AAAI 2019 Notes
Honolulu, HI, USA

David Abel*
david_abel@brown.edu

January & February 2019

Contents

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

This document contains notes I took during the events I managed to make it to at AAAI in Honolulu, Hawaii, USA, including sessions of the Doctoral Consortium. 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

AAAI was fantastic – the invited talks offered impressive videos, inspiring visions of the future, and excellent coverage of many areas, spanning game playing, learning, human-robot interaction, data management, and exciting applications. I also enjoyed the two evening events: 1) the 20 year roadmap for AI research in the US, and 2) the debate on the future of AI. Both events raised compelling questions for researchers and practitioners of AI alike.

I also want to highlight the doctoral consortium (DC). This was the first DC I have participated in; in short, I strongly encourage grad students to do at least one DC during their program. You will get exposure to some fantastic work being done by peers all around the world and receive tailored mentorship on your presentation skills, how you write, and your research objectives and methods more generally.

AAAI really struck me as doing a great job of mixing together many subfields that don't often spend as much time talking to one another – I met plenty of folks working in planning, constraint satisfication, automated theorem proving, AI and society, and lots of ML/RL researchers.

A final point that was raised at the roadmap – naturally, a huge fraction of research/industry is concentrated on ML at the moment. But, it's important that we continue to push the frontiers of knowledge forward across many different areas. So, if you're considering going into grad school soon, do consider pursuing other topics/areas that offer fundamental and important questions (of which there are many!) beyond ML.

And that's that! Let's dive in.

2 Sunday January 27th: Doctoral Consortium

It begins! Today I'll be at the Doctoral Consortium (DC) – my goal with the notes is both to give folks a sense of what a DC entails, and to share the exciting research of some great grad students.

2.1 Overview of the DC

I *highly* recommend doing a doctoral consortium at some point during grad school. I learned a huge amount from the experience –
For those that don't know, a DC involves preparing a short abstract summarizing your work, and giving a 10-20 minute presentation to your peers and their montors. Each student participating is assigned a mentor (from their area) that helps with preparing your presentation and gives you more general advice on your research.

It was a great experience! I had the pleasure of meeting many wonderful grad students and hearing about their work.

2.2 Neeti Pokhriyal: Multi-View Learning From Disparate Sources for Poverty Mapping

Focus: Learning from multiple disparate data sources, applied to sustainability and biometrics.

Specific Application: Povert mapping. Spatial representation of economic deprivations for a country. A major tool for policy planners.

Current method is a household survey, which is 1) costly, 2) time consuming, 3) only available for small samples.

Research Goal: Get accurate, spatially detailed and diagnostic poverty maps for a country.

Lots of data available via weather, street maps, economic data, mobile phones, satellite imagery. But! Each of these data sources are structured very different.

> **Definition 1** (Multi-View Learning): *A style of learning takes as input separate, semantically distinct kinds of data, and brings them together into a factorized representation for use in predictive models.*

Method: learn a Gaussian Process (GP) Regression model combined with elastic net regularia-tion [48].

Using this model yields the map pictured in Figure 1. Then perform quantitive analysis and validates that their model is making high quality predictions by comparing to census data.

Objective 2: learn a factorized representation from multiple data sources. The hope is that we can disentangle explanatory factors that are unique to each data source.

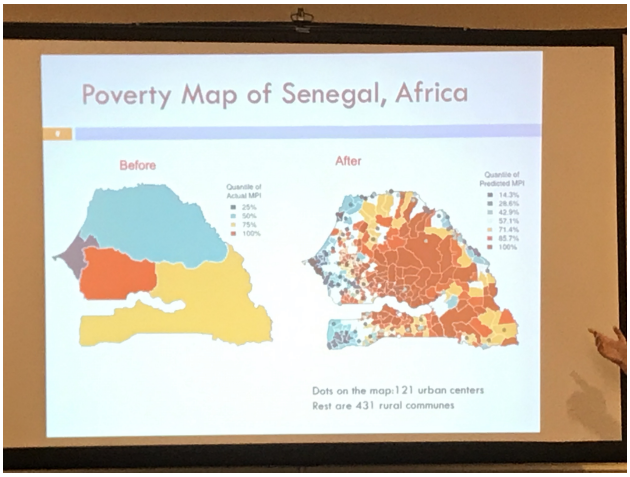


Figure 1: Higher fidelity poverty prediction

Sort of an EM like approach:

1. *Learning Step:* MAp views Y and Z to shared subspaces $X_i, \ldots$.
2. *Inference Step:* Perform inference on these subspaces.

Q: Main question, then: how do we learn the shared subspace?

A: Separate data belonging to different class across different views is maximized, while ensuring alignment of projects from each view to the shared space. Can be solved using a Generalized Eigenvalue problem, or using the kernel trick.

.....

2.3 Negar Hassanpour: Counterfactual Reasoning for Causal Effect Estimation

Problem: Consider Mr. Smith, who has a disease and some known properties (age, BMI, etc.). Doctor provides treatment X and observes the effect of treatment X (but does *not* get data about the counterfactul: what would have happened if doc had applied treatment Y?).

Goal: Estimate the "Individual Treatment Effects" (ITE) – how does treatment X compare to Y?

Datasets:

- Randomized Controlled Trial (RCT): See lots of both X and Y. But, it's expensive (lots of trials) and unethical (giving placebos when you know the right treatment).
- Observational Study: provide the preferred treatment. But, sample selection bias.

  Example: Treating heart disease, a doc prescribes surgery to younger patients and medication to older patients. Compare survival time – but, clear bias in who gets what treatment.

This is a really fundamental problem called "sample selection bias" – rich patients receiving expensive treatment vs. poor patients receiving cheap treatment and so on.

Overview of this work:

- Generate realistic synthetic datasets for evaluating these methods (since good data is hard to come by)

  → Take an RCT and augment it with synthetic data.

- Use representation learning to reduce sample selection bias.

  → Want $Pr(\phi(x) \mid t = 0) \approx Pr(\phi(x) \mid t = 1)$ to be similar, with ϕ the learned representation and t the treatment.

- Learn underlying causal mechanism with generative models.

  → Learn causal relationships between treatments and outcomes by using generative models. Can we identify the latent sources of outcome from observational dataset?

- Perform survival predictions.

  → Can we predict outcomes that are censored or take place after studies end?

- Going beyond binary treatments

  → Many, but not all, treatments are binary. Can we go beyond this to categorical or real valued treatments?

- Providing a course of treatment

  → Call on reinforcement learning.

.....

2.4 Khimya Khetarpal: Learning Temporal Abstraction Across Action & Perception

Q: How should an AI agent efficiently represent, learn, and use knowledge of the world?

A: Let's use temporal abstractions!

Example: preparing breakfast. Lots of subtasks/activities involved like (high level): choose eggs, type of toast (mid level) chop vegetables, get butter, and (low level) wrist and arm movements.

Definition 2 (Options [37]): *An option formalizes a skill/temporally extended action as a triple: $\langle, I, \beta, \pi\rangle$, where $I \subseteq S$ is a initiation set, $\beta : S \to \Pr(S)$ is a termination probability, and $\pi : S \to A$ is a policy.*

Example: A robot navigates through a house between two rooms. To do so, it has to open a door. We let I denote the states where the door is closed, β is 1 when the door is open and 0 otherwise, and π opens the door. Then, this option defines the "open the door" skill.

Main Question: Can we learn useful temporal abstractions?

Hypothesis: Learning options which are specialized in situations of specific interest can be used to get the right temporal abstractions.

Motivation: AI agents should be able to learn and develop skills continually, hierarchically, and incrementally over time.

So, imagine we had a house decomposed into different rooms. Then we would like to learn skills that take the agent between each room. Further, the agent should be able to transfer for one agent to another.

Objective 1: Learn options and interest functions simultaneously.

New idea: break the option-critic assumption [2] that $I = S$. Instead, consider an interest function:

> **Definition 3** (Interest Function): *An interest function is an indication of the extent to which an option is interested in state s.*

Now learn a policy over options and an interest function – we can jointly optimized over both things. Derive the policy gradient theorem for interest functions, intra-option policy, and the termination function.

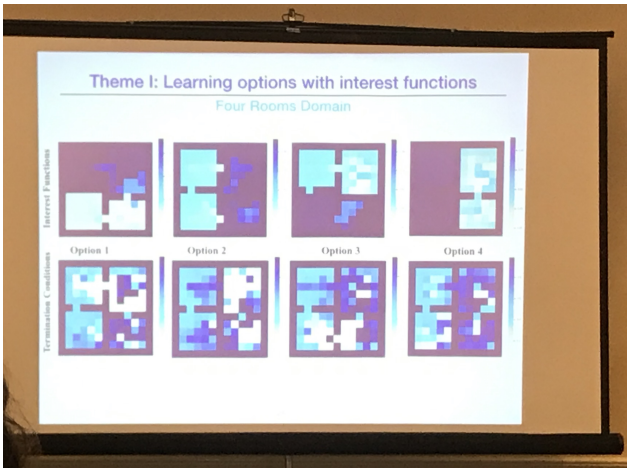


Figure 2: Learned interest functions

Also explore learning interest functions in continuous control tasks, showing nice separation between the learn options.

Objective 2: Consider a never-ending stream of perceptual data. We'd like to learn a stream of percepts and behavior over time.

Challenges:

- How can we the agent automatically learn features which are meaningful pseudo rewards?
- Where to task descriptions come from?
- How can we achieve the most general options without hand designing tasks/rewards?
- Evaluation in a lifelong learning task? Benchmarks?

.....

2.5 Ana Valeria Gonzalez-Garduo: RL for Low Resource Dialogue Systems

Goal 1: Create more informed approaches to dialogue generation.

Goal 2: Use RL for domain adaption in goal oriented dialogue.

(And: can we do this in a language agnostic way? So, introduce models that can work with any/many languages).

Dialogue systems are divided into two subfields:

1. *Open ended dialogue generation:* typically use encoder-decoder architectures
2. *Goal oriented dialogue:* predominantly tackle using "pipeline" methods. So, automatic speech recognition unit, then an understanding unit, and so on.

Current Focus: "state tracking". That is, state tracking deals with inferring the user intent or belief state during the conversation.

But, limitation: intents usually rely on a particular ontology that defines which intents are valid.

Current Status of the Project: Bridge the gap inb goal oriented dialogue. Main goal: can we get rid of the need for annotations?

General idea: given a bot's utterance ("how can I help?"), and a user response ("I want to change payment methods"), we want to find a relevant query from prior conversations to identify what the user said. Or really, use it to condition the decoder.

Result: this model works very well! On BLEU their model performs favorably, but more importantly, on a human evaluation, their responses were consistently chosen over the baseline.

Q: But, what if our domain is not in the pool of relevant conversations?

A: Work in progress! Idea $\rightarrow$ Use RL:

1. Phase 1: Use existing data for state tracking, pretrain models in a supervised manner.

   → Turn level supervision, slots and values represented using word embeddings.

2. Phase 2: Use RL to finetune pretrained model.

   → Rely on dialogue level supervision (joint goal accuracy) as reward. So, how many slot-values ("Food-Mexican, Price-Cheap"), to determine the reward.

Challenges in using RL for state tracking: dialogue is long (credit assignment is hard!), sample efficiency, might be able to leverage curriculum learning.

Main Future Direction: Enable dialogue state transition model to generate new unseen slots.

.....

2.6 AAAI Tutorial: Eugene Freuder on How to Give a Talk

Start with an example! Or a counter example.

These are just his conclusions! So decide for yourself, of course.

This talk is not intended to be mean spirited – he'll be talking about mistakes people make.

