NIPS 2017 Notes
Long Beach, CA

David Abel*
david_abel@brown.edu

December 2017

Contents

*https://cs.brown.edu/~dabel

NIPS 2017 just wrapped up yesterday – what an outstanding conference. My head is absolutely packed with new ideas, methods, and papers to read. This document contains notes I took during the events I managed to make it to. Please feel free to distribute it and shoot me an email at david_abel@brown.edu if you find any typos.

1 Conference Summary and Highlights

My personal highlights were:

1. John Platt's talk on the next 100 years of human civilization.[1]
2. Kate Crawford's talk on AI and bias.[2]
3. Ali Rahimi's test of time talk.[3]. This had lots of conversation buzzing throughout the conference. In the second half, he presented some thoughts on the current state of machine learning research, calling for more rigor in our methods. This was heavily discussed throughout the conference, (most) folks supporting Ali's point (at least those I talked to), and a few others saying that his point isn't grounded since some of the methods he seemed to be targeting (primarily deep learning) work so well in practice. My personal take is that he wasn't necessarily calling for *theory* to back up our methods so much as *rigor*. I take the call for *rigor* to be a poignant one. I think it's uncontroversial to say that effective experimentation is a good thing for the ML community. What exactly that entails is of course up for debate (see next bullet).
4. Joelle Pineau's talk on Reproducibility in Deep RL. One experiment showed that two approaches, let's call them A and B, dominated one another on the exact same task *depending on the random seed chosen*. That is, A achieved statistically significant superior performance over B with one random seed, while this dominance was flipped with a different random seed. I really like this work, and again take it to be at just the right time. Particularly in Deep RL, where most results are of the form: "our algorithm did better on tasks X and Y".

Overall my sense is that the Deep Learning hype has settled to an appropriate level. There's of course loads of excitement about new methods and applications, but we're no longer seeing the over saturation of papers tackling the same thing (except perhaps GAN and Deep RL architectures). The Deep Learning work has found its role in the broader context of ML, which I take to be a good thing.

A few papers I'm excited about reading, most from this NIPS but a few that popped up from conversation:

- Dijk et al. [15]: Information theory and RL.
- Solway et al. [46]: Bayesian model selection for learning good/optimal hierarchies.
- Teh et al. [48]: Robust multitask RL.

[1] Include's opening remarks: `https://www.youtube.com/watch?v=L1jLpkvKPh0`
[2] `https://www.youtube.com/watch?v=fMym_BKWQzk`
[3] `https://youtu.be/Qi1Yry33TQE`

- Harutyunyan et al. [30]: Best paper from the Hierarchical RL workshop. Looks at finding terminal conditions for options that are off policy and prove this enables the right kind of trade offs.
- Hadfield-Menell et al. [29]: Attacks the problem of how to specify reward functions that elicit the right kind of behavior.
- Dann et al. [13]: presents a unifying framework for the PAC-MDP criterion and regret. Basically each of these methods of evaluation have some issues, so they introduce the "uniform-PAC" criterion which overcomes some of these issues (an algorithm is uniform PAC if it holds for all ε).
- Kaufmann and Koolen [31]: Focuses on a problem that sits in between full RL and bandits.
- Finn et al. [20]: Some of the Meta-Learning (for RL) work from Chelsea Finn and colleagues.
- Andrychowicz et al. [2]: the Hindsight Experience Replay (HER) paper. Looks at how to make the most of previous data in RL to be more sample efficient.

2 Monday

First, tutorials.

2.1 Tutorial: Deep Probablistic Modeling w/ Gaussian Processes

The speaker is Neil Lawrence.

Big question: How do we approximate the full Gaussian Process (GP) problem?

Some references to take a look at for more on GPs:

- Deep Gaussian Processes and Variational Propgataion of uncertainty Damianou [12].
- A GP review [11].

A nice observation: GPs are extraordinarily mathematical elegant but algorithmically inefficient. Conversely, neural networks are not mathematically elegant but are algorithmically efficient. So how about a hybrid that takes the nice parts of each?

He actually suggested a method to integrate out all of the input/outputs of a deep neural network that then translates a neural network into a gaussian process. *This is the idea of a deep GP.* We form our distribution over the functions implemented by the network.

Mathematically, then, a Deep GP is just a composite multivariate function:

$$g(x) = f_5(f_4(f_3(f_2(f_1(x))))). \tag{1}$$

So why write this down? His point was that it's important that each
Why deep?

- GPS give priors over functions
- Elegant: derivatives of process are also Gaussian distributed
- For some covariance functions they are universal approximators
- Gaussian derivatives might ring alarm bells
- *a priori* They don't believe in function 'jumps'.

Question: how deep are deep GPs? Why? The paper here [18] takes a stab at answering them. Some application of these techniques [41].

> **Takeaway**: Deep Learning already does supervised learning for text/vision very well. GPs shouldn't try to do those. They should do other things, like Uncertainty Quantification.

.....

2.2 Tutorial: Reverse Engineering Intelligence

Next up, Josh Tenenbaum is giving a talk on reverse engineering intelligence from human behavior.

Paper from Warneken and Tomasello [53] on emergent helping behavior in infants. He showed a video where a baby comes over and helps an adult without being told to do so. Cute!

Goal: Reverse-engineering common sense.

Tool: probabilistic programs. Models that can generate next states of the worlds that act as an approximate "game engine in your head" is really powerful. Potentially *the* piece missing. A mixture of intuitive physics and intuitive psychology.

Engineering common sense with probabilistic programs:

- *What?* Modeling programs (game engine in your head).
- *How?* Meta-programs for inference and model building, working at multiple timescales, trading off speed, reliability, and flexibility.

Lots of different systems working over different time scales: Perception, Thinking, Learning, Development, Evolution (from milliseconds to minutes to lifetimes).

Paper: The intuitive physics engine [6]. The idea is to use monte carlo like simulations from a probabilistic intuitive physics engine to make inferences about properties of the world. For instance, we particular tower of blocks might more often lead to the tower falling in simulation, we we ascribe the property of "unstable" to the tower.

A separate paper focuses on a similar problem but tries to use a CNN to *learn* the intuitive physics engine ("PhysNet"). Training (on 200,000 images!) on simulated data it was able to transfer to

real colored blocks.

Conversely, the intuitive physics engine can answer questions about complex compositional predicates in scenes. Can transfer language directly into simulations.

Task: What if a table is bumped hard enough to knock blocks off a table? There are two types of blocks, red and yellow. We saw different configurations of blocks and answered which type might fall off. Really cool demo!

Question: How might neural networks come in to the intuitive physics paradigm? Some of his papers that focus on this: Neural Scene De-rendering [56], De-animation [55]. Uses these to do some really cool physics predictions. Can also answer questions like "what if X happened?".

Another paper that explores something similar, but evaluates an agent planning Baker et al. [4]. Presented some follow up work (in prep) where we try to infer what an agent is doing based on a snapshot of behavior.

At the end, Josh talked about learning for a bit and how it fits into this picture. When Josh says "learning", he means "program learning programs". He showcased a cool testbed of handwritten omniglot characters based on MNIST [34], and described a probabilistic program technique for learning compositional programs that learn to recognize and write characters. Ultimately this is just Bayesian Inference: find the program that would have generated these characters.

Ultimate Learning Goal: Rapid Learning of rich concepts (like language, tools, cultural symbols, skills).

Big picture, cogsci/psych tell us that learning is much more about modeling the world, and acquiring commonsense understanding. Program synthesis gives us a path to study these.

Takeaway: Learning can/is/should be cast as programming the game engine of your life.

.....

2.2.1 Part Two: Implementing These Ideas

Vikash Mansinghka is giving a talk that follows up on Josh's with an emphasis on how we can actually implement a lot of the core principles introduced.

Probabilistic Programming can be boiled down to two technical ideas:

1. Probabilistic models can be represented using programs that make stochastic choices
2. Operations on models such as learning and inference can be represented as meta-programs that find probable executions of model programs given constraints on their execution traces.

He's basically going over how to write probabilistic programs. Summary of languages:

Language	Specialty
BLOG, ProbLog	Logic and Probability
Figaro Infer.NET	Embedded
Stan, LibBi, PyMC	Bayesian Stats
Church WebPPL	Automatic Inference
Edward Pyro ProbTorch	Deep Learning Support

Choosing the right inference strategy is really important; sometimes the wrong inference strategy will do poorly even with a big computational budget.

This work culminated in the Automatic Statistician [24], which tackles the problem of going from data to a probabilistic program that generates the data.

The code is actually pretty simple. He showed two blocks of code, each of which was about 5 or so lines, that carry out the simulation and prediction. Really cool! In a table he showed that the core of the code was 70 lines, compared to 4000 lines in a non-probabilistic program implementation.

