poster session left and then I'm off to the airport.

Edits

I've received some messages pointing out typos:

- Thanks to Anca-Nicoleta Ciubotaru for catching typos.
- Thanks to Pierre-Yves Oudeyer for helpful clarifications.

References

[1] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *AAAI*, pages 1726–1734, 2017.

[2] Daniel E Berlyne. Conflict, arousal, and curiosity. 1960.

[3] Daniel E Berlyne. Curiosity and learning. *Motivation and emotion*, 2(2):97–175, 1978.

[4] Mathew Botvinick, Sam Ritter, Jane X Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. Reinforcement learning, fast and slow. *Trends in cognitive sciences*, 2019.

[5] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, pages 4754–4765, 2018.

[6] Nathaniel D Daw and Peter Dayan. The algorithmic anatomy of model-based evaluation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1655):20130478, 2014.

[7] Philip Dawid. On individual risk. *Synthese*, 194(9):3445–3474, 2017.

[8] Nat Dilokthanakul, Christos Kaplanis, Nick Pawlowski, and Murray Shanahan. Feature control as intrinsic motivation for hierarchical reinforcement learning. *IEEE transactions on neural networks and learning systems*, 2019.

[9] Carlos Diuk, Andre Cohen, and Michael L Littman. An object-oriented representation for efficient reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 240–247. ACM, 2008.

[10] Harrison Edwards and Amos Storkey. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*, 2015.

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[12] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018.

[13] Jean Harb, Pierre-Luc Bacon, Martin Klissarov, and Doina Precup. When waiting is not an option: Learning options with a deliberation cost. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[14] Anna Harutyunyan, Peter Vrancx, Pierre-Luc Bacon, Doina Precup, and Ann Nowe. Learning with options that terminate off-policy. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[15] Anna Harutyunyan, Will Dabney, Diana Borsa, Nicolas Heess, Remi Munos, and Doina Precup. The termination critic. *arXiv preprint arXiv:1902.09996*, 2019.

[16] Úrsula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1944–1953, 2018.

[17] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[18] Matthew Johnson, Katja Hofmann, Tim Hutton, and David Bignell. The malmo platform for artificial intelligence experimentation. In *IJCAI*, pages 4246–4247, 2016.

[19] John T Jost and Mahzarin R Banaji. The role of stereotyping in system-justification and the production of false consciousness. *British journal of social psychology*, 33(1):1–27, 1994.

[20] John M Kennedy and Igor Juricevic. Blind man draws using diminution in three dimensions. *Psychonomic bulletin & review*, 13(3):506–509, 2006.

[21] Michael Kim, Omer Reingold, and Guy Rothblum. Fairness through computationally-bounded awareness. In *Advances in Neural Information Processing Systems*, pages 4842–4852, 2018.

[22] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016.

[23] Anurag Koul, Sam Greydanus, and Alan Fern. Learning finite state representations of recurrent policy networks. 2019.

[24] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in neural information processing systems*, pages 3675–3683, 2016.

[25] Adrien Laversanne-Finot, Alexandre Péré, and Pierre-Yves Oudeyer. Curiosity driven exploration of learned disentangled goal spaces. *arXiv preprint arXiv:1807.01521*, 2018.

[26] Andrew Levy, George Konidaris, Robert Platt, and Kate Saenko. Learning multi-level hierarchies with hindsight. In *ICLR*, 2019.

[27] Xiang Li, Luke Vilnis, Dongxu Zhang, Michael Boratko, and Andrew McCallum. Smoothing the geometry of probabilistic box embeddings. In *ICLR*, 2019.

[28] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. *arXiv preprint arXiv:1802.06309*, 2018.

[29] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*, 2019.

[30] Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine. Near-optimal representation learning for hierarchical reinforcement learning. In *ICLR*, 2019.

[31] P-Y Oudeyer, Jacqueline Gottlieb, and Manuel Lopes. Intrinsic motivation, curiosity, and learning: Theory and applications in educational technologies. In *Progress in brain research*, volume 229, pages 257–284. Elsevier, 2016.

[32] Pierre-Yves Oudeyer. *Self-organization in the evolution of speech*, volume 6. OUP Oxford, 2006.

[33] Steindór Sæmundsson, Katja Hofmann, and Marc Peter Deisenroth. Meta reinforcement learning with latent variable gaussian processes. *arXiv preprint arXiv:1803.07551*, 2018.

[34] Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. Ordered neurons: Integrating tree structures into recurrent neural networks. *Proceedings of ICLR*, 2019.

[35] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[36] Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. *arXiv preprint arXiv:1901.10430*, 2019.

[37] Luisa M Zintgraf, Kyriacos Shiarlis, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Caml: Fast context adaptation via meta-learning. *arXiv preprint arXiv:1810.03642*, 2018.