Meta-message: presenting a talk is a skill that can be studied and practiced! And it's worth doing – spend years researching and 10 minutes presenting. The 10 minutes should be polished.

Six Points:

1. Convey Enthusiasm
2. Make it Easy to follow
3. Employ examples
4. Expressive
5. Enhance your presentation with visuals/dynamic material
6. Engage the audience

2.6.1 Enthusiasm

The secret of a good talk: **Enthusiasm!**

→ If you're not enthusiastic about your work, how do you expect anyone else to be?

Fear of public speaking: "glausophobia" – ranked as the most common fear in the USA (more so than spiders/death).

Q: How do you get over this fear?

A: Remember the audience is on your side! Breathe. Drink water.

Tricks:

- Look over their heads (instead of in their faces– can be easier).
- Or, turn it into an individual conversation, or a bunch of individual conversations.
- Science is fun! So have fun.

Sometimes it feels like there's the me giving the talk and the me monitoring me giving the talk. Unfortunate, potentially, as I'm then not present.

It is **really** hard to be too enthusiastic. The speaker is standing on a chair to demonstrate enthusiasm –

2.6.2 Easy to Follow

One major goal of the talk: get people to read and build on your work. Details are in the paper– job in the talk is to get them to read the paper.

KISS principle: Keep It Simple Stupid!

Story from Feynman's lost lecture: someone asked Feynman to prepare a lecture on some complicated physics concept related to particle spin. Feynman said he would go off and work on it for a few days and come back and give a lecture: "I'll be able to give a freshmen level lecture in a few days!". But then he came back: "Okay, I couldn't do it. I couldn't turn it into a freshmen level lecture. *Which means we don't yet understand it yet.*"

Audience doesn't distinguish hard work from the researcher having a hard time explaining it.

Let them see the forest – not the trees.

Explain math/formalisms with visuals and metaphor. Minimize definitions, don't overestimate audience.

People make talks too difficult by going through material too quickly, or trying to force too much in. Time yourself!

2.6.3 Use Examples

Start with an example! Even before the title/introduction. Dave: Hm! Bold move. Hard to do this in a conference, to me. But it's a neat idea.

He played a clip from a TED talk (Niri G?) where the speaker opened just with "Two twin domes...". It was engaging.

Even if you don't start with one – use one. It can be hard to make the example simple enough.

Make your example(s):

- Illustrate what you've done
- Simple
- Concrete
- Only add complexity later (or use a running example!)

2.6.4 Be Expressive

Use your voice and body language to be expressive.

Do's and Dont's:

Do: smile (he played Nat King Cole!), listen to yourself beforehand (and look for "ums", body swaying, etc), try to make it a conversation (not a declamation), vary your voice (louder/softer, higher/lower), pause occasionally, articulate, look people in the eye, turn off your phone, silence computer.

Don't: Read from a prompt, speak in a monotone, speak too fast or too slow, be distracting (like playing with your hair, rocking back and forth or front to back → a good alternative is to hang onto the lectern), say "um"/"uh" too much, mumble, fidget, turn around a face the screen, look at your laptop too much.

2.6.5 Enhance your talk with visuals/dynamics

Do's and Dont's:

Do: use visuals, minimize text, keep notes for yourself, remember short term memory of audience (repeat details, highlight big things, etc.), pull out what is important.

Don't: use bulleted lists.

2.6.6 Engage the Audience

Q: How do you engage an audience directly?

A: Ask questions!

Consider why your audience is here: they want to hear what you did. Tell them right away what you're going to do. Big results at the beginning.

Can turn your talk into a story:

- A problem to be solved. Someone came to me in a company with a problem, and so on.
- Could be the "2x2 matrix" story: folks have done Red things and Blue things, and folks have done Small things and Big things, but no one has done Big Red things! I'm going to do that.
- Could contradict conventional wisdom
- The "journey" and not just the endpoint.

Make your talk fun! "A spoonful of sugar makes the medicine go down." Use gimmicks, like: playing songs, use props, video, etc.

Hypothesis: on average, at a CS conference, by the time you reach the halfway point, at least half the people would be tuned out.

→ So: if the hypothesis is even close to being right, think of the waste!

.....

2.7 Aida Rahmattalabi: Robust Peer-Monitoring on Graphs

Problem: Suicide is a critical public health problem in the US. Second leading cause of death among students.

One approach: gatekeeper training (suicide prevention program). Can identify warning signs, but can only train limited number of individuals.

Main Goal: Improve gate keeper training with social network information by taking into account characteristics of individals in the population.

Technical problem: optimize:

$$\max_{x,y} \sum_{n \in N} y_n, \tag{1}$$

subject to $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, which define the feasible sets of possible choices of trainers.

Basically: social characteristics are very important in assigning a gatekeeper. So, impose constraints on who is chosen as a gatekepeer (based on things like race/gender).

But, new problem: participation uncertainty. Not all chosen trainers might not participate.

Optimization problem is formulated as:

$$\max_{x \in \mathcal{X}} \min_{\varepsilon \in E} \sum_{n \in N} y_n(x, \varepsilon), \tag{2}$$

where the x is the gatekeeper we choose, and natre is acting adversarially over choice of σ which determines participation rate. $y_n(x,)$ denotes the coverage for each of the n individuals. Can cast this as a polynomial size mixed integer program which is nice and tractable. That is, main result:

Theorem 2.1. *For a fixed vale of K, their main optimization problem (the "k adaptibility problem"), can be reformulated exactly as a polynomially sized mixed integer linear program.*

Experiments: compare to a greedy robust approach on a real social network. Measuring "coverage", which is captured by the function y above. Also evaluate according to a "Price of Fairness".

.....

2.8 Nikhil Bhargava : Multi-Agent Coordination under Uncertain Communication

Consider under water gliders: they go underwater for months at a time, so we can't actually communicate with them very often.

So, most folks usually use a "real time executive" (RTE), which is a centralized, real-time means of dispatching actions to different agents.

But: in reality, too much uncertainy in action execution. So, RTEs often include state updates based on the outcome of action executions.

Lots of prior work on RTEs that can handle highly expressive plans, can adapt to uncertainty, etc.

But back to our gliders: lots of different autonomous agents with uncertainty in actions and communications.

Goal: Can we upgrade this notion of an RTE into a multi-agent system?

Three core ideas/changes to the traditional RTE model:

1. Multi-agent aware planning

   → Turn Simple Temporal Networks With Uncertainty (STNUs) into DTNUs/POSTNUs, which is analogous to turning MDPs into DecPOMDPs, POMDPs.

2. Action dispatches and communication requirements.

3. Change "immediate state updates" to delayed and noisy state updates.

   → Can improve controllability by shortening window to hear about events or learning information sooner.

Q: Are there known good approximations for DTNUs or POSTNUs?

3 Monday January 28th: Doctoral Consortium

Onto day two! Today I'll again be at the Doctoral Consortium.

3.1 Sandhya Saisuramanian: Adaptive Modeling for Risk-Aware Decision Making

Agents: commonly use "reduced models" – a simplified model of the world.

Simplified for: 1) tractability, 2) unavailable information.

To simplify the world, we could change the *state* space or the *action* space – in this work, we'll focus on restricting the action outcomes.

Drawbacks of planning with reduced models:

1. Over optimistic
2. Sub-optimal action selection
3. Excessive replanning
4. Unreachable goal(s) from some/all states

Reduced model literature: Improving Planning time (FF, FF-replan) [42], bounded number of exceptions [30].

Q: How do we choose the right reduced model?

A: it's hard! 1) Representation is problem specific, 2) trade-off in simplicity/risk, 3) Hard to deal with incomplete information.

Thesis: Investigate how to improve risk awareness by taking into account the complexity of planning under uncertainty.

Current Focus:

1. Improving model fidelity to address over optimism

   → Main Idea: selectively improve model fidelity by accounting for risky outcomes in selected states. Do so by setting a threshold based on amortizing risk.

2. Replanning in critical states

One idea: "determinize" (s, a) pairs by ignoring some of their stochastic outcomes. This leads to a simpler model. Conduct experiments showing the impact of different thresholds on when to determinize, showing a decrease in risk.

In the future, hope to extend these ideas to settings with incomplete information.

Q: Slide 24: time savings relative to what?

Q: Why use this measure for the effectiveness of a reduction? $Q-$.

.....

3.2 Abhinav Verma: Interpretable and Verifiable RL through Program Synthesis

Example: a deep RL agent (DDPG) trained a simulation of a car driving.

But, we shouldn't actually *deploy* this trained agent even it if works in simulation.

So, the goal of this work: how can we systematically uncover failures/weaknesses/strengths of the approach?

Running Example: The Open Racing Car Simulator (TORCS) – continuous control task involving driving a race car around a track. Quite complex: high dimensional input, agent controls steering/acceleration/breaks.

Goal: Change our policy representation to something more interpretable.

Q: What does that mean?

A: Previously, we use a neural network for a policy. Now, we'll instead propose a programmatic policy. Gives us access to more logical/symbolic kinds of understanding about the agent.

Main Idea: Automatically discover expressive policies in a high level domain specific language for RL environments.

→ Method is to use DRL to find a good policy and distill it into a high level program.

Main Benefits: 1) Interpretability, 2) Verifiability (that is, we can prove properties like robustness), and 3) Generalizability.

Challenge: Searching for a good programmatic policy is hard. The search space is non-smooth, need to do many rounds of discrete optimization.

→ Can address this challenge by defining a domain specific language so that the policy search space is much smaller (so, instead of unleashing a Turing complete language, instead look at task-specific conditions/concepts).

To handle the optimization problem, use some imitation learning.

Experiment on a transfer variant of the racing domain, find smooth policies (w.r.t. the driving of the car), and good performance on tracks on seen at training time.

.....

3.3 Ruohan Zhang: Attention – From Data to Computational Models

Goal: Understand biological attention mechanism at the behavioral and neural level.

→ If we can understand these mechanisms well enough, we can open doors to new techniques in AI/RL.

Humans have *foveated* vision: full-resolution vision in the central 1-2 visual degree of the field.

Neat example of a person playing the Atari game Freeway, showing the gaze of the person dart around the screen as they play.

Research Question 1: How do we vuild a visual attention model from eye movement data?

Collect data-sets of: 1) a person playing some Atari games, 2) a person walking outside on rough terrain with fully body motion capture, 3) virtual data driving in urban areas.

Propose a "Gaze Prediction Network" that takes as input 4 consecutive images and outputs predicted probability distribution of gaze.

Results are promising! Predicted distribution matches groun truth well.

Research Question 2: Can we use insights from attention to do imitation learning more effectively?

Simplest form of imitation learning is called "behavioral cloning", in which the learner tries to exactly match the demonstrator's behavior.

Now, the new Q: predict a human player's action given in a game frame.

Idea: use the predicted gaze to bias a network in both action prediction (in imitation learning) and in RL. In both cases the gaze helps learning across a variety of Atari games.

.....

3.4 Faraz Torabi: Imitation Learning from Observation

Example: Babies playing with each other after watching a Pixar movie (where two characters do the same!).

Research Question: In what ways can autonomous agents learn to imitate experts using visuals observations?

Main Contributions:

- A model-based algorithm for imitation from observation → + An application of the algorithm in sim-to-real transfer
- A model-based algorithm for imitation from observation → + An application of the algorithm in sim-to-real transfer

Imitation learning: learn how to make decisions by trying to imitate another agent.

Typical assumption: observations of other agent consists of state-action pairs.

→ Challenge: we don't often have state-action pairs! Often we just have observations, not states or actions (action ontology might not be the sae).