Next up: inferring goals from action via inverse planning [4]. Had some cool examples where probabilistic programs infer goals based on behavior.

Now we're looking at how to get to full on scene understanding from 2d images. These techniques are explored by their 2015 paper [32].

Future directions:

1. Integration of probabilistic programming and deep learning.
2. Optimized platforms for scene understanding
3. Scaling up from perception to thinking, learning, and development.

Question: What techniques can get us there?

- Meta-languages: might allow composition or transfer across inference strategies or models.
- Monte Carlo Techniques: fast runtime for MC.

Takeaway: Probabilistic programs are a really powerful (but simple!) tool for complex inference and prediction tasks.

.....

2.3 NIPS 2017 Welcome

General Chairs: Icabelle Guyon and Ulrike von Luxburg

New this year: competition track. Five tracks and the art contest:

1. Conversational Intelligence
2. Human-computer Quiz Bowl
3. Learning to run
4. Personalized medicine
5. Adversarial attacks and defense
6. DeepArt context

Best Papers:

1. Safe and nested subgoal solving for imperfect-information Games by Noam Brown and Tuomas Sandholm
2. A Linear-Time Kernel Goodness of Fit-Test by Jitkrittum, Xu, Szabo, Fukumizu, Gretton.
3. Variance-Bases Regularization with Convex Objectives by Namkoong, Ducci.

Test of time award:Random Features for Large-Scale Kernel Machines by Ali Rahimi and Benjamin Recht (NIPS 2007).

2.3.1 Statistics about the Conference

Used a mixed-integer linear program to assign papers to reviewers, and reviewers to area chairs. Reviewer Constraints:

- No conflicts of interest
- Maximize positive bids and minimize negative bids
- No more than 2 reviewers from any institution on any paper
- No more than 6 papers per reviewer
- No more than 18 pages per area chair
- No more than 8 area chairs per senior area chair

.....

2.4 Keynote: John Platt on the next 100 years of Human Civilization

Goal: Want to live in a world in which every person on earth can use as much energy as a US resident does today. Because energy is a necessary condition for a high standard of living.

Total power consumed has increased 2.2% per year throughout human history. If we extrapolate, we'll be used 113 TW, which is the US power rate for 11.2 billion people.

Total Registration	8000 (vs. 5000 last year)
Tracks	2
Submissions	3240 (vs. 2500 last year)
Subject Areas	156 (150% increase)
Top Area:	Algorithms (900), Deep Learning (600), Applications (600)
Unique Authors	7,844
Author demographics	90% men, 10% women
More	Industry 12%, 88% Academic
Reviewers	2093
Area Chairs	183
Reviews	9847
Acceptances	679
Acceptance Rate	21%
Orals	40
Spotlights	112
Posters	527
Paper on arXiv?	43% Yes
Reviewer saw on arXiv?	10% Yes
Not Posted Acceptance?	If not online, 15% accepted
Posted Acceptance?	If online, 29% accepted
Reviewer saw online	If reviewer saw, 35%

Point: Human civilization requires tremendous power. In 2017, Human civilization uses 1 Mount St. Helens' eruption every 93 minutes. By 2100, we'll use 1 Mount St. Helen's every 15 minutes.

We'll need 0.2 Yottajoule, which is "alotta Joules". It's enough energy to boil off the Great Lakes 3.4 times over. 0.2 Yj is the amount of energy consumed by humans so far.

Point: Wow we need a lot of energy.

Question: Can we use existing technology to cover 0.2 Yj? Maybe. How about any of our existing methods, like:

- Fossil fuels? Nope! They won't scale. Rise in global mean temp is too high, so, we can only supply 8% of next 100 years of energy, **max**.
- How about solar? wind? fission? carbon?

  The point: Zero-carbon energy is electricity. Electricity has a particular economic structure:

  - Capacity costs: buiilding the plant, salaries, and so on. These are measured in terms of $ per peak power.
  - Output costs: fuel costs, maintenance, and so on. Cost here is $ per unit energy out.

– So: the utilization of the plant gives you:

$$\frac{\text{Capacity costs}}{\text{Utilization}} + \text{Output Costs} \tag{2}$$

– So it's less expensive to run your plant all the time. If you run it less often, you amortize the cost of your capital across many Joules.

– You can also waste the output of a plant. This happens with variable of sources like wind/solar. Trouble is: gap between solar generation and demand. Maybe you triple capacity of solar. Sometimes this will result in wasted energy.

Conclusion: Utilization is critical, and must take into account all sources of energy, like variable sources (solar, hydro, natural gas, etc.). As you increase the amount of variable generalization, you get lower utilization, and so you increase the cost of your energy.

So we have to evaluate these techniques in a holistic way, as in Platt et al. [38].
Assumptions about the future (from the paper above):

- Solar power cost decreases by 60%
- Storage cost goes down by 30%
- Natural gas goes up 120%
- High voltage DC lines to ship renewable energy across US

Question for the model: compute marginal system cost for renewables.
Natural gas: 4 cents / kW hour. If you've already built the plant, then it's only 2 cents / kW hour.

Point: So, if you use more and more renewables, you start to incur these extra costs (discussed above). Particularly as you get to around 0.8 usage.

Bottom Line:

- Renewables are competitive up to 40% of electrical demand
- Renewables can't compete with paid-off fossil fuel pants
- Renewables can't replace industrial heat

So, we can use lots of renewables, but not all. Dave: I want to hear/research more about this point.

2.4.1 Radical Research into Zero-Carbon Energy

One idea: Fusion Energy. Fusion is why the sun shines. At sun's core, atoms become plasma. Lots of temperature (15k Kelvin?) and lots of pressure (250 billion atmospheres). Under these conditions, energy is released. If we can harness this process, we'd need 6 km^3 of ocean to get to our 0.2Yj.

Fusion energy has been pursued for 70 years. Lots of ideas for fusion, but basically it's insanely difficult. Google working with a company called TAE, specializes in a plasma architecture called FRC.

Metric for fusion success:

$$Q = \frac{E_{out}}{E_{in}} \tag{3}$$

The goal is to get to *breakeven*, where $Q > 1$. The highest achieved so far is $Q = 0.67$. What we really need is to do better than the renewable cost: this is when $Q > 5$.

Criterion for $Q = 1$. The triple product:

$$nT_{ion}\tau \geq C \tag{4}$$

Where n is the number of nuclei per unit volume, T_{ion} is the temperature of nuclei, τ is the confinement time, C is some constant that depends on fuel cycle, plasma, and so on.

This leads us to a scaling law, where τ has to be long enough, and T has to be hot enough. Preliminary data suggests that FRC scales really well: higher electron temperature, the lower your heat loss rate.

2.4.2 Machine Learning for Fusion

Project one: Norma has thousands of parameters, like:

- How much plasma?
- What ionization voltage?
- Voltage and timings of coils?
- And more...

And, some parameter settings harm the machine. So it's both a constrained optimization problem and an exploration problem.

Innovation: MCMC with human preferences, where physicists and accept or reject and do new exploration. This is all summarized in this paper: Baltz et al. [5].

Project Two: Debugging Plasma w/ Inference.

Experiments can go wrong! And they want to figure out why. they have some sensors available like magnetic sensors, video camera, and so on.

Goal: Infer hidden state of the plasma (position, shape, and state), based on the sensor data. Use a prior over plasma state. Then do variational inference! ELBO! Wowza. Cool.

Question: Are there other projects like this?

3 Tuesday

The test of time talk from Ali Rahimi was split in half: the second half entirely dedicated to poignant commentary about the current state of ML, and deep learning in particular. This chunk of the talk is available here.[4]

Basically, Ali suggested that ML is the new alchemy. That is, we're moving in a bad direction that misses good, simple, explanatory models, experiments, and theory.

His main call was for simplicity: "Simple experiments and simple theorems are the building blocks that help us understand complicated systems. Dave: Love it!

Next up: Kate Crawford!

3.1 Kate Crawford: The Trouble with Bias

Kate is amazing! Today she's talking about bias in AI/ML.

Claim: The rise of machine learning might be as big as the rise of computers. Dave: Sure!

Lots of high profile news stories about bias. For instance, Google's NLP service was labeling things like 'gay' as negative, and "white power" as positive. Maps of Amazon fresh availability look eerily similar to 1930s segregation maps.

One of the most famous articles was "Machine Bias: There's software across the country to predict future criminals, and it's biased against blacks."[5].

Mustafa S.: "Because of the scale of these systems, they can be hitting 1-2 billion users per day. That means the costs of getting it wrong are very high."