Approach 1: model based approach. Consider convential imitation learning $D_{train} = \{(s_0, a_0), \ldots\}$. But now: $D_{train} = \{(s_0), \ldots\}$.

→ Algorithm: Behavioral Cloning from Observation (BCO). Run some policy in the environment to learn an inverse dynamics model which is then used to *predict* the missing actions from D_{train}.

Experiments in Mujoco ("Ant"), demonstrator works very well. Traditional imitation learning methods work, but they have access to actions to learn from. Their approach (without actions!) performs competitively.

Next Q: Can we do Sim-To-Real transfer with BCO?

A: Yep! Great setting for imitation learning since physical robot trajectories are costly to collect, but simulation is cheap.

Final approach: model-free! Generative Adversarial Imitation from Observation (GAIfO). Experimental results are promising! Also experiment with a manipulator robot.

.....

3.5 Christabel Waylacce: Stochastic Goal Recognition Design

Point: most activities are goal oriented.

Definition 4 (Goal Recognition): *The problem of goal recognition involves identifying the goal of a given actor.*

Lots of applications – security domains! build the environment to identify dangerous actors, intelligent tutors (what is the objective of the student?) and so on.

Problem: Goal Recognition Design (GRD). Want to find behavior that communicates the goal of an agent as early as possible.

Metric for describing worst-case: "worst-case distinctiveness" (wcd) – the longest sequence of actions an agent can take without revealing its goal. Want to find changes to environments that minimizes the wcd.

Three Typical Assumptions:

1. Agents are optimal
2. Action outcomes are deterministic
3. All agents have full observability

But: lots of limitations imposed by these assumptions!

Research Question: What are the advantages and limitations of relaxing assumptions in the GRD? People are suboptimal! Lots of stochastic action outcomes. And, agents are always under partial observability.

Lots of related work that relaxes some of these assumptions, but not all [21]. This work builds on this prior literature by relaxing the determinism assumption.

Objective: GRD problem, but now minimize the *expected case distinctiveness*. Also extend this to *partially observable* case, where now actions can be stochastic and we only receive observables, not states.

.....

3.6 Satyha Ravi: Numerical Optimization to AI and Back

Consider Regularization: some method we have for preventing our learning algorithms from overfitting:

- Explicit Regularization: Constraints, penalties.
- Implicit Regularization: algorithms, priors.

This work: mostly focused on the use of *constraints*:
→ Experimental design for sparse models: well studied when function of interest is linear (can be solved with convex optimization).
Problem: D-Optimal design, with resource constraints:

$$\min_{S \subset N} \log\det\left(\sum_{i \in S} x_i x_i^T\right)^{-1}, \qquad \text{s.t. } |S| \leq B.$$

Can translate the above into a convex optimization problem. Dave: (I missed the details of what the variables above denote). Evaluate on a Neuroimaging dataset.

Next: flow problem – that is, given two images, track something about the movement of pixels between the images. Goal is to develop general purpose algorithm for all sorts of flow problems.

.....

3.7 Atena Tabakhi: Preference Elicitation for Constraint-Based Methods

Example: Smart Home Device Scheduling (SHDS). We'd like our home to automatically infer our preferences about how to manage aspects of our home (lights, temperature, etc.).

Objective: find a schedule that minimizes power consumption and discomfort of homeowners.

Induces a Weighted Constraint Satisfication Problem (WCSP):

Definition 5 (WCSP): $P =< X, D, F >$*:*

- X *set of variables*
- D *set of finite domains for each variable*
- F *set of constraints, assigns a cost to each constraint.*

Solution: an optimal assignment $\vec{x}$ *that minimizes the sum over all costs.*

Research Method 1: Interleaving search and elicitation.

First approach: use a brute force approach (BFS) to find assignments. Then, propose 3 parameterized heuristics: 1) Least Uknown Cost Elicitation (LUC), 2) Least Known Cost Elicitation (LKC), and 3) Combination (COM).

Evaluate heuristics empirically, measuring runtimes and costs averaged over 100 random graphs.

Research Method 2: Preference Elicitation in Preprocessing.

Now, modeled as a multi-agent system (multiple owners in same household). Again model this as a WCSP.

Two proposed methods for eliciting before solving the problem: Minimax regret (MR) and Maximum Standard Deviation (MS).

Further empirical evaluation to evaluate heuristics: 10 homes with 10 devices, time horizon of 6, average over 100 synthetically generated homes. Compare the heuristics vs. the random base-line (RD).

Future work: learn user preferences from imperfect feedback or uncertain user feedback.

.....

3.8 Emmanuel Johnson: Using Automated Agents to Teach Negotiation

Example: Intelligent Tutoring System. AI is good at teaching "hard" skills like math/computing. But! They aren't as effective for"softer" skills like negotiation.

In fact: most of us are bad at negotiating. 90% court cases are settled outside of court through negotiation [9]. Also important for negotiating salaries.

"Negotiation" here means the following: two people, each have a set of preferences over a set of objects. There's a way they can divide these items given these preferences.

Import distinction between value claiming and creating:

- *Value Creating*: the process of maximizing joint utility often referred to as "growing the pie".
- *Value Claiming*: the process of getting as much in the negotiation as possible.

This work: focused on providing personalized pedagogical feedback to help individuals improve negotiating.

Different negotiation principles fit nicely into value creating/claiming buckets, like not committing too early, holding ground, and so on.

Data set: conflict resolution agent tests. 156 human-agent negotiations (wizard of oz style – someone controlling the agent). There's a collection of objects on a table, the person and the bot negotiate back and forth over who gets what objects (video demo: it was awesome!).

Metrics: look at predicted outcomes of the deal based on the negotiation principles. Principles measured: good initial claim, agreement time, etc.

Pilot Study: with the wizard of Oz agent. Study the predicted quality of the negotiation, the amount of information you've gathered from the questions asked, and so on.

Main Question: can we get people to claim more value in initial offer and in overall negotiation?

Now, move on to a fully automated agent with IAGO [25]. Tested 90 people split into 3 categories: 1) no feedback, 2) general feedback, and 3) personalized feedback.

Results: we're good at teaching value claiming, but not value creation.

Next steps: maybe we need to rethink how we're capturing value creation. Could draw on opponent model to better understand negotiation.

3.9 Thayne Walker: Multi-Agent Pathfinding in Complex Domains

Example: would you want to ride in a particular air taxi?

Three kinds of taxis: 1) bounded suoptimal, 2) resolution suboptimal, 3) one that replans conveniently around a hot airballoon, and 4) a taxi that moves smoothly around obstacles.

So: we want a taxi that can plan quickly and come up with a good solution.

Objective: Multi-agent planning algorithms that are efficient and have bounded sub-optimality.

Classic Multi-Agent Path Finding (MAPF) problem:

> **Definition 6** (Multi-Agent Path Finding): *Consider k agents, each with a unique goal $g_1, \ldots, g_k$, moving in a grid. Agents collide if they move into the same cell.*
>
> *Find the multi-agent policy $\pi : \mathcal{S} \rightarrow \mathcal{A}^k$ that delivers all agents to their goals as quickly as possible.*

"Complex Domain" means non-unit costs, variable length action durations, agents with definite size and shape, and movement with variable speeds.

Q: Lots of measures of success: low variance over time-to-goal, lower the min, lower the avg, etc.

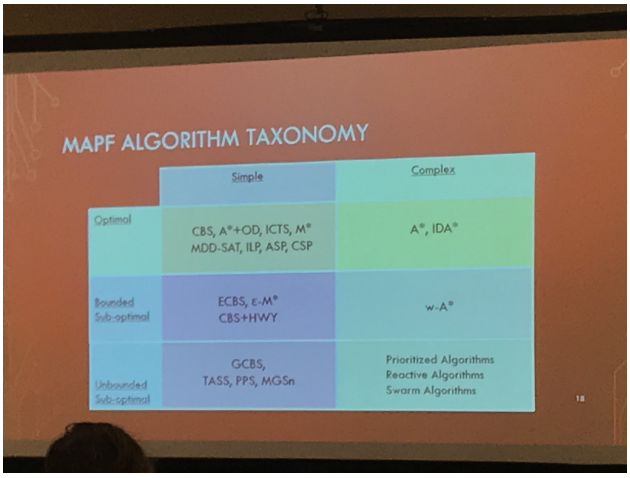


Figure 3: Taxonomy of MAPF approaches

New approach 1: CBS + CL, which fills in the bottom right slot of Figure 3.

New approach 2: Extended ICTS, which extends the "increasing cost tree search" algorithm for non-unit costs, which fits in the top right slot of Figure 3.

Current Work: extends conflict based search (CBS) CBS [33] by adding a conflict avoidance table and prioritization of conflicts.

Hypothesis: a particular statistic, "ecap", can predict the effectiveness of approaches to conflict avoidance. "ecap" means "equivalent-cost alternate paths", which is roughly: $\frac{\text{num states in equivalent paths}}{\text{num states in optimal path}}$.

Extension: apply constraints over both time, overlaps of paths/edges, and other dimensions/planning constituents.

.....

3.10 Christopher Fourie: Adaptive Planning with Evidence Based Prediction

Goal: robot needs to understand what a person is going to do in the context of a joint robot-person collaborative activity.

This involves: activity recognition, segmentation, and predictions.

Idea: we get less idle time with people/robot if robots can anticipate the person's behavior! So, let's get robots to anticipate behavior.

Research Goal: Get collaborative systems to accommodate 1) individual behavioral patterns, 2) preferences in task ordering and timing, 3) new behaviors as they occur.

Focus: Repetitive Task Environments (RTEs).

Technical Approach:

1. *Modeling and an Activity Prediction Framework:* predicting activities in RTEs.
2. *Planning for Improved Fluency:* in RTEs.
3. *Human Experiments:* and the controlled evolution of fluency/efficiency in RTEs.

Human-Study: collect data of people performing a partially ordered task as naturally as possible. Fit a gaussian model to the time taken by each person – can then evaluate how well that model predicts the time taken for other people.

→ Finding: we're not sure if we can expect models like this to predict orderings and time taken for *new* people. Doesn't generalize all that well! Individuals are highly unique. But, inter-ordering behavior is consistent.

So, idea: define a temporal predictor for each ordering. Idea is to use a mixture model for predicting time-taken given the ordering (so its a mixture of orderings).

Next idea: learn an activity recognition model for sets of trajectories, and use this model to augment the temporal prediction.

.....

3.11 AI Roadmap Panel

To close out the night there will be a panel discussing the next 20 years of AI-research, chaired by Yolanda Gil and Bart Selman.

Video here: `https://vimeo.com/313933438`

Figure 4: AI Roadmap

US Congress said, in November: we do not want to legislate anytihng about AI/ethics/society, since we (meaning congress) don't understand AI (instead they should turn to researchers).

Research roadmap was commissioned by the NSF – Computing Research Association (CRA) and Computing Community Consortium (CCC) are main institutions behind the roadmap.

Objectives:

- 10-20 year Roadmap
- Guidance for funding agencies and congress.
- Relate to AI research in industry.
- International AI initiatives.

Other documents:

- US National AI RandD strategic plan, 2016: `https://www.nitrd.gov/news/national_ai_rd_strategic_plan.aspx`
- US National Robotics Roadmap, 2006, revised 2006: `https://cra.org/ccc/wp-content/uploads/sites/2/2016/11/roadmap3-final-rs-1.pdf`
- 100 year study of AI, 2016 report: `https://ai100.stanford.edu/`

- AI strategies/investments abroad: `https://medium.com/politics-ai/an-overview-of-national-ai-str`

Held three workshops on 1) Integrated Intelligence, 2) Meaningful Interaction, and 3)
They'll next give summaries of each workshop and their objectives.

3.11.1 Integrated Intelligence

Marie Desjardins is the speaker. We thought about three big themes: 1) Mind, 2) Knowledge repositories, and 3) Contextualize AI.

Goal: Be cross cutting, focus on big interdsciplinary areas that can have big impact. Identified four core areas:

1. **Science of integrated AI.** This is all about the *integration* of the many subfields we've studied thus far. How do we bring together perception, deliberation, and control? Metareasoning, reflection? What are the components of intelligence?

2. **Contextualized AI.** involves personalization, social cognition, persistent over time, highly customizable to individuals.

3. **Open Knowledge Repositories,** we *can't* have closed off knowledge bases for individual institutions. This needs to be a community resource! Dave: Wow that's a fascinating idea.

4. **Understanding Human Intelligence.** Unifying theories of human and artificial intelligence, AI to understand human intelligence, and AI inspired by human intelligence.

3.11.2 Meaningful Interaction

The speaker is Dan Weld. This workshop focused on collaboration, trust and responsibility, diversity of interaction channels, improving inline interactions.

A few societal vignettes to concentrate on, including robot caregivers, training for robot repair jobs, custom personal devices, and so on.
Main technical areas:

1. **Collaboration:** how can we model human mental states, get AI systems to understand people better, reliability and ethical behaviors.

2. **Diversity of Interaction Channels:** diversity of human ability, multimodal explanations, privacy preservation across channels.

3. **Trust and Responsibility:** Transparency and explanation, debugging behaviors, boundaries and responsibility.

4. **Improving Interactions Between People:** customized presence, collaborative creation, reputation and factfulness.

3.11.3 Self-Aware Learning

The speaker is Tom Dietterich. This workshop focused on robust and trustworthy learning.

Main technical areas:

1. **Robust and Trustworthy Learning:** Quantifying Uncertainty, identifying risks/failure modes, failing gracefully.
2. **Deeper Learning for Challenging Tasks:** learning from few examples, learning through interactions, long-term adaption.
3. **Integration of Symbolic and Numeric Representations:** abstracting symbols from numeric representations, explainable and instructable AI, representing complex structures beyond word embeddings.
4. **Learning in Integrated AI/Robotic Systems:** robust object manipulation, learning from humans by demonstration and instruction.

3.12 A New Era of Audacious AI Research

Now Bart Selman is the speaker. "Audacious" AI research tackles broader AI goals. Basically: how can we do massive scale AI projects? (*a la* the hubble, LHC, human genome project?)

Proposed Recommendations:

1. **Open National AI Platform:** shared ecosystem/infrastructure for AI research, example resources, data repositories, wide range of contributors, hardware/data/software/services.
2. **Broaden AI Education:** need for education of official degrees/certifications in AI at all levels, fellowships for grad students, need creative incentive mechanisms.
3. **Promote AI Policy and Ethics:** promote AI research that focuses on characterizing and quantifying AI systems, need to promote emerging cross-cutting disciplines for AI.

3.12.1 Q& A

Now onto some Q & A! They also gave out an email address folks can contact with questions or ideas: `gil@isi.edu`, `selman@cs.cornell.edu`, and `cccinfo@cra.org`

Q: You mentioned some public resources (like data sets/hardware available for public use) – who will build such a thing?

→ A: Well, we're mostly defining the agenda, and trying to highlight the landmarks that we can hit if we invest our resources as a society in the right way.

Q: If we build a communal knowledge graph, how do we do that in a way that is unbiased? Make it open? Targeted initiatives?

→ A: Yes! We discussed this point a lot. Social norms and counterfactuals and fictional worlds are all very challenging things to embed in a knowledge graph. There will absolutely not be the "one true" perspective on the world.

Q: What's the plan for the roadmap given that governments around the world are realizing the power of AI and are starting to regulate AI? Might interfere with what we do/want to do?

→ A: We take the position that we support fundamental research in AI, and open research in AI. There's a military component that we do not really address.
→ A2: many ways folks might want to control AI – control for good of society, good of individual/country. We discuss control of intellectual property in the doc. Core philosophy: roadmap is how we open the development of AI techniques so we benefit all of society.

Q: Sharing info among AI researchers, as with arXiv. Some of the presentations from today were online, for instance. It's helpful to have shared presentations/papers. Any ideas for sharing work more easily?

→ A: That's an important insight! The more we understand each others' expertise/perspective the better. So let's continue to focus on open/accessible software/data/papers.

Q: How does modeling business processes show up in the roadmap?

→ A: Definitely came up in the workshop(s).

Q: How do companies/industry fit into the roadmap? I didn't see them show up all that much.

→ A: Many folks in industry are involved, lots of people at the workshops were from industry. Your point is well taken and we welcome involvement from industry.

Dave: Now the Q& A is moving to a more open "suggestion/general question" session
Q: A suggestion – focus on natural disasters (see: Harvey, Katrina, wildfires).

Q: Lots of societal challenges on the horizon – thoughts on how to tackle them?

→ A: well, we're trying to draw on many disciplines to better understand and prepare for this impact.

Q: This is really an international problem – we're working on the same thing in Europe. So, perhaps we should connect our agendas.

→ A: Absolutely! We hope we can work on that soon.

Q (from Ed Feignenbaum!): To do what you laid out is going to take an *army* of AI research superstars. Half the trickle of people we train go off to Google/Microsoft/Facebook. Are you worried about this education problem? We need an army, we're getting a platoon!

→ A: Yes, this came up. We have to make academia an environment for doing this kind of audacious AI research. People find industry exciting because of the money, people, and research. Great question – we'll raise this question in the report and will try our best to address this.

→ A2: we had a big discussion about this. Someone *during* the workshop got a 7 figure offer. No, we can't match that. But, maybe we can offer something else.

→ A3: Another thing we're addressing is how fast society can adapt to changes of all sorts. Also critical to ensure we can increase the diversity of CS education!

Q (CTO of lincoln labs): Aloha! Thanks for hosting this in Hawaii. Reminded of the early days of the internet/digital age: Optimism was huge, but later we found massive issues we didn't predict. My feedback is this: you have an opportunity/responsibility to put forward that perspective. It needs to be at the forefront.

Q: Curious about the definition of advancement of AI? What does it mean to progress AI?

→ A: We interpret intelligence very broadly – human/biological/animal/artificial are all included.

Q: Lots of wonderful ideas for areas to focus on – do you have suggestions for how to improve our research methods going forward, given the demands placed on us as a community?
Q: So many of us probably saw Mark Zuckerberg explain really basic aspects of facebook and the internet to a panel of senators. My question is: how should we deal with the fact that policy makers will be forced to wrestle with areas they have no expertise in? What can we do to both work effectively with policy makers and make sure knowledgable people are involved in critical decision making?

4 Tuesday January 29th

The official conference sessions begin! We start with some opening remarks from the President of AAAI, Yolanda Gil.

New code of conduct available on the conference website[1].

Now, the chairs: Pascal Van Hentenryck and Zhi-Hua Zhou.

First, we remember some people we lost this past year:

- Alan C Schultz (1957-2019), Naval Research Laboratory.
- Zohar Manna (1939-2018): wrote many books in AI, focus on temporal logic and automated theorem proving.

Acknowledgements:

- Best Paper committee: Boi Falting, Fei Sha, Dave: one other I missed :(
- Eugene Freuder as Presentation Chair – put a lot of work into improving the oral presentations.
- For priceless advice, Sheila McIlraith and Killian Weinberger (chairs from last year) and Yolanda Gil (president).
- Peng Zhao, workflow chair.
- The 89 AC, 322 SPC, and 3450 PC members.
- Amazing AAAI staff: Carol Hamilton, Monique Abed, Diane Mela, Ipshita Ghosh, Juliana Rios, Mike Hamilton, and Kapil Patnaik.
- Kevin Leyton-Brown and Milind Tambe for organizing the AI for Social Impact track.

What's new this year:

- Summary Reject procedure: if paper is not relevant to AAAI, violates blind submission policy/page limit, clearly too low quality, is plagiarizing. Very conservative! Ended up "summary rejecting" 234 papers (out of $\approx$ 7,000, so only 3%.
- Bidding for papers: only chosen a smaller set for PC members to bid (150, ish).
- Used Toronto Paper Matching System and subject area to match reviewers.
- More strict double blind policy: SPC identity not visible to reviewrrs, AC identity is not visible to SPC and reviewers, reviewers did not no any identitifies.
- Added the following question to let SPC/AC judge how senior/junior the reviewer is.

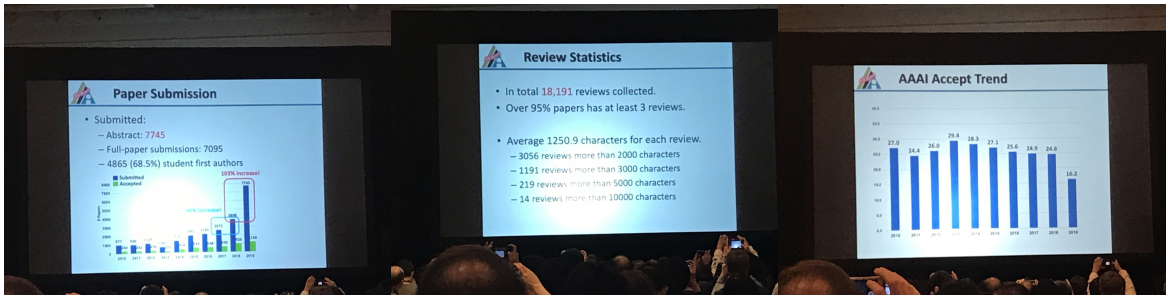
[1] `https://aaai.org/Conferences/AAAI-19`

- Presentation format selection: accepted papers had the opportunity to upload slides to determine if the paper would be well suited for an oral. SPC and AC make the recommendations. PC co-chairs finalize presentation format.
- 7,095 submissions. Most submissions ever! Quality was very high, though. Average scores were significantly higher than last year. → AAAI will look for larger future venues to accommodate growth.