Kate: "When we consider bias as purely a technical problem, we are already missing a big part of the picture".

Talk today:

1. What is bias?
2. Harms of allocation (where we are now)
3. Harms of representation (what we're missing)
4. Politics of classification (the big picture)
5. What can we do?

[4] https://youtu.be/Qi1Yry33TQE?t=660

[5] https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

So, **what is bias?** Bias means judgment based on preconceived notions or prejudices as opposed to facts and evidence. This notion of bias is really hard to avoid in pure Machine Learning terms. We can have an unbiased system producing a biased system in a legal/social sense.

Where does bias in ML come from? Lots of places! Sometimes the training data was constructed poorly. For instance: Stop and frisk (from 2006). 4.5 million(ish) people were stopped on suspicion of being criminals. 83% of people stopped were hispanic or black. The social scientists came in and suggested that the practices of the police officers was itself highly biased.

The Problem with Bias: a main distinction in studies of bias is about *allocation* as opposed to *representational* bias. Allocation is immediate and easily quantifiable, whereas representation is long term and difficult to formalize. So, allocative harms have been focused on previously. A few examples/types of these sorts of harms:

- *Stereotype:* Google translate: translate "O bir hemsire, O bir doktor" (from Turkish) into English, they translate into "She is a nurse, He is a doctor", even though the Turkish language is gender neutral.
- *Representational harms:* Cameras mischaracterize asian faces as blinking, can't recognize people with darker skin colors.
- *Denigration harms:* if you type in "jews should" to Google, the auto-complete was "be wiped out".

From a paper by Kate and her colleagues, they suggest some technical responses, including improving accuracy, blacklisting certain methods/data, scrubbing data sets to neutral, being aware of demographics.

Aristotle: "natural classification". Go out into the field, observe, and conclude.

John Wilkins, founding member of Royal Society, published a book that classified the universe into 40 categories during the enlightenment. Wilkins shows the popularity of *language* as a means of classifying.

Kate: "The world of classification will always be a result of culture." Follow up: "Data sets reflect the hierarchy of the world in which they are collected...".

Kate Q: What would it look like to make a visual/encyclopedia of machine learning? Her and her colleagues made one (I'll add the image later). These systemic choices are often cultural, and more importantly, arbitrary. Another example: Wang and Kosinsky paper detecting sexual orientation from images [52].

In conclusion, we should do three things:

1. Need to start working on fairness forensics.
2. Really start taking interdisciplinary research seriously.

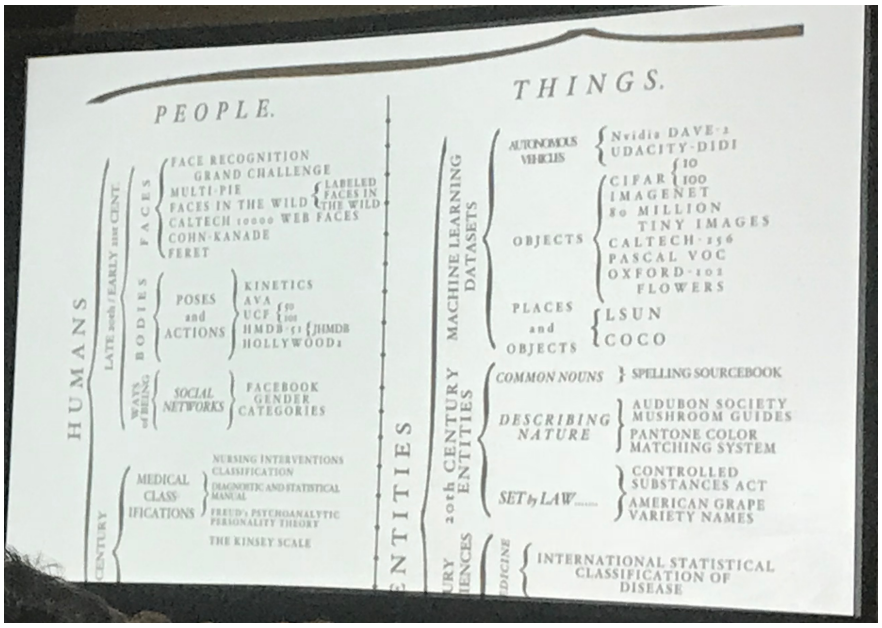


Figure 1: A Snapshot from the Encyclopedia of Machine Learning

3. Think harder about the ethics of classification.

   Trump is asking the ML community to write software to do screening at the border. So, the question is: are there some things we shouldn't build? And how should we answer that question?

Takeaway: Bias is a highly complex issue that permeates every aspect of machine learning. We have to ask: who is going to benefit from our work, and who might be harmed? To put fairness first, we *must* ask this question.

.....

3.2 Best Paper Talk: Safe and Nested Subgame Solving for Imperfect Information Games [10]

Focus: Imperfect-Information games like poker where information is hidden. Poker is the variant that has been established as the core baseline of the area. 10^{161} decision points. Libratus:

- 120,000 hands over 20 days in January 2017
- Libratus beat top humans in this game. 147 mbb/hand.
- p-value of 0.0002
- Each human lost a lot to Libratus.

Q: Why are imperfect information games so hard?

Answer: Because an optimal strategy for a subgame cannot be determined.

Their strategy emphasizes *exploitability*. AlphaGo and OpenAI's Dota bot both suffer by leaving open lots of paths to take advantage of the AI's weaknesses. Conversely, exploitability is a theoretical property that ensures safe distance from a losing strategy.

They introduce several new results that lower exploitability by a huge factor:

1. **Safe subgame solving**
2. **Reach subgame solving**
3. **Nested subgame solving**

Takeaway: They beat humans at no-limit heads up poker. The innovation is to focus on variants of subgame solving that lead to lower exploitability.

.....

3.3 Theory Spotlights

3.3.1 Bandit Optimization

Problem: generalization of multi-armed bandits to a full on optimization problem.

You have an unknown convex function: $L : \Delta^K \mapsto R$. Observe vector $\hat{g}_t$, proxy of the gradient such that w/ probability $1 - \delta$:

$$|\hat{g}_t - \Delta_i L(p_t)| \leq C \tag{5}$$

New Algorithm: UCB Frank-Wolfe.

3.3.2 Computational Perspective on Shallow Learning

Paradigm: Empirical Risk Minimization. Can't compute exact solution due to large dataset, so instead approximate it with SGD/Neural Networks/Kernels. Need smooth functions, but most functions aren't smooth (non-smooth kernels lead to computational difficulty). They came up with an eigen trick to enforce smoothness.

3.3.3 Monte-Carlo Tree Search by Best Arm Identification [31]

Claim: In between Bandits and AI.

Problem: find best move at root from samples of leaves (of a game/min-max tree). Suppose coins at leaf that determine payoff, but coin weights are unknown. How might you solve the game?

Can we guarantee that the learner uses a small number of samples to find the best policy?

When tree is depth one, it's just the multi-armed bandit (LUCB, UGapE, and so on).

One idea is to put a learner at each node in the tree and solve it hierarchically. Doesn't actually work.

Their idea:

- Put conf. intervals at each of the leaves.
- Interpolate child intervals into the parent.
- Then, at the root, we do a typical bandit algorithm (best arm alg). Use an optimistic policy to figure out how to sample.

Algorithm is: (1) Correct: that is, it's (ε, δ)-PAC and (2) has sample complexity of roughly:

$$O\left(\sum_{l \in \mathcal{L}} \frac{1}{\varepsilon^2} \log \frac{1}{\delta}\right) \tag{6}$$

3.4 Deep Learning Spotlights

Now for some Deep Learning:

3.4.1 Deep Mean-shift priors for image restoration [9]

Goal: Restore degraded images by removing blur or noise.

Challenges: ill posed, needs prior knowledge, generic.

Their idea: Use a deep mean-shift prior given a large database of images/patches. Focus on density estimate using denoising autoencoders.

3.4.2 Deep voice-2: Text to Speech [25]

Goal: Text to speech.

Their idea:

- Neural TTS
- Trainable speaker embeddings
- Synthesize speech from multiple speakers, augment each model with a low-dimensional speaker embedding.
- Speaker embeddings initialized randomly.
- Majority of parameters shared between speakers.

This combination of things seemed to work! Neat.

3.4.3 Graph Matching via MWU

Problem: GraphMatch! Find some entities in two images that match.

They have a new result that under the MWU update, the Lagrangian is monotonically increasing. Also, the converged solution is KKT optimal.

That is: KKT = Karush-Kuhn-Tucker conditions. Dave: Worth looking at!

3.4.4 Dynamic Routing in Capsules [43]

Idea: Capsule Network.