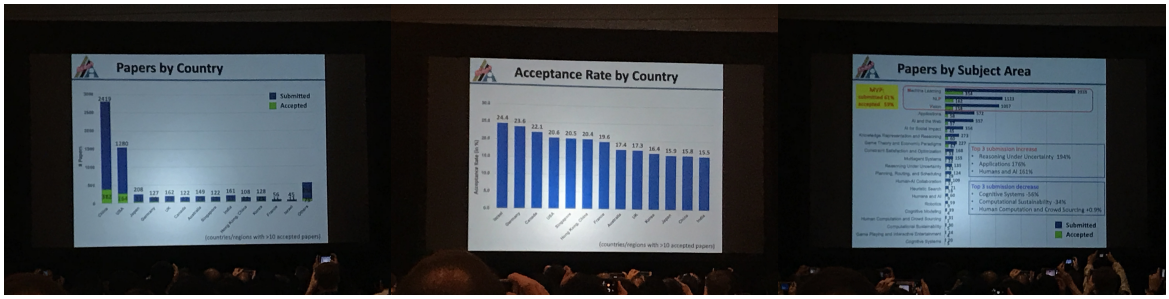
Now Zhiao will share somr statistics:

- Abstract: 7745
- Full paper: 7095
- 18,191 reviews collected
- Over 95% papers had at least 3 reviews
- Average 1250 characters for each review.
- Accepted 1147, 460 orals, 687 poster papers.
- 122 Technical Session

Some images to summarize:

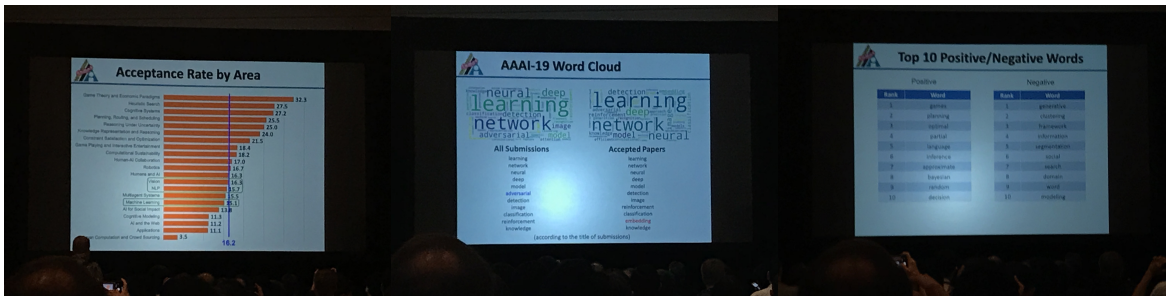


(a) (b) (c)

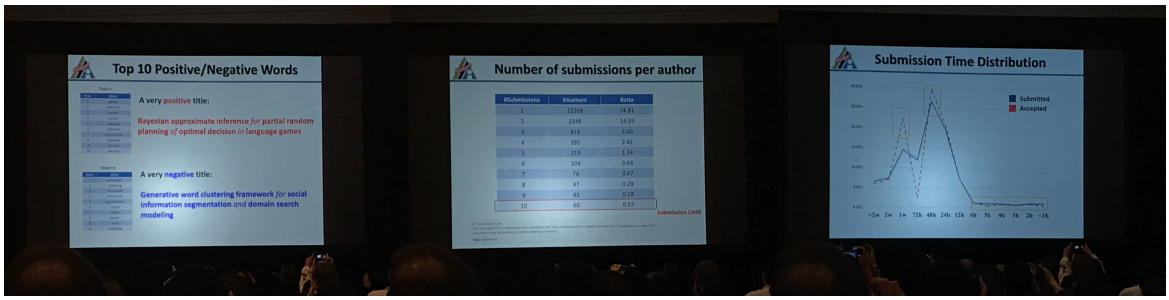


(d) (e) (f)

Figure 5: Stats on AAAI



(a) (b) (c)



(d) (e) (f)

Figure 6: More stats on AAAI

Next up, a quick summary of IAAI: huge diversity of application areas! 5 deployed application awards focusing on programming, life insurance. Milind Tambe won the "Robert S. Engelmore Memorial Award".

Awards:

- Senior Member Presentation Track, chaired by David Aha and Judy goldsmith.

  → Blue Sky Idea Track: 30 submissions, 15 accepted. 4 winners:

  1. Explainable Normative and Justified Agency by Pat LAgley
  2. Building Ethically Bounded Agents by Francesca Rossi
  3. Recommender Systems: A Healthy Obsession by Barry Smyth

Awards:

- New Fellows: Vince Conitzer, Luc De Raedt, Kirsten Grauman, Charles Isbell, Jiebo Luo, Huan Liu, Peter Stuckey.
- Senior Members: Bo An, Roman Bartak, Yiling Chen, Cristina Conati, Minh Do, Eric Eaton, Vincent Ng, Marco Valtorta, Yevgeniy Varbeychik, and Kiri Wagstaff.
- Classic Paper Award: Contend-Boosted Collaborative Filtering for Improved Recommendations: "Content Boosed Collaborative Filtering for Improved Recommendations" by Prem

Melville, Raymond J. Mooney, and Ramadass Nagarajan, presented at AAAI 2002: "For showing a way to cmopletement content-based and collaborative filtering approaches in recomenndation systems.""

- Classic Paper Honorable Mentiond: $D*$-Lite by Sven Koenig and Maxim Likhachev, also at AAAI 2002, "For developing an incremental heuristic search algorithm for robot navigation in unknown terrain that is easy ot udnerstand analyze and extend."
- Distinguished Service Award: Shlomo Zilberstein, "For his sutained and conscientious service and leadership both to AAAI as a councilor and conference committee chair and to the broader AI community, as the president of ICAPS."
- Feigenbaum Prize: Stuart Russell: "In recognition of high-impact contributions to the field of AI through innovation and achievement in probabilistic knowledge representation, reasoning, and learning."
- AAAI/EAAI Outstanding Educator Award: Ashok Goel.

.....

4.1 Invited Talk: Cynthia Brazeal on Living and Flourishing with AI

Video here: `https://vimeo.com/313938302`

To start, consider: many of our world leaders *don't understand AI*. So, we have a grand challenge to work hard toward a positive future despite this fact.

We all know AI is transforming the workplace, but recently it's started to transform our personal lives too. See: amazon echo, home robots, drones, siri, and so on. Children today are growing up in a world where they can interact with AI.

Great products are not only useful, but the *experience* of them need to be emotionally uplifting and offer enhancement to the human experience.

Main Question: Can AI help us flourish?

→ Relational AI ("relationship", not predicate/relation): AI that understands and treats people as people. We need:

1. Emotional engagement.
2. Collaboration between humans and AI as allies
3. Personal relationships.

To do this, we need to think of AI in terms of *socio-emotive* 1) perception, 2) learning, 3) interaction, and 4) expression.

Next set of transformative products and services will come from the intersection between AI and Design. Right now, these two fields are not deeply intertwined. We should be thinking about how to bring these two things together.
Main Goal: Bring together these two fields. Today we'll see work in Social Robots and social/emotional intelligence in AI.

Cars? Baxters? These are (or will be) social in some capacity. Using gazes and gestures.

Three Key Themes:

1. Human Engagement.
2. Allied Engagement, Personalization, and Learning.
3. Aging: Fostering Community.

4.1.1 Human Engagement

We have tons of processing power of our brain set aside for social processing – so, we *need* AIs in our community to do the same.

We are a species that evolved to be in the physical co-presence of others, and we've found that the more our machines/robots embody these characteristics precisely because it's how our brain works.

To see this, let's look at some chat systems like Alexa.

Experiment: three different AI chat systems answering the question "tell me about yourself".

- Alexa: I'm alexa, I can tell you the weather, etc.
- Jibo: My favorite things to do are talking to people and dancing. I also like Abe Lincoln.
- Google: I'm your google assistant, we can play mad libs, or spin the wheel.

"Robot Speed Dating" where families are brought into lab to interact with 3 different VUI agents. They ask questions to the three and then the people offer a diagnostic of the personality of these systems.

Takeaways: 1) people spend most of their time with these systems on social interactions, 2) the more "personality" in the system, the more engaging people find them.

Thesis: We are so profoundly social, so we have to pay attention to social aspects of these systems.

Q: What should the social dynamics of these systems be?

A: Can draw inspiration from human social interactions. It's dynamic, we need to form trust, and so on.

Three broad categories: 1) Intrapersonal context (smile detection), 2) Interpersonal context (trust), and 3) Intentional context (story telling).

Experiment with two children reading a story together. Basically: we want to model emotional understanding as intentional inference. Take a Bayesian theory of mind – storyteller views it as a planning problem, uses social cues to monitor listeners understanding (POMDP).

4.1.2 Allied Engagement, Personalization, and Learning

Bloom 2 sigma effect: prek-12 US education, not ready to learn, can't catch up. 60% children don't attend preschool. 37% of 12th graders read at or above proficient level.

Different designing a tutoring system for kids vs. adults. Kids learn through play and interaction!

→ So, main focus has been on developing a robot for teaching kids through play and social interaction. Treat Robots as peer-like learning companions.

Big Finding: social influence and emotion. If a kid plays a math game with a robot where the robot says "your brain grows the more you try!" it nurtures a *growth* mindset. Basically kids want to emulate the robot if they feel raport with the robot.

Q: What Role should a peer-like learning companion take and when?

A: Explored this question with a robot helping kids learn vocabulary (robot video!). Robot has different roles: could be a tutor or tutee. Different behaviors depending on whose "turn" it is. Robot can offer explanation, give definitions, correct demonstrations as a tutor. As a tutee, the robot can ask for help, ask for explanation, show curiosity and growth mindset. The kid said "I believe in you!" to the robot.
Dave: This video was powerful! Really gives a hint of an incredible vision of the future

Used RL in this setting. Have actually applied this system to schools in diverse Boston public schools.

Finding: Actual students learn better when the robot plays the adaptive role (both tutor/tutee) than the other cases.

Recent work on storytelling [29], new paper at AAAI this year.

Q: How can we foster a positive relationship between kids and robot tutors/peers?

A: See recent work! [41].

4.1.3 Aging: Fostering Community Connection

We live in a "silver tsunami" – we can't train enough people or build enough facilities to help elders.

Q: So, how can we use robots and AI to help with elder care? And so in a way with respect and dignity?

Main struggles of elderly: lonelieness, helplessness, boredom.

One route explored: use Jibo the robot to bring some joy to elders. It can dance, play the radio, take photos, tell jokes, and talk with folks. Designed to be a smart pet that can help with things – people that might usually be intimidated by technology readily engage and have fun with Jibo.

→ So, explored social robots for bringing joy to people. Older adults are the *most open* to these technologies. Huge positive response, they want to embrace these technologies.

Research Q: Can a social robot foster human-human connection in communities? (not just human-robot)

A: Yes! This really works. The social robot becomes a catalyst for people to be more openly social and form deeper social connections among each other.

→ huge opportunity for humanistic and social design to make an impact in peoples lives!

Takeaway: Social engagement isn't just about making it fun.

Would love to see the community focus on:

- Human factors and long term interactions between people/robots.It's hard but we need to do it!
- Ethics: educate people/next generation (including elementary school!), design appropriately, democratize AI – need AI to close the prosperity gap, not enhance it.

To close: a video of (young!) kids playing with Scratch – they're in early elementary school. And the amazing thing is, the right tools (Scratch) can actually give kids the opportunity to learn about the ideas in ML/AI/CS.

Punchline: "in a world of intelligent machines, humanity is the killer app" – it is so important that we design these systems to be **human centered**.
Dave: I have meetings now, will check back in for learning theory in the afternoon

.....

4.2 Learning Theory

Now for some learning Theory!

4.2.1 Precision-Recall vs. Accuracy [17]

Paper by Brendad Juba and Hai Le.

Definition 7 (Class Imbalance): *One (or a few) classes severely out-represent other classes in some dataset.*

So, pracitioniers find classification harder for imbalanced data, but learning theories suggest that imbalance shouldn't matter.

Goal: Analyze disconnect between learning theoretic view of data imbalance and practice.

Idea: need a new metric. Namely, learning theory usually suggests the use of *accuracy*, but we really need high precision or recall.

Case Study: Machine Translation – we find that 10s of billions of examples increases accuracy. But, why? Why does it need so much data?

Main contribution: derive relationship between precision-recall & accuracy.

→ Takeaway: Large data set is the cure for data imbalance.

Main theorem:

Theorem 4.1. *Suppose D is a distr over examples with boolean labels with a base psotive rate of $\mu \Pr_D(c = 1)$, h is a weak learner, and $\varepsilon_{prec}, \varepsilon_{rec}$ and ε_{acc} are the precision, recall, and accuracy error for h on D. Then:*

$$\varepsilon_{max} = \max[\varepsilon_{prec}, \varepsilon_{rec}],$$

satisfies:

$$\mu\varepsilon_{max} \leq \varepsilon_{acc} \leq \mu\left(\varepsilon_{rec} + \frac{1}{1-\varepsilon_{prec}}\varepsilon_{prec}\right)$$

Conducted experiments comparing performance of different techniques for fixing class imbalance. Observations:

- Training on a large data set improve precision and recall of class imbalance problem,
- Can't rely on preprocessing to fix this.
- Hard to achieve high precision-recall under severe class imbalance unless one possess a large amount of training data.
- Methods for correcting class imbalance don't often help.
- **Advice:** Incorporate explicit prior knowledge about the domain.

4.2.2 Theoretical Analysis of Label Distribution Learning

Paper by Jing Wang and Xin Geng.

Definition 8 (Label Distribution Learning (LDL)): *Learning setting wherein each label y is related with the instance x with label description d_x^y.*

Basically: feature space $X \in \mathbb{R}^d$, label space, label distribution function $\eta : X \times Y \to \mathbb{R}$. Training set $S = \{(x_1, d_{x_1}^{y_1} \dots d_{x_1}^{y_n} \dots)\}$. Learn a function from x to y given only these descriptions.

Model: one hidden layer and multi-output neural network with softmax output function, sum-squared loss.

Main theorem bounds the Rademacher complexity of AA-B and SA-ME (two different learning algorithms for the LDL problem), which can then be used to bound the (generalization?) error:

Theorem 4.2. *Upper and lower bounds on the Rademacher complexity of AA-BP: for a loss function ℓ with Lipschitz constant L_ℓ;*

$$\square \leq \mathcal{R}(\ell \dots) \leq \square$$

Dave: The bounds were complex, see the paper for details!

4.2.3 Dynamic Learning of Sequential Choice Bandit Problems

Paper by Junyu Cao and Wei Sun.

Example: you probably have received app notifcations/emails from apps you use. Positive: might boost engagement rate, but Negative: lead to marketing fatique and cause disengagement.

Goal: How can we address this trade-off? Questions: can we,

- Determine optimal sequence of messages?
- Dynamically learn users preference/patience?

Problem setup: N different messages to choose from. Each message i generates revenue r_i when selected by a user. For a user arriving at time t, the platform determines a sequence of messages $\boldsymbol{S} = S_i \oplus S_{i+1} \dots$.

User's choice: a user can either accept or reject a message.

Abandonment Distribution: Assume the probability a user abandons can be modeled as a geometric distribution. So: sequential choice model under marketing fatigue. User's valuation of a message denoted $u_i \in [0, 1]$.

Get an expected utility optimization problem (optimizing over choice of sequence of messages). So:

$$\max_S \mathbb{E}[U(S)] \tag{3}$$

Study both the online and offline variant. Contribution:

- **Offline:** Come up with an efficient $O(N \log N)$ offline algorithm. $\rightarrow$ Prove more patient customers will bring higher payoff.
- **Online:** Present a Upper Confidence Bound (UCB) like approach for the online SC-Bandit setting. Analyze this regret, which comes out to:

  $$\text{Regret}(T, u, q) = O(N\sqrt{T \log T}),$$

  where T is the horizon, u is the valuation, Dave: missed q.

Further *personalize* to individual users using Contextual SC-Bandits. Adopt the generalized linear bandit framework to do similar UCB like updates in the contextual setting.

Experiments in both the traditional SC-Bandit and the contextual SC-Bandit.

4.2.4 Near-Neighbor Methods in Random Preference Completion [24]

Paper by Ao Liu, Qiong Wu, Zhengming Lu

Consider recommender systems. Data are normally 1/5 stars, etc. But, in more generaly systems, we might imagine ranked preferences, or pairwise rankings.

Q: Can we use near-neighbor methods for doing random preference completiong?

Setting: $y_1 \dots y_m$ alternatives, and $x_1, \dots, x_n$ agents with given preferences over the alternatives.

More formally: KT-kNN algorithm, based on:

$$NK(R_i, R_j) = \frac{\#\text{ Pairs ranked opposite in } R_i, R_j}{\#\text{ Pairs ranked both by } R_i \text{ and } R_j}$$

Can then use NK between all agents to find nearest neighbors.

Q: Is NK an effective metric for doing nearest neighbors in this context?

A: Yes! See Katz-Samuels and Scott [20].

Open Question: Algorithms work under deterministic setting usually also work under random settings. Why?

Main Result 1 shows the predicted nearest neighbor by KT-kNN is far off from the desired prediction under a particular noise model (Plackett-Luce noise):

Theorem 4.3. *For 1D latent space with at least .5 probability:*

$$||x_{KT-kNN} - x^*|| = \Theta(1),$$

with x^* the "desired" prediction.

So, given this result: can we overcome this difficulty?

A: Yes! Through Main Result 2 $\rightarrow$ Anchor-kNN.

Anchor-kNN uses information from other agents' rankings. We now get features characterizing other agent's choices. Then:

Theorem 4.4. *For 1D latent space with at least .5 probability, if all agents rank at least poly-log(m) alternatives, with probability $1 - o(n^{-2})$:*

$$||x_{Anchor-kNN} - x^*|| < o(1),$$

with x^* the "desired" prediction.

Further conduct some numerical experiments to validate anchor-kNN relative to the KT-kNN. Takeaway: no matter what metric they tested with, they got great performance with Anchor-kNN. Further evaluate on a real dataset (Netflix) and find a huge improvement

Conjecture: their theorems generalize to higher dimensional spaces (instead of just 1D).

4.2.5 Dimension Free Error Bounds from Random Projections [18]

Paper by Ata Kaban.

Research Question: What makes some high dimensional learning problems easier than others?

Background: Learning from high dimensional data is challenging. Generalization error depends on input dimension in an essential way. How can we get a more flexible/available notion of generalization error?

Notation is typicl: $\ell\mathcal{Y} \times \mathcal{Y} \rightarrow [0, 1]$, $\mathcal{H}_d : \mathcal{X}_d \rightarrow \mathcal{Y}$ is the hypothesis class, training set, $\mathbb{E}[g]$ will mean generalization error, $\mathbb{E}[\hat{g}]$ will mean training error.

Main Idea: Translate some high dimensional data via a random project to make learning easier.

Definition 9 (Compressive Distortion): *Take $R \in \mathbb{R}^{k \times d}$ to be a random matrix full rank. Apply R to all input points.*

Consider an auxiliary function class $\mathcal{G}_R = \ell \circ \mathcal{H}_d$.

The compressive distortion of a function $g \in \mathcal{G}_d$ relative to $g_R \in \mathcal{G}_R$ is the following:

$$D_R(g, g_R) = [g_R \circ R - g]$$

Has some nice be properties: bounded independently of target if loss is Lipschitz, 0 if k is sufficiently large, choice of k is up to us.

Can then define a new complexity measure:

Definition 10 (Data complexity): *Of a function class $\mathcal{G}_d$ is given by:*

$$C_{2N,k}(\mathcal{G}_d) = \mathbb{E} \sup_{g \in \mathcal{G}_d} \inf_{g_R \in \mathcal{G}_k} D_R(g, g_R)$$

Main theorem:

Theorem 4.5. *For any $\delta > 0$, w/ Pr $1 - 2\delta$ unfiromly for all $g \in \mathcal{G}_d$:*

$$\mathbb{E}[g] \leq \hat{\mathbb{E}}[g] + 2C_{2N,k}(\mathcal{G}_d) + \underbrace{\cdots}_{\textit{Rademacher term}} \tag{4}$$

Applications: can reduce or eliminate dimensional dependence of generalization guarantees for a variety of domains.

Dave: I have meetings the rest of the day until the debate!

.....

4.3 Oxford Style AI Debate

Video here: `https://vimeo.com/313937094`

Proposition: "The AI community today should continue to focus mostly on ML methods."

The debaters:

- Team one (AGAINST): Oren Etzioni (OE), Michael Littman (ML)
- Team two (FOR): Jennifer Neville (JN), Peter Stone (PS)

Kevin Leyton-Brown (KLB) is moderating.

4.3.1 Opening Statements

JN: Let's start by talking about the goals of AI research. Goal is to understand the nature and limitations of computational intelligence. We also aim to create robust autonomous agents that can behave rationally and intelligently. Point 1: history of AI! 1956 Summer at Dartmouth they thought they could *solve* computer vision (in a summer). Reason being: unlike many other AI problems, vision is tractable and easy to encode. Rough timeline: AI Winter in the 70s. Things turned around in 80s and 90s, like IBM used statistical models for translation. Then we started using big data sets for learning, Tesauro made TD-Gammon in the 90s, 2006 Netflix competition to develop an ML technique for reducing prediction error (Netflix says 75% of things watched are from reccomendations). Hinton's group in 2012 used CNNs on ImageNet to great success, AlphaGo in 2016. All these breakthroughs were all due to machine learning. Therefore, we should continue focusing on ML b/c that's where our progress came from.

OE: Last five years have seen populist movements. 1) The Donald in the USA, 2) BREXIT, 3) a populist movement toward machine learning. You might ask: what's wrong with ML? We're hip! We get good results. I worry about an AI winter. Populist movements are all fine, they get 10% gains on things b/c of CNNs, RNNs, ABC, HBO, etc. But, what happens when this continues? Let's look at the key people in these movements: Roger Stone dresses like Hannibal Lecter and was arrested by FBI agents that volunteered to arrest him. The machine learning movement has Peter Stone! No coincidence there? ML populist movement is plotting. They're trying to BUILD A WALL between ML and AI! We know better. Let's get a little more serious and dispense with some obvious points. ML has been successful: deep learning has exceeded expectations! Let's define ML, though. Our adversary may attempt to redefine ML in many serious ways. ML is ML (is ML) – it's supervised learning (with a cherry on top!). In reality, ML is limited: people choose the architecture, data, regularization, loss function, and so on. It takes blood sweat and tears to get it to work. ML is 99% human engineering/art. But the ML populist movement is claiming a lot more! AlphaGo – what about MCTS?

4.3.2 Main Statements

KLB: thanks for those great remarks on Brexit and some AI.

PS: What you're asked to judge is not how deep our voice is, or how funny we are – the question is whether we should continue focusing on ML in AI. And we should! For two reasons 1) it's our weakest link, and 2) we have the resources/people to make rapid progress. Focusing on ML now is healthy for our field. Many successes in AI – **decades** spent on symbolic reasoning. We then recognized it would be possible that it is impossible to solve perception perfectly. We're now in a phase of focused attention on understanding the degree to which ML can recognize and understand the world. This is very healthy! We're not taking a position on whether ML is more important. Just that we should focus on it now. Now is the moment for ML in the field! 60% of papers at AAAI were on ML/NLP. What will this focus provide? We may be able to answer some of our central questions like how to scale with human interaction, transfer learning, one shot learning and so on. Lots of energy – let's not throw it away. To summarize: we should keep focusing on ML b/c 1) it's our weakest link and we don't yet know the limitations, and 2) we have the critical mass to make serious progress.

ML: Nothing against ML! My initials. That being said, Peter would have the rest of AI be left alive in small dedicated communities. That's worrisome! 60% of papers in ML related disciplines? What about the change over time! This trend might actually be problematic. We had the easier side to argue partly because there's a lot of ways to tackle the problem of creating intelligence. So, our side is right if literally anything else makes progress. Something ML is doing right is *redefining problems to be well suited doing ML*. I made the case to Peter Norvig that we should use more structure in robotics – we don't see people solving problems in cross word solving. This would be a total mess to do deep learning! To me, ML systems are really attacking a different part of cognition than other parts of AI. It's sort of system 1 and 2 from Kahenmann and Tversky – ML is system 1. System 2 is reflection/structure/consistency. We need System 2. No long term coherence. Example (because y'know, puns): pun generation. Punning riddle: what do you call a green cow in a field? Invisibull! A deep learning system was trained on puns then generated: What do you call a short sense of humor? A charming saucer! It sounds like a joke, but all the depth/meaning/structure is missing. It's very surfacey. Alternatively, using GOFAI, same task: what do you call a murder that has fiber, a cereal killer. And that was better! so, QED.