New way of doing classification. Take each hidden layer and turn it into a capsule. Each capsule is responsible for various entities in the data.

Claim: Capsules enable networks to be equivariant.

Dave: Also, Intel hosted a party tonight at a bar near the venu with Flo Rida. We drove by it on the way to dinner and the line was around the block. Puzzling times!

4 Wednesday

I had some meetings so I missed the morning session.

4.1 Speaker: Pieter Abbeel on Deep RL for Robots

Played a video of a PR1 doing some household chores from the 80s(?). But! It's teleoperated. Far more time consuming than just doing the chores yourself. However: the mechanical engineering is *there*. We just need to fill in the AI piece.

This talk: the unsolved pieces to the AI robotics puzzle:

1. Sample efficient RL
2. Long horizon reasoning
3. Taskability (Imitation Learning)
4. Lifelong Learning
5. Leverage Simulation
6. Maximize Signal Extracted from Real World Experience

4.1.1 Sample Efficient Reinforcement Learning

Compared to supervised learning, RL has some more challenges:

- Credit assignment
- Stability
- Exploration

Deep RL has done lots of cool things: Atari, Go, Chess, Learning to Walk (all from high dimensional inputs). One cool example I hadn't seen: the superball robot [42].[6]

Q: How good is the learning, actually?

Answer. One main weakness: *huge* gap in learning efficiency (between AI/people).

Q: Can we develop faster/more data efficient algorithms that take advantage of the world's structure?

Pieter's suggestion: Meta-Learning. Let's *learn* the algorithm itself. Meta-RL! Learning to do RL.

Goal of Meta-RL is basically to search in agent space for a good agent that can solve environments drawn from some distribution. They first try to explore this space in Multi-armed bandits and find

[6] `https://www.youtube.com/watch?v=0eC4A2PXM-U`

their algorithm does as well as Gittins. Apply a similar technique to some 3d mazes that involve vision as observation and it works. Algorithms are called RL^2, $MAML$, $SNAIL$.

Idea, based on prior experiments: Can we do end-to-end learning for parameter vector θ that is good init for fine-tuning for many tasks? Yes! They do exactly this (paper at ICML this year). Model called MAML[21]

New Theory: MAML is fully general.[19]
Dave: What does "general" mean? Will have to read that paper.

4.1.2 Hierarchies

At the low level, 10^7 timesteps per day of low level control.

Pieter Thought on how to think about hierarchies: Let's formulate Hierarchical RL as meta learning: "Agent has to solve a distribution of related long-horizon tasks, with the goal of learning to do well across the distribution."

4.1.3 Imitation Learning

Idea here: how can we learn from demonstrations?

Goal: one-shot imitator! How can we learn to solve a collection of related tasks from a single demonstration? They do some block stacking variants in this paper: Duan et al. [17].

4.1.4 Lifelong Learning

Focus: continuous adaptation. That is, their formulation treats an agent to be good at dealing with **non-stationary** environments. They do some Meta-Learning again and evaluate on a RoboSumo task.

4.1.5 Leverage Simulation

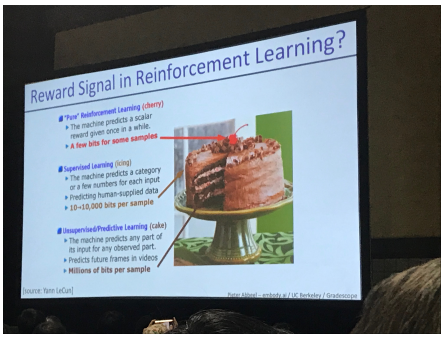
A few approaches:

1. Use really accurate simulators.
2. Domain adaptation with approximate simulations.
3. Domain randomization: if the model sees enough simulated variation, the real world may look like another simulation.

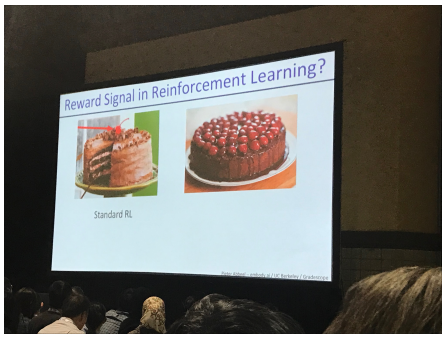
4.1.6 Yann LeCun's Cake

Learning is a cake! Where:

- Reinforcement learning is a cherry
- Supervised learning is the frosting



(a) Yann LeCunn's Cake



(b) Pieter Abeel's Cake

- Unsupervised learning is the cake

Pieter Abbeel: I think it's more like a cherry cake! Cherries everywhere! Claims that Hindsight Experience Replay [2] is exactly this cherry cake.

Other Challenges:

- Safe Learning
- Value Alignment
- Planning and Learning

Meta-Learning: enables discovering algorithms that are powerwed by data/experience as opposed to human ingenuity. This will require more compute, which is something we can definitely get.

Takeaway: The mechanical aspect of AI driven robots has been around for awhile. What we need is to address some central challenges of RL (like sample efficient RL, lifelong and hierarchical RL). Pieter's group uses a Meta-Learning approach for most of these problems.

4.2 RL Session

Now for the general RL session.

4.2.1 ELF: Framework for RL + Game Research [49]

RL involves lots of design choices:

- CPU, GPU?
- Simulation?
- Replays?
- Concurrency?

Goal: make a flexible but efficient RL framework for experimenting:

- Extensive (any games with C++ interface can be used)
- Lightweight (Fast, minimal resource usage, fast training)
- Flexible (choice of different RL methods, environment-actor topology)

Available at: `https://github.com/facebookresearch/ELF`.

4.2.2 Imagination-Augmented Agents for Deep RL [54]

Dave: The title seems unjustified to me. It's a new architecture for model-based RL.

Advances in Deep RL, however, they're data inefficient and limited generalization.

Solution: model-based RL. Enables planning so we can explore and generalize more effectively. Sadly, no model-based RL methods have worked yet.

Imagination Augments Agents: a model combining model-free and model-based aspects. The learns a model of the world and queries it for information and learns to interpret the predictions in order to act.

High Level:

- Environment Model: recurrent model that makes environment prediction at every time step.
- Simulation Policy: policy used for simulated rollouts.
- Rollout encoder: recurrent model which extracts information from the rollout.
- They compute an **imagination-augmented** code from the rollout encoder, and a **model-free** code.
- They then combine these two codes and use them to compute a policy.

Tested on sokoban from images. Their thing does better than A2C, and an augmentation of A2C with more parameters.

The coolest result is that they do well even when the model used is bad:
Someone asked a good question: what does the rollout encoder actually produce? The author said: I don't know.

During Q&A, Tom Dietterich said (roughly): yesterday we heard a call for the rigor police, I'm here on behalf of the hype police. Retract the term imagination from this paper. This is just model-based RL. Dave: I agree!
Next, 5 minute spotlights.

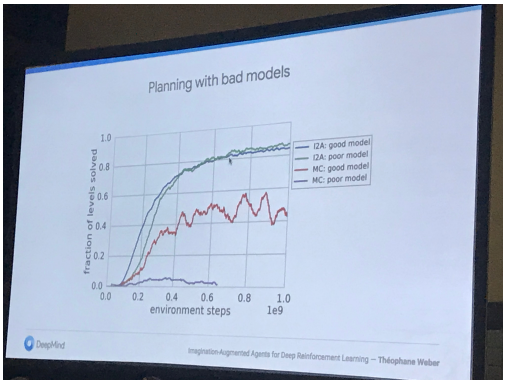


Figure 2: Even with a bad model the approach seems to work well.

4.2.3 Simple module for relational reasoning [44]

Relational reasoning is important! They introduce the relational network:

$$\mathrm{RN}(O) = f_\phi \left(\sum_{i,j} g_\theta(o_i, o_j) \right) \tag{7}$$

Where g is a function that operates on pairs of objects, and o_i and o_j are objects. Tested on some data sets, including CLEVR, which is about question answering from visual data. Achieved human level performance on the task.

4.2.4 Scalable TRPO w/ Kronecker Appromixation [57]

They build on TRPO by focusing on removing some expensive computation at its core.

Their key insight is the Kronecker-Factored Approximation. Main insight: approximate the I-th Fisher matrix block using a Kronecker product of two small matrices. As a result, they reduce the computational burden by inducing compact matrix representations.

4.2.5 Off-Policy evaluation for slate recommendation [47]

Adith presents! Motivated by tasks that call for an ordered ranking of items. They introduce a new estimator for evaluating policies and show that under certain conditions, the policy is unbiased, and experimentally requires much less data than previous estimators of this form.