KLB: Transition to free-for-all stage. Respond to things people made.

OE: Peter Stone and I definitely agree – just like Roger Stone, Peter has a perception problem. At 9000 problem there were people at AAAI! Biggest point to make: we don't want to throw it away. We want to note that it's a tool, not a panacea.

PS: Thank you Oren for reiterating my point exactly! It's a tool in the toolbox, and one we know the least about so far. Would also like to respond to Oren – we need to keep other communities alive. Continue our focus now on ML. Michael's thesis, was also quite good, algorithms for sequential decision making using machine learning! Yes Peter Norvig should be here. Yes my name is Stone.

JN: Michael said he was disappointed in Peter and I for originally being ML people that worked in complex structure problems that now we do learning – deep learning just worked better when I applied it! Also, when we say machine learning we do *not* mesn deep learning. We mean all of machine learning. It's a big umbrella! As a community, we start off by asking if problems are intractable. We've gotten to larger and larger problems over time because of machine learning. Like MCTS was able to be successful in AlphaGo because of learning!

KLB: AAAI chairs pointed out that most papers submitted come from ML but most folks in the crowd said no!

OE: It's simple! If you're thinking in the short term, like publishing, it's a way to publish. Yesterday in the road map session the question was raised "how do we build a general theory of intelligence". It's one of the most profound scientific questions we can ask. We're going to need a lot more than gradient descent.

ML: It's important to do ML in the context of structures/human in the loop.

OE: I'd like to clarify. The proposition is about it being *mostly* ML. We're not about it being zero ML. Let's not bargain over the price, let's think about the fundamental questions. The debate is about cognitive architecture. What's more complex? Intelligence or chrome browser? Chrome has 7 million lines of code. Are we really going to learn 7 million lines of code? We need structure. Mostly ML ain't going to cut it.

PS: We're not disagreeing about the larger picture. We want a general theory of intelligence. Our big question is to understand the nature and limits of computational intelligence. And yes in the long run we take a big tent view: all areas are important. But, we already know what to do when we have symbols. We don't know how to get the right symbols, we don't know the limits of ML. We should focus on ML until we identify their limits.

JN: I'd like to go back to McCarthy's statement at the Dartmouth Summer Conference – "every aspect of learning or other feature of intelligence". Just want to point out: first thing he points out is learning. Our focus on ML was at the core of AI as described to begin with.

4.3.3 Closing Statements

ML: So this was interesting! I kind of signed up to be on the other side when the proposition was different. I don't know if I buy Peter's story about "do ML now" and then "do other stuff later". The statement is sort of about focusing on ML indefinitely.. Anyway. I've been working with lots of scientists that want to skip the hard computation and get to something that just works quickly, which brings them to deep learning. I really do worry about these systems not having internal consistency. Machine generated music is not music – it's nonsense. Worried about the fact that we redefine the problems so that we can generate slightly better nonsense. Ultimately ML is definitely integrated into the bigger picture. We need to think about how to integrate symbolic/deep. In fact we need to think about how symbolic approaches might need to change in light of recent advances in learning.

PS: It's clear that this statement is true. At our current state of knowledge in the field, ML is our weakest link, and we have lots of energy focused on it now. So we should capitalize on that to put a lid on those questions, now is the moment for ML.

OE: Kevin let's not get personal here that's my job. My worthy adversaries in no way address my comments about Trump or Brexit. Since our so called moderator accused our side of lacking substance, here we go: we're in the midst of a revolution, certainly. But we need to go beyond classifiers and think about the deep issues we face. These require synthesis with other approaches and caring about different problems like using common sense to avoid brittleness, choosing what to learn in the first place. Newell and Simon studied cognitive architecture in the 80s! We drifted away from these problems. We can study these problems now with ML, but we can't just do ML.

JN: As a panel we got together and decided we were all pragmatists and that we care deeply about solving real world problems. So, we all care about including ML methods with a host of other techniques that come from other aspects of AI (planning, expert systems, reasoning on open worlds),

so I don't think as a community we shouldn't focus on function approximation as a solution to all of these problems. I do agree with Oren's point (somewhere in his Brexit rant) that ML uniquely designs tasks so that their methods work well. Instead of finding tasks we can solve we need to find tasks that need solving. So in that sense, I'm arguing a bit for the other side.

KLB: Thanks! That was fun!

.....

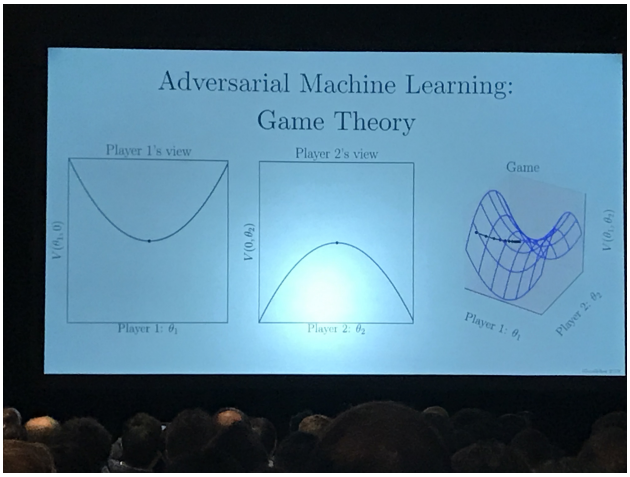


Figure 7: Game theory and optimization.

5 Wednesday January 30th

Onto Wednesday! Today starts with a keynote by Ian Goodfellow.

5.1 Invited Talk: Ian Goodfellow on Adversarial Learning

Video here: `https://vimeo.com/313941176`

Topic: Adversarial ML! And how it relates to other topics.

Traditional ML: based on optimization – choose a cost function $J(\theta_1, \theta_2)$, find θ_1 and θ_2 that minimize J. This works great for classifiers! We can gradually minimize J through various techniques until we find the right parameters.

But: lots of other parameters we can't really optimize.

→ Let's instead fall back to game theory. Two players play a game – player 1 wants to minimize the score of the game, player 2 wants to maximize. If it converges, we find an equilibrium point.

Cambrian explosion of ML research topics:

- used to be (2007), let's get ML to work!

  → Then, one that works, we can do vision/NLP and so on.

- Now, with ML working, we can go on to do loads of other things (neuroscience, security, RL, domain adaption, and so on).

  → So we're starting to see that happen.

5.1.1 Generative Modeling

Main idea: take a collection of training data, and learn a distribution that can generate similar samples [19].

Use Generative Adversarial Nets (GANs) [15]:

- Train a 1) generator to generate images, starting out random
- Train a 2) discriminator, to recognize fake images from real images.
- These two play a game to convergence.

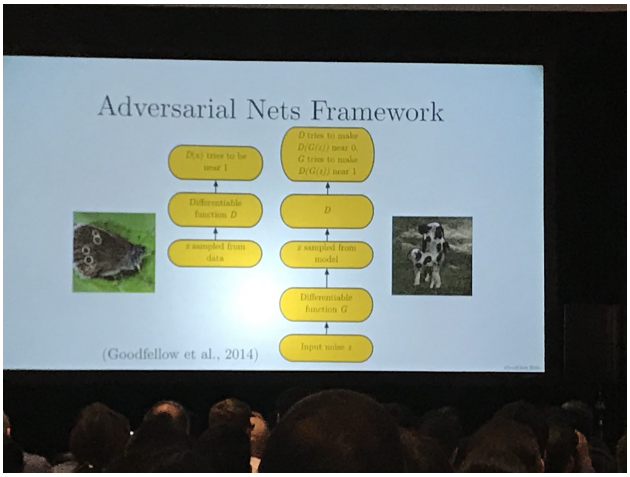


Figure 8: GAN Framework.

Rapid progress in image generation (and especially faces – see Figure 9). Also progress in imagenet, but it's harder because so many more classes.

GANs solve the generative modeling problem, but also the domain translation problem – can translate video stream during the day to video streams at night *without paired day-night examples*.

Cool example: cycleGan [47] converts horses to zebras. Also lets us diagnose some issues with the techniques – CycleGAN only does image-to-image, so it's not targeting video coherence. Also picks up on
Also showed an *amazing* video of a fake/animated dancer that can be used to change video stream of people to dance in that style.

Application: New company that generates custom made dental crowns very quickly, which dramatically improves over the traditional method for making crowns.

Prediction: Fashion! Should be possible to generate clothing that fits well/satisfies individual tastes.

Figure 9: GAN Progress.

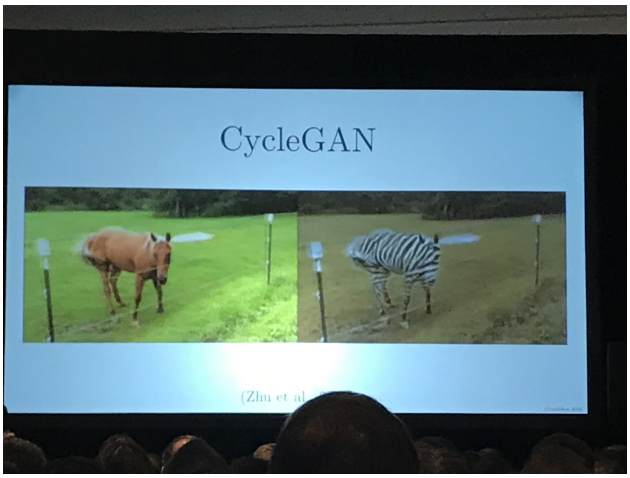


Figure 10: CycleGAN turns horses into zebras (actually a video).

5.1.2 Recent Developments

Basically: ML works now! So we can do all sorts of cool things, including:

- Security
- Model-Based Optimization
- Reinforcement learning
- Fairness and accountability
- Neuroscience

First, some new ideas in GANs:

- Self-Attention [39]: Attention mechanism you can add to a CNN that lets you focus on other parts of previous layers' feature maps, given a piece that the network just generated [] – so, when it generates an eye of an animal, you can highlight the eye and see what it was focused on.

  → Not constrained to any shape of attention region – it can highlight an arbitrary region.

- Also: BigGAN [4], large scale TPU implementation. Can generate images at high resolution that are good enough to fool human observers.

Security:

Adversarial Examples! We think these are the result of assuming i.i.d. data – attackers can violate both "i" assumptions.

→ attacker can choose examples not drawn from the same distribution (stop sign with graffiti, putting an apple in a mesh bag). → attacker can violate independence, too, to fool a model.

These completely work in the physical world too.

Advervarsarial training as a minimax problem:

$$\theta^* = \arg\min_\theta \mathbb{E}_{x,y} \max_\eta [J(x, y, \theta) + J(x + y, \eta, \theta)]$$

Open research direction: change the i.i.d. assumption and train on adversarial examples, too. Can often make classifiers more robust to these examples.

Model-Based Optimization:
Idea: train an ML model, and instead of training it to label new data, we use it to train the search for new data points.

→ DNA Sequence design.

Reinforcement Learning:

Recent work identifies adversarial examples for RL, too!

But, we can also use GANs to help RL. Consider Arthur Samuel's 1959 checkers player – used self play to improve itself [4], which is very much in use now.

GANs can be used to provide learned reward functions, as in SPIRAL [13]. Can generate reward functions in the appropriate input domain (robot camera percepts). Usual MSE won't work, but GANs can actually provide a useful distance measure such that a robot can learn to solve robot problems.

Extreme Reliability: We want extreme reliability for medical diagnosis, surgery robots, and other safety critical domains.

→ Adversarial machine learning might be able to produce extremely reliable systems because they are explicitly trained to be robust to attacks.

Virtual Adversarial Training [26]: take an unlabeled example, but we know that it ought to be labeled the same whether or not an adversary messes around with it. Works *extremely* well for semi-supervised learning (sample efficient, regularizes well).

Domain Adaptation: We train in one domain (perhaps where data is abundant) and test in another domiain.

→ A "domain" here is a particular choice of distribtion for training – ImageNet, videos of people walking down a street, and so on.

→ If we can achieve domain adaptation, pretty good evidence for reliable/robust generalization.

One main approach: Domain adaptation networks [12]. Idea: try to recognize the domain itself, which forces the discriminator to generalize well.

Another important instance of domain adaptation focuses on transferring from *simulated* training data to *real* data. Works well with recognizing where eyes are looking, for instance.

→ Can do robot grasping by training simulation and then actually grasp in the real world.

Fairness, Accountability, Transparency: GANs can learn more fair representations

→ Fairness: An adversary tries to infer a sensitive variable S from a representation. Learner tries to learn while making S impossible to recover [8].

→ Transparency: Interpretability and adversarial learning should talk more. Interpretability means getting the right answer, while adversarial training means the learner is getting the "right" thing.

Neuroscience:

Adversarial examples affect both computer and time-limited human vision:

.....

Dave: I have meetings now until the RL session.

5.2 Reinforcement Learning

Now for some RL! (yay!)

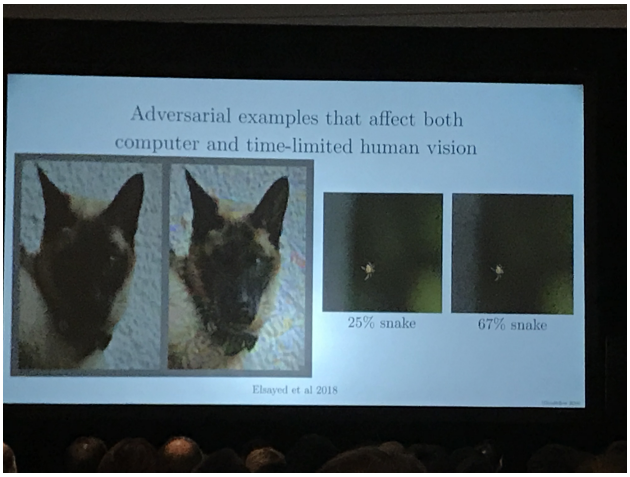


Figure 11: People are also prone to adversarial examples, too

5.2.1 Virtual Taobao: Online Environment for RL [34]

Paper by Jing-Cheng Shi1, Yang Yu1, Qing Da, Shi-Yong Chen, An-Xiang Zeng.

RL is sample inefficient – so, good virtual environments can be useful.

Recommendation in TaoBao is an RL problem. Taobao is an online store like Amazon – actions involve clicking around, searching, looking at products/reviews, and so on.

Claim: RL in TaoBao is infeasible.

Problem:

- User sends search request (can also click/pay/leave).
  → Goal: optimize customer preference
- TaoBao displays request (via a ranking policy, π).
  → Goal: Maximize performance

Main contribution:

- Use a GAN-SD (?) to learn a simulated customer policy. Does a good job of simulating the customer's distribution (Dave: over states? actions? I wasn't sure).
- Then do Multi-Agent Imitation Learning (MAIL) to learn customer shopping policy under a dynamic platform.
- Evaluated on TaoBao.

5.2.2 QUOTA: Quantile Option Architecture [45]

Distributional RL: Learns a *distribution* over the return instead of just the mean.

Standard RL: learns $v(s;\theta) \rightarrow \mathbb{E}[R(\theta)]$.

Distributional RL: learns full distribution, $\mathcal{N}(\mu, \sigma^2)$.

Research Question: Why need distributional RL?

$\rightarrow$ Main answer: can derive risk sensitive policies to explore more effectively.

Quantile encoding of a distribution encode *rank statistics* of a distribution. Use the median of each quantile to represent that segment.

Quantile-DQN: Maps each (s, a) to a value distribution of that state-action pair (extension to DQN).

Action selection is usually *based on the mean* of this distribution. Test on Atari, works better than DQN. Also test on "roboschool" (similar to mujoco). Also works.

Instead: select based on the k-th quantile:

$$a_t = \arg\max_a q_k(s, a),$$

where k is the k-th quantile of the value distribution.

Bencmarked several algorithms in a simple chain domain to gain intuition – compare a "pessimistic"-QR vs. "optimistic-QR". Can encode optimism/pessimism via choosing high/low quantile.

QUOTA: Hierarchical formulation where each quantile is an option.

5.2.3 Combined RL via Abstract Representations [11]

Paper by Vincent François-Lavet, Yoshua Bengio, Doina Precup, and Joelle Pineau.

Goal: Combine model-based and model-free RL to do hierarchical RL.

Definition 11 (Model-basedRL): *learn a model and do planning to compute Q.*

Definition 12 (Model-free RL): *learn Q/π directly*

Combined! Might be better (more interpretable, sample efficient, and so on).

Idea: Combined Reinforcement via Abstract Representations (CRAR). Model-based learns transition function to act, model-free learns value both in an off policy way.

Learning: $\rightarrow V$, use DDQN.
$\rightarrow T, R$, use an encoder. But! Trivial transition function often learned where all states are abstracted into the same state (which makes T prediction very easy).

So, add a regularizer/cost term that encourages more states.

Test in a labyrinth task. Yields a meaningful representation that can be visualized in 2dimensions that is interpretable.

Main evaluation on a randomly generated set of labyrinths – train on a small set of sampled MDPs, then test on a new set of samples. Also do zero-shot transfer on labyrinths.

Conclusion:

- CRAR can generalize while being efficient
- Can work from off policy data
- Approach recovers a low-dimensional representation of the environment even in the absence of model-free objective, which is important for 1) transfer, 2) exploration, 3) interpretability.

Dave: Now the poster spotlights, 2 minute highlights

5.2.4 Poster Spotlights

2 minute spotlights:

- **Diverse Exploration via Conjugate Policies:** noted that on policy methods suffer from a lack of exploration. Exploration is hard!

  $\rightarrow$ solution: diverse exploration. Deploy a set of diverse of exploration policies, where diversity means each policy will behave dissimilarly in each state.

  $\rightarrow$ Contributions: variance analysis of policy gradient objective under these dissimilar exploration policies.

- **State-Augmentation Transformations:** Look at MDPs in a risk sensitive manner.

  $\rightarrow$ Consider reward functions that take as input s, a *and* s'.

  $\rightarrow$ Analzse results if you consider the (s, a, s') based reward function instead of the "typical" kind. Provide a recipe for translating any transition based MDP to a state based MDP.

- **Trust Region Evolution Strategies:** Enhance evolution strategies for black box optimization using RL.

  $\rightarrow$ Contribution: make more efficient use of sampled data by optimizing surrogate objective function for multiple epochs of updates. Prove guarantees for this next optimization procedure, and a practical algorithm that makes a few approximations.

- **Comparative Analysis of Expected and Distributional RL:** Where does the advantage of distributuonal RL come from?

  → Main Q: in what settings does distributional RL behave differently from expected RL?

  → Conclusions at the poster!

- **Hybrid RL with Expert State Sequences:** Learning scenario where an RL has access to incomplete but obtainable expert demos.

  → Propose an efficient dynamics model to infer unobserved actions.

  → Joint policy optimization via RL and behavioral cloning.

- **Natural Option Critic:** Builds on Option Critic by combining with Natural Gradients.

  → Typical Option Critic uses regular gradient which is NOT variant to reparameterization.

  → Thus, if we extend to natural gradient, we can then do reparameterization, which leads to many practical improvements (without having to invert a matrix).

- **Utility of Sparse Representations for Control:** Fixed sparse representation like tile coding has been effective for control, but is not scalable b/c number of features explodes.

  → Goal here is to learn a sparse representation with neural networks (last layer has *sparse* activations).

  → Sparse representation is scalable to high dimensional inputs and is indeed helpful for RL.

6 Thursday January 31st

Next up, the joint IAAI/AAAI talk on smart cities.

6.1 Invited Talk: Yu Zheng on Smart Urban Cities

Video Here: `https://vimeo.com/313942000`

Consider: Rapid Progress of urbanization. Has led to huge challenges in dense urban areas, from traffic to housing.

Vision: Use urban computing to improve lives in cities through:

- Urban Sensing
- Data Management
- Data Analytics
-

Figure 12: Overview of urban computing vision.

6.1.1 Challenge 1: Urban Sensing

Challenges:

1. *Resource deployment:* How should we deploy sensors to gather the right data? Candidate selection is an NP-Hard problem.
2. *Measuring:* Hard to define a measurement for evaluating the deployment
3. *Biased samples:* taxi flow vs. traffic flow. Taxi traffic is biased toward particular routes (but we can get taxi data).

4. *Data Sparsity:* limited air quality sensors but want fine grained air quality throughout a city.
5. *Data Missing:* Communication/sensor errors.

If we can overcome these challenges, we can then collect a diversity of data: air quality, pedestrian traffic, bus use, etc.

→ Summarize data into a particular 6 kind ontology, based on their spatio-temporal type (static, temporal, dynamic, and point-based vs. network based. Goal is to make this scalable so that any new data types can be captured into the future. Summarized in Figure 13.

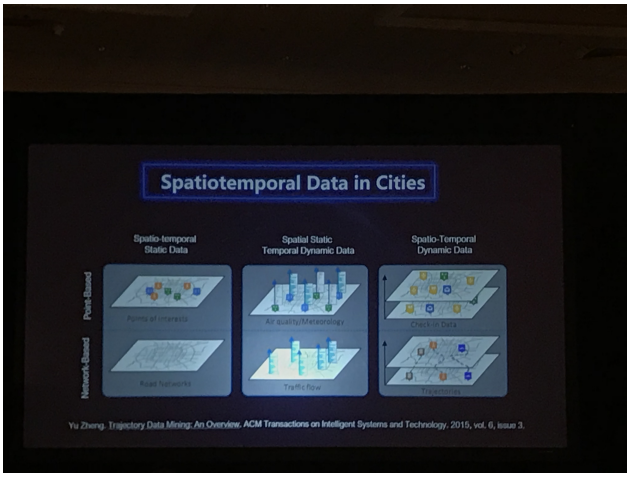


Figure 13: Categorization of kinds of data in urban computing, designed to scale with new types of cities/data/challenges.

Spatiotemporal data is unique:

- Spatial properties like distance, correlation. We can even apply triangle inequality.
- Spatial hierarchy: different spatial/geographic granularities, such as buildings $\in$ neighborhoods $\in$ districts.
- Temrporal closeness, period, and trend give us further structure to exploit.

6.1.2 Challenge 2: Data Management

Problem: large scale (city level!). So we need to handle a huge volume of data efficiently.

Hard to handle spatiotemporal data in particular:

- Trajectory data is represented by a sequence of data types is highly complex.
- Unique queries make searching the data difficult.
- Data is spread across different domains, so requires hybrid indexing for maning multi-modal data.

Example: detect vehicle illegal parking using bike-share trajectories.

→ Solution: Can spot when bikes take odd swerves on a road to estimate where cars might be parked illegally [16]. Use classification to spot the anomalous trajectories.

6.1.3 Challenge 3: Data Analytics

Example: Predicting crowd flows in a city. Difficult because of complex factors that effect this situation – depends on time, events, weather, traffic, and so on.

→ Solution: partition a city into a uniform grid. Count