4.2.6 Transfer learning with HIP-MDPs [14]

Motvation: Real world tasks are often repeated, but not exactly.

Define a new class of MDPs that includes a parameter θ which defines a parameterized transition function. Then, learning is done in the parameterized space; if the agent effectively learns the parameter, it can transfer knowledge to any MDP in the class.

4.2.7 Inverse Reward Design [29]

Motivating Question: At best, (deep) RL reduces the problem of generating useful behavior to that of designing a good reward function.

Observation: Reward engineering is hard.

Problem: When you write down pieces of the reward function, you effect the value structure of the rest of the world! So, it makes it really hard.

Goal: Preferences over some state of the world that imply other preferences over other states in the world. What we'd instead like to do is to move the other states to question marks.

Difficult question: which part of a reward function should be considered unspecified?

Key Idea: Rewarded behavior has high true utility in the training environments.

4.2.8 Safe Interruptibility [27]

Q: How can we safely interrupt RL agents?

Provide some new clarificatory definitions on this space, add some new theorems as a result.

4.2.9 Unifying PAC and Regret

New Framework: Uniform-PAC that unifies the PAC-MDP criterion and Regret.

Limitations:

- PAC: we don't know how big the mistakes are. Also, algorithms aren't incentivized to get better than ε optimal.
- Regret: doesn't distinguish between a few severe mistakes and many small mistakes.

Since they capture different aspects, they can't be translated to one another. If you have a PAC bound, then you get a highly suboptimal regret bound, and a regret bound tells you nothing about PAC bounds.

Uniform-PAC: bound mistakes for all settings of ε. This immediately guarantees strong PAC and regret bounds.
For instance, a Uniform-PAC bound of:

$$\tilde{O}\left(\frac{SAH^4}{\varepsilon^2}\right) \tag{8}$$

Gives you the same PAC bound, and a regret bound of:

$$\tilde{O}\left(H^2\sqrt{SAT}\right) \tag{9}$$

Also use this to make a new algorithm.

4.2.10 Repeated IRL [1]

Again, emphasis on difficulty of specifying a reward function.

IRL yields an unidentifiability problem: lots of reward functions consistent with same behavior. The good news is that we don't really need the reward; we can still leverage IRL tools to do quite well. However: can't generalize.

Their work: receive demonstrations in a bunch of tasks. Some unknown reward function θ_*. Learn θ_* from demonstrations on a few tasks and generalize to new ones.

5 Thursday

First, Yael Niv on Learning State Representations!

5.1 Yael Niv on Learning State Representations

Main question: how do we learn from relatively little experience, a representation that facilitates generalization and effective decision making across a variety of tasks?

The computational problem: the curse of dimensionality at the core of RL forces the statistical burden to be extremely high. So, we *need* some dimensionality reduction.

We'll think about learning/generalization of representation as clustering. Specifically, Bayesian inference with infinite capacity prior (Chinese restaurant prior). A generative model of the environment. That is:

1. Observed events are caused by latent causes.
2. A prolific latent cause is more likely to cause the next observation.
3. The number of possible latent causes is unbounded.

$$P(C_t = k) = \begin{cases} \frac{N_k}{t+a} & \text{if } k \text{ is old cause} \\ \frac{a}{t+a} & \text{o/w} \end{cases} \tag{10}$$

Main Hypothesis: People group experiences and generalize learning across them.

Yael and Sam Gershman ran some experiments with humans to evaluate this hypothesis. A few points:

- "Inference about latent causes defines the boundaries of generalization."
- "Real world learning learning as clustering with a growing set of clusters."
- "Learning happens within a cluster, not across boundaries."

Takeaway: Goal for AI: see AI's solve complex tasks with little data. Suggestion: transform complex tasks to easy ones via abstraction.

5.2 Deep RL Symposium

First, David Silver on Go.

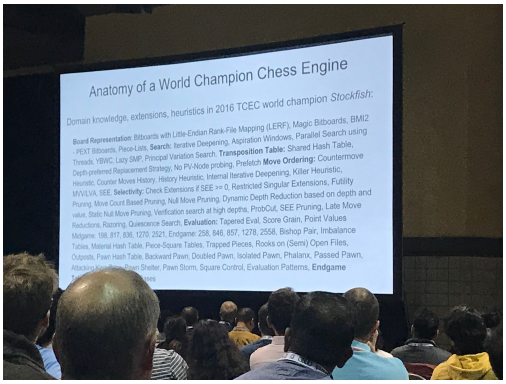


Figure 3: The set of heuristics used in stockfish.

5.2.1 David Silver: AlphaGo and AlphaZero

AlphaGo consists of:
(1) A policy network:

$$\pi : \text{Position} \mapsto \Pr(a) \tag{11}$$

(2) A Value Network predicts likelihood of win/loss:

$$\hat{v} : \text{position} \mapsto [-1 : 1] \tag{12}$$

Main Idea: Use these two networks to avoid exhaustive search. Policy networks reduces breadth. Value network reduces depth. Then, add MCTS. They beat the top human players with ALphaGo Master 60-0.

Now, AlphaGo Zero:

- No human data: learns solely by self-play reinforcement learning, starting from random.
- No human features: only takes board as input.
- Single neural network: policy and value networks are combined into on neural network (resnet).
- Simpler search: no randomized Monte-Carlo rollouts, only uses neural network to evaluate.

Timeline: after three days of training time, AGZ surpasses Alpha Go.

Most recent work: applying AlphaZero to Chess, Shogi, and Go. How does AlphaZero do across games?

Existing Chess World Champion: Stockfish. Consists of a huge array of heuristics about chess: Couple things that might make Go differnet from Chess/Shogi, like the presence of drawing as a game outcome, board symmetry, action space, and so on.

In four hours of training, ALphaZero beat stockfish. In two hours, it surpassed Elmo (the computer champion at Shogi). After eight hours, it beat AlphaGo Lee.

What next? Move AlphaZero like things beyond games.

Takeaway: Self play works extremely well, in part because the value of the data each round gets better. Going forward we should make simple approaches to be more general (work on multiple games).

5.2.2 Soft Actor Critic

Based on optimizing the MaxEnt Objective:

$$\pi^*(\cdot \mid s_t) = \arg\max_{\pi} \mathbb{E}_{\pi} \left[\sum_t r(s_t, a_t) + \alpha H(\pi(\cdot \mid s_t)) \right] \tag{13}$$

Does really well on a variety of cheetah and robotic domains, seems really robust to turbulence. Dave: I'm feeling sick so I went home. I'll fill in some notes on the Joelle's reproducibility talk later when I watch the video

6 Friday

Dave: I missed the Friday workshops because I was feeling a bit sick.

7 Saturday: Hierarhical Reinforcement Learning Workshop

First up, David Silver on subgoals in HRL.

7.1 Invited Talk: David Silver on Subgoals and HRL

Subgoal Vector Space: Each subgoal parameterized by a vector $g \in \mathbb{R}^n$. Could do it w.r.t. pixel control or feature control (they try both with various deep RL agents).

UNREAL architecture: unsupervised RL w/ auxiliary tasks. A3C with subgoal-out for coordinate pixel/feature control. Shoe some results: it does well.

What is the nature of subgoals? What does it mean to have good subgoals?

1. **Abstraction!** Been used in classical planning for decades. With abstraction, we can *ignore* some dimensions (features).
2. **Coordinated Subgoals**: rich behavior requires coordination between representation of subgoals. Sutton calls this the "subgoal keyboard".

So how can we achieve these ideas? We can specify the geometry of the subgoal space. Their idea: "target subgoals". That is, just get to state s_g. The weakness of this approach is that you don't get abstraction. Instead, care about directed subgoals; pseudo-reward given when you get closer to solving the subgoal pair.

Temporal Abstraction w/ Subgoals. Feudal RL as a framework for HRL [51]. Main idea is that actions choose subgoals.

7.2 Contributed Talks

Now onto quick 10 minute talks.

7.2.1 Landmark Options via Reflection in Multi-task RL

Motivation: learn several goal states simultaneously that cover the state space. Learn from a vector of rewards, from *multiple goal* perspectives. When motivated to reach a given state, leverage knowledge nearby landmark states. Example: what's 12×13? Well, we know it's close to 12×12. That's the intuition.

Multi-task Learning:

- Sequence of episodic, goal-based MDPs $\{M_i\}_{i=1}^T$
- Minimize lifelong reward
- Landmark states $L \subset S$, with accompanying options/value functions.

Two phases:

1. Phase 1 (pre learning phase): agent explores environment and can learn some preliminary things.
2. Phase 2: agent is assigned a sequence of episodic MDPs.

Restricted access to $o_\ell \in \mathcal{O}$. Landmark options can only be accessed via reflection.

New results: The lifelong regret is at most $T(2SAD)$, with D the diameter and T the horizon (I think).

7.2.2 Cross Modal Skill Learner

Starts with speedrun of super meat boy. A key aspect is sound-based feedback to make decisions.

So: they use *all* available percepts to make decisions. Main challenge: how can we incorporate multi-modal information in a sample efficient way?

New algorithm: Crossmodal attentive skill learner (CASL).

7.3 Invited Talk: Jurgen Schimdhuber on HRL and Metalearning

True goal: build a general problem solver that can learn on its own.

First problem: how can we generate the right subgoals? Jurgen mentioned an old 1990 paper of his that generates subgoals in a differentiable way. Now suggests that we ought to compress one network into another (and adds that this is from a paper of his in 1991-92).

Now introduces PowerPlay [45]. Solves and continually invents problems; training an increasingly general problem solver that searches over the simplest of all currently unsolved problems (I'd guess a Solomonoff induction like inference over problem space).

7.4 Invited Talk: Pieter Abbeel on HRL

Pieter thinks HRL is crucial for success in RL. Dave: I agree!. Mostly focused on Feudal Networks and the FuN paper from Deepmind that David Silver talked about.

Information Theoretic approaches:

1. Jurgen: formal theory of creativity
2. Shakir Mohamed: Variational info max for intrinsic motivation.
3. Karol Gregor: Variational intrinsic control.

Why HRL?

- Credit Assignment
- Exploration
- Generalization to new tasks

Their papers: Stochastic Neural Nets for HRL [22], RL^2 [16].

7.5 Best Paper Talk: Learning with options that terminate off policy [30]

Focus is on the termination conditions of options. Option length induces a tradeoff:

- Longer options yield more efficient learning
- But, since we commit to them, shorter options yield better solutions.

Key Problem: How to benefit from the efficiency of long options without suffering from their potentially non-ideal quality?

Challenge is that the solution of planning with option is tied to the termination condition.

Their Contribution: Options that terminate off-policy, which extends the counterfactual off-policy intuition to option termination and formulate an algorithm that achieves this. Present an algorithm

for doing this

Off policy learning is the ability to disambiguate behaviors from targets. On-policy means you're acting according to the target policy, while off-policy means the behavior policy is different from the target policy. One idea is to use importance sampling.

Intuition for the Key Technical Idea:

- On a single step (s, s'), an option o, and a policy over options μ. Consider the intra-option td error:
$$\delta_s^{u,\mu} = R_s + \gamma \left((1 - \beta_{s'})q(s, o) + \beta_{s'}\mathbb{E}_{o' \sim \mu}\left[q(s', o')\right]\right) - q(s,) \tag{14}$$
- Looks like off-policy corrected TD-error:
$$\delta_s^{\mu} = R_s + \gamma\mathbb{E}[q(s', o')] - q(s, o) \tag{15}$$
- On-policy option-local TD-error:
$$\delta_s = R_s + \gamma \max_{o'} q(s', o') - q(s, o) \tag{16}$$
- **Key Idea:** Instead of doing this implicitly, lets do it explicitly.

Summary: It's useful to think of options and multi-step temporal differences in a unified way.

7.6 Posters

A few posters caught my eye during the spotlights:

- Recursive Markov Decision Processes: A new formulation of hierarchies by recursing on MDP structure.
- Eigenoption Critic: An extension to Marlos' work [35] to a new NN architecture.
- Hindsight Policy Gradients: use hindsight to improve sample efficiency of policy gradient.
- Empirical Evaluation of Optimisim with Option from Ronan Fruit (follow up to [23]): compare UCRL, SUCRL, and FSUCRL.
- Intrinsic motivation + Hierarchical RL for human modeling: trying to address the question about how *humans* create hierarchical representations.
- Option-critic + Proximal Policy Optimization: Goal is to achieve good options! How?

7.7 Invited Talk: Jan Pieters on imitation HRL for Robotics

Dave: I missed some of Jan's talk from being at lunch
What are the big questins of HRL:

- Low-level execution policies: how do we continue to adapt to lower-level policies without physics loss?

- Elementary task policies: How do we achieve generality?
- Supervisory Policy: How do we form the primitives?

Some answers from Jan's group:

- Q: How do we parse demonstrations into sequences of primitives?

  Solution: probabilistic parsing into primitive libraries (use EM at the core, end up with a good task segmenter).

- Q: How to improve primitive selection via RL?

  Solution: relative entropy policy search (REPS) [37]. Showed a video of a robot playing ping pong (pretty well)! Very cool. Then showed a video of a robot playing a game like curling. Neat!

Q for Jan: What should we gain from deep RL, specifically in regard to policy-gradients and actor-critic methods? Is there any new insight or is it all just incorporating better function approximators?

Jan: I haven't seen much that's very different from the classic approaches.

7.8 Contributed Talks

Next up we we have the contributed talks, starting with Mel and the DAQN.

7.8.1 Deep Abstract Q-Networks

Motivation: DQN does really well for short horizon policies, but struggles on long horizon domains [36].

How can we fix this? A couple folks have tried:

- Intrinsic motivation and pseudo-counts Bellemare et al. [7]
- Hierarhical Methods [33].
- Model-based methods like R-Max, but don't really work in Atari.

Their approach: combine the benefits of model-based RL and DQN.

DAQN:

- Provide an abstraction function.
- DQNs transition between abstract states.
- Expected to learn: $\mathcal{T}_\phi$, π, π_ϕ.
- Shortens horizon of low-level learners.

Evaluation: Toy Montezuma's Revenge. Big discrete grid-world with traps/doors/keys.

Result: DAQN explores lots of rooms in the Toy MR. Previous methods (DQN, intrinsic motivation) only explore a few rooms, but the DAQN explores everything.

7.8.2 Hierarchical Multi-Agent Deep RL

Motivation: Distributed planning, as in traffic control.

Task they consider: implement a dialog agent for scheduling a trip. That is, their doing ordered or partially ordered planning. They propose a new architecture well tuned for this setting.

7.8.3 Master-Slave Communication for Multi-Agent Deep RL

Open Question in Multi-Agent RL: how to facilitate effective communication? How can we plan in a globally optimal way?

Introduce a new Master-Slave architecture for addressing this issue, run it on some multi-agent tasks, including a Starcraft micro task where two groups of marines fight against each other.

7.9 Invited Talk: Emma Brunskill on Sample Efficiency in Hierarchical RL

Goal: faster learning and better sample efficiency with hierarchy.

Emma thinks we really don't have a sense of when/why/how hierarchies help. Yeah!

Today: when learning from the *past*. Neat!

Emma thinks a lot about RL systems that interact with people. One advantage of this setting for RL is that there's lots of prior data available! Education, healthcare, customers are all domains that produce rich and large datasets. This also means we're doing Batch RL (entirely offline).

Scenario of interest: Interventions in classrooms. One major challenge here is generalizing to untried policies.

Problem: batch data policy evaluation. Game called "refraction" developed by Zoran Popovic and colleagues [39].

One solution: Per-Decision Importance Sampling (PDIS) [40]. It's unbiased but high variance. To overcome the high bias, let's try using options to reduce variance: paper at NIPS this year [28].

Theorem 7.1. *(From Guo et al. [28].) Variance of PDIS is exponential in H in the worst case:*

$$Var[PDIS(D)] = \Omega(2^H) \tag{17}$$

First idea: do PDIS for options policies. Given some data generated by π_b, want to evaluate π_e. In this setting the importance weights are over the meta-policy (over options), not over the primitive policy. We might also imagine that we only tweak some aspects of the options; in this setting, the variance can only be effected by at most the horizon of the options that changed. Using these insights they reduce the variance of PDIS. Cool!

Second idea: options and bottlenecks. Key insight is that there are many ways to get to the same thing. Could have a variety of different options but the same set of terminal states. We can exploit this structure to again get some benefits. They partition trajectories according to choke points, or places where they know they have the same termination states. Allows for the PDIS estimate to go from a huge product into a sum over smaller products. Again they see a huge reduction in the variance of PDIS.

7.10 Invited Talk: Matt Botvonick on Information Bottleneck in HRL

Matt called this "A tale of two bottlenecks".

Q: Can we score which options are best?

A: Yes! Let's take a Bayesian model-selection perspective and do this [46].

Claim one: Bottlenecks structure action and structure relevant information.

The information piece is based on Dijk et al. [15]. From an information theoretic perspective: to what extent does my initial action correlate with the goal I'm pursuing. That is, what's the mutual information between my action and the goal I'm pursuing, given state:

$$I(A; G \mid S) \tag{18}$$

Takeaway: information theoretic quantities reveal bottlenecks.

Next Bottleneck is based on Tishby's information bottleneck method [50].

Want to satisfy two desiderata.

1. First, maximize relevant info, minimize irrelevant info:

$$R_{IB}(\theta) = I(X, Y; \theta) - \beta I(Z, X; \theta) \tag{19}$$

2. Penalize latent code for departing from the true distribution

$$J_{IB} = \sum_{i=1}^{n} \mathbb{E}_{\varepsilon \sim p(\varepsilon)} \left[-\log q(y_n \mid f(x_n, \varepsilon)) \right] + \beta KL \left[p(Z \mid x_n), r(Z) \right] \tag{20}$$

Bringing the two bottlenecks together. Let's place a bound on the mutual information of relevance:

$$I(A; G \mid S) \leq I(Z; G \mid S) \tag{21}$$

In practice: sample a goal, sample a trajectory of actions, penalize encoder for departures from prior.

Some thoughts on neuro-science: all of this sounds a lot like automatic and controlled processing. Did a quick demo of the Stroop Test Bench et al. [8], where you have to read the color of words that differ from the color described by the word itself.

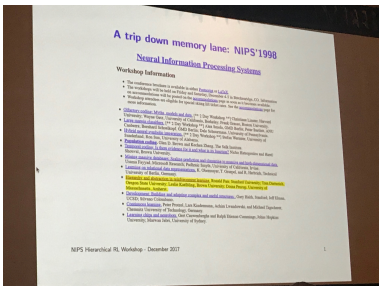


Figure 4: NIPS HRL Workshop in 1998

7.11 Invited Talk: Doina Precup on Progress in Deep Temporal RL

Started chatting about a workshop held in 1998 at NIPS on Hierarchical RL:
New Perspective: Think of an Option as a generic behavioral *program*. This suggests call-and-return execution, recursion, variables, and so on.

Frontier: what is the right set of subgoals or options?

Their motivation: achieve continual lifelong learning.

Q: Is it possible to solve a problem and learn good options for transfer at the same time? Some of their earlier work introduces the Option-Critic Architecture Bacon et al. [3]. The Option-Critic introduces a new gradient update for learning options. It consists of two components: (1) Each option should use primitive actions that are better, and (2) Find good termination conditions (lengthen when the option is good, terminate when it's bad). A third term also encourages the meta-policy to take better options.

Sidenote: Doina said that initiation conditions are a bother and we should get rid of them anyways.

How about regularization? Might be useful to encourage diversity in the options:

- Length collapse: options dissolve into primitive actions over time
- Assumption: Executing a policy is cheap, deciding what to do is expensive. Dave: Really? I'd think the opposite. Real world action is expensive. Simulation/thinking is cheap.
- Deliberation is also expensive in animals due to energy costs and missed opportunity costs.
- So: Agent should be penalized for choosing among the options more-so than choosing among primitive actions.

They choose to add a deliberation cost to switching between options. This deliberation cost induces a Value function (has a corresponding Bellman Equation). This presents a *regularized* objective that doesn't deliberate too long while still trying to maximize reward:

$$\max_{\Omega} \mathbb{E}\left[Q_{\Omega}(s,\omega) + \tau Q_c(s,\omega)\right] \tag{22}$$

Future work:

- More empirical work in option construction.
- Leveraging option models.
- Tighter integration with Neural Turing Machine [26] and similar models.

Challenge Question: Does deep HRL have a reproducibility issue? Is the work rigorous enough? Is it reproducible enough? What steps forward can/should we take in this direction?

Doina Answer: It's more difficult to reproduce results in HRL in some cases because the amount of computation/resources we have makes it hard to reproduce. We have the potential for doing this successfully because the code and domains are available. Those of us who are in a position to have a large amount of computational infrastructure should make it available to others.

7.12 Panel: Doina, Jurgen, Matt, David, Jan, Marcos

Q: Why hierarchies?

- Jurgen – connected to all aspects of intelligence
- Matt – Looked at it from the perspective of neuroscience, seems essential to biological intelligence
- Doina – optimistic that we're making progress.
- Jan – It's necessary! Can't have intelligent robots without it.
- Marcos – works in industry applying the techniques of HRL, seems to work really worl for a lot of applications.
- David – Interested in building general purpose RL agents that can solve challenging problems. If we care about solving challenging problems, we need hierarchies.

Q: What's left in HRL?

- Matt – Craving an understanding of hierarchy that accommodates *shared structure*. Spreading PB vs. Jam, but also waxing a floor, mustard on a hotdog. Wants to see something like a smooth combination of Options.

Q: For David Silver – Is it surprising that Chess/Go algorithms don't require hierarchies?

- David – Well, the time horizon is pretty small in these games. It forms abstraction hierarchies over state, but doing so over time probably wouldn't matter.

Q: There's a criticism that hierarchies are just another trick to make neural networks work (like convolution) – is it a trick? Or is it something more?

- Doina – hierarchical modeling of the world is quite different. We want models that take big steps over time in order to do efficient planning. If you really want to solve problems with many many steps, you need a good model. Right now, your model isn't just a choice of network hierarchy.

- Jan – some problems come with explicit hierarchical structure, like robotics.
- David – well, if you move to continuous time, you *have* to pick a time scale at which you reason. In this regime it's obvious this is necessary.
- Jurgen – hierarchies are one of a million algorithmic types that can be relied on in your learning. Ultimately about exploiting your algorithmic regularities.
- Matt – I don't know what "trick" means. Like convolution is just exploiting structure in the world, so to is hierarchy making a similar structural assumption. Dave: Love this point!

Q: For Marcos, what do you see as real world problems that require HRL?

- Marcos – In the short term, everything we do in the industrial setting.
- Doina – I'm going to be bold. If we can induce behavioral programs from data, then we have automated programs. If we can solve this problem, we have general purpose intelligence.
- Jurgen – Responding to Doina. Well, LSTMs/RNNs can already do that. Really optimistic we'll solve lots of open problems relatively soon. At some point said: "Babies are less fragile than robots", which I think was in reference to a baby's ability to adapt and recover from mistakes.
- Jann – Let me be more pessimistic. Our biggest limitation is our hardware. Robots are made to be programmed and controlled. The more interesting question is what can we learn when we have better robot learning algorithms.

Q: When we look at brains we look at plausibility. What kind of advice would you have for someone coming to HRL from comp. neuro or cogsci. And second: which areas of neuroscience do you think are ripe of translation to HRL?

- Matt – Neuro has taken RL on board. Hierarchy only becomes relevant as problems become complex. Thus, research question: we need to focus on complex problems. Some of the most important problems questions in neuro are also about hierarchies.

Q: Lots of paradigms for hierarchy. Any hope that we can get one that wins? Or one that maps onto what humans do?

- Matt – Sure! Folks in neuro have done this, like Michael Frank at Brown. I think Hierarchy in the wild faces us with challenges that we tend to avoid in AI. We tend to talk about tasks. Humans/animals don't have talks. Once we grow in which the agents live, we'll get at the question of what it means to decompose behavior.

Q: Do you think research in HRL will involve models/planners, or will be more model-free?

- David – We should be trying everything! We'd be really speculating about what things will look like. We should not put all of our eggs in one basket.

- Doina – we should try tons of different things! Maybe we're not doing modeling right? So we need break throughs in how to learn models (with or without models). Maybe we need partial or local models. We've all focused a lot on performance, but there's another aspect which is communication. We can quickly specify goals in the abstract.
- Marcos – communication can also provide transparency and insight into the decision making.

Q: In imitation learning, we don't provide the hierarchy. How do you know that you have the right hierarchy? How do you deal with providing the right demonstrations for hierarchies?

- Jan – In robotics we start bottom up. Move from motion controls to something higher. These days, when we go a layer up, we actually try to learn grammars for behavior. When we can get such grammars, and ones that make sense to a human being, we'll be in a great position.

Q: What are some areas of research that are under explored? (that are relevant/related to HRL)

- Doina – exploration! And how to do model-learning in an efficient way. We don't yet make the distinction between models that predict agent behavior vs. models that predict world behavior.
- Jurgen – here's an idea from a paper in the late 90s. RL algorithms that compete with one another (left brain and right brain). They cooperate on building an experimental protocol to execute. However, they disagree on the outcomes. They bet against each other in a zero sum game.
- Matt – also model-learning! And in particular, rapid model-learning.
- David – still a lot of inspiration we can draw from classical planning. Idea space is infinitely large but infinitely dense; lots of great ideas out there!
- Marlos – reward specification.

Q: Benchmarks for HRL? What games exist?

- Doina – we shouldn't make benchmarks specifically for HRL. We want to solve the real problems out there, so let's set our sights on those. Some already out there like Atari, Starcraft, Robotic control.

References

[1] Kareem Amin, Nan Jiang, and Satinder Singh. Repeated inverse reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1813–1822, 2017.

[2] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, pages 5053–5063, 2017.

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

[4] Chris L Baker, Joshua B Tenenbaum, and Rebecca R Saxe. Goal inference as inverse planning. In *Proceedings of the Cognitive Science Society*, volume 29, 2007.

[5] EA Baltz, E Trask, M Binderbauer, M Dikovsky, H Gota, R Mendoza, JC Platt, and PF Riley. Achievement of sustained net plasma heating in a fusion experiment with the optometrist algorithm. *Scientific Reports*, 7(1):6425, 2017.

[6] Peter W Battaglia, Jessica B Hamrick, and Joshua B Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45): 18327–18332, 2013.

[7] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1471–1479, 2016.

[8] ClJ Bench, CD Frith, PM Grasby, KJ Friston, E Paulesu, RSJ Frackowiak, and RJ Dolan. Investigations of the functional anatomy of attention using the stroop test. *Neuropsychologia*, 31(9):907–922, 1993.

[9] Siavash Arjomand Bigdeli, Matthias Zwicker, Paolo Favaro, and Meiguang Jin. Deep mean-shift priors for image restoration. In *Advances in Neural Information Processing Systems*, pages 763–772, 2017.

[10] Noam Brown and Tuomas Sandholm. Safe and nested subgame solving for imperfect-information games. *arXiv preprint arXiv:1705.02955*, 2017.

[11] Thang D Bui, Josiah Yan, and Richard E Turner. A unifying framework for gaussian process pseudo-point approximations using power expectation propagation. *Journal of Machine Learning Research*, 18(104):1–72, 2017.

[12] Andreas Damianou. *Deep Gaussian processes and variational propagation of uncertainty*. PhD thesis, University of Sheffield, 2015.

[13] Christoph Dann, Tor Lattimore, and Emma Brunskill. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 5711–5721, 2017.

[14] Samuel Daulton, Taylor Killian, Finale Doshi-Velez, and George Konidaris. Robust and efficient transfer learning with hidden parameter markov decision processes. In *Advances in Neural Information Processing Systems*, pages 6245–6250, 2017.

[15] Sander G van Dijk, Daniel Polani, and Chrystopher L Nehaniv. What do you want to do today?: Relevant-information bookkeeping in goal-oriented behaviour. *Procs of Artificial Life XII*, 2010.

[16] Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl˜2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

[17] Yan Duan, Marcin Andrychowicz, Bradly Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. *arXiv preprint arXiv:1703.07326*, 2017.

[18] Matthew M Dunlop, Mark Girolami, Andrew M Stuart, and Aretha L Teckentrup. How deep are deep gaussian processes? *arXiv preprint arXiv:1711.11280*, 2017.

[19] Chelsea Finn and Sergey Levine. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. *arXiv preprint arXiv:1710.11622*, 2017.

[20] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

[21] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. *arXiv preprint arXiv:1709.04905*, 2017.

[22] Carlos Florensa, Yan Duan, and Pieter Abbeel. Stochastic neural networks for hierarchical reinforcement learning. *arXiv preprint arXiv:1704.03012*, 2017.

[23] Ronan Fruit, Matteo Pirotta, Alessandro Lazaric, and Emma Brunskill. Regret minimization in mdps with options without prior knowledge. In *Advances in Neural Information Processing Systems*, pages 3168–3178, 2017.

[24] Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521 (7553):452–459, 2015.

[25] Andrew Gibiansky. Deep voice 2: Multi-speaker neural text-to-speech. In *Advances in Neural Information Processing Systems*, pages 2966–2974, 2017.

[26] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[27] Rachid Guerraoui, Hadrien Hendrikx, Alexandre Maurer, et al. Dynamic safe interruptibility for decentralized multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 129–139, 2017.

[28] Zhaohan Guo, Philip S Thomas, and Emma Brunskill. Using options and covariance testing for long horizon off-policy policy evaluation. In *Advances in Neural Information Processing Systems*, pages 2489–2498, 2017.

[29] Dylan Hadfield-Menell, Smitha Milli, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Inverse reward design. In *Advances in Neural Information Processing Systems*, pages 6749–6758, 2017.

[30] Anna Harutyunyan, Peter Vrancx, Pierre-Luc Bacon, Doina Precup, and Ann Nowe. Learning with options that terminate off-policy. *arXiv preprint arXiv:1711.03817*, 2017.

[31] Emilie Kaufmann and Wouter Koolen. Monte-carlo tree search by best arm identification. In *Advances in Neural Information Processing Systems*, 2017.

[32] Tejas D Kulkarni, Pushmeet Kohli, Joshua B Tenenbaum, and Vikash Mansinghka. Picture: A probabilistic programming language for scene perception. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 4390–4399, 2015.

[33] 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.

[34] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[35] Marlos C Machado, Marc G Bellemare, and Michael Bowling. A laplacian framework for option discovery in reinforcement learning. *arXiv preprint arXiv:1703.00956*, 2017.

[36] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[37] Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *AAAI*, pages 1607–1612. Atlanta, 2010.

[38] John C Platt, J Pritchard, and Drew Bryant. Analyzing energy technologies and policies using doscoe. 2017.

[39] Oleksandr Polozov, Eleanor O'Rourke, Adam M Smith, Luke Zettlemoyer, Sumit Gulwani, and Zoran Popovic. Personalized mathematical word problem generation. In *IJCAI*, pages 381–388, 2015.

[40] Doina Precup, Richard S Sutton, and Sanjoy Dasgupta. Off-policy temporal-difference learning with function approximation. In *ICML*, pages 417–424, 2001.

[41] Rajesh Ranganath, Adler Perotte, Noémie Elhadad, and David Blei. Deep survival analysis. *arXiv preprint arXiv:1608.02158*, 2016.

[42] Andrew P Sabelhaus, Jonathan Bruce, Ken Caluwaerts, Yangxin Chen, Dizhou Lu, Yuejia Liu, Adrian K Agogino, Vytas SunSpiral, and Alice M Agogino. Hardware design and testing of superball, a modular tensegrity robot. 2014.

[43] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3857–3867, 2017.

[44] Adam Santoro, David Raposo, David GT Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. *arXiv preprint arXiv:1706.01427*, 2017.

[45] Jürgen Schmidhuber. Powerplay: Training an increasingly general problem solver by continually searching for the simplest still unsolvable problem. *Frontiers in psychology*, 4, 2013.

[46] Alec Solway, Carlos Diuk, Natalia Córdova, Debbie Yee, Andrew G Barto, Yael Niv, and Matthew M Botvinick. Optimal behavioral hierarchy. *PLoS computational biology*, 10(8): e1003779, 2014.

[47] Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miro Dudik, John Langford, Damien Jose, and Imed Zitouni. Off-policy evaluation for slate recommendation. In *Advances in Neural Information Processing Systems*, pages 3634–3644, 2017.

[48] Yee Teh, Victor Bapst, Razvan Pascanu, Nicolas Heess, John Quan, James Kirkpatrick, Wojciech M Czarnecki, and Raia Hadsell. Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 4497–4507, 2017.

[49] Yuandong Tian, Qucheng Gong, Wenling Shang, Yuxin Wu, and C Lawrence Zitnick. Elf: An extensive, lightweight and flexible research platform for real-time strategy games. In *Advances in Neural Information Processing Systems*, pages 2656–2666, 2017.

[50] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[51] Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. *arXiv preprint arXiv:1703.01161*, 2017.

[52] Yilun Wang and Michal Kosinski. Deep neural networks are more accurate than humans at detecting sexual orientation from facial images. 2017.

[53] Felix Warneken and Michael Tomasello. Altruistic helping in human infants and young chimpanzees. *science*, 311(5765):1301–1303, 2006.

[54] Théophane Weber, Sébastien Racanière, David P Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adria Puigdomènech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, et al. Imagination-augmented agents for deep reinforcement learning. *arXiv preprint arXiv:1707.06203*, 2017.

[55] Jiajun Wu, Erika Lu, Pushmeet Kohli, Bill Freeman, and Josh Tenenbaum. Learning to see physics via visual de-animation. In *Advances in Neural Information Processing Systems*, pages 152–163, 2017.

[56] Jiajun Wu, Joshua B Tenenbaum, and Pushmeet Kohli. Neural scene de-rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 699–707, 2017.

[57] Yuhuai Wu, Elman Mansimov, Shun Liao, Roger Grosse, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. *arXiv preprint arXiv:1708.05144*, 2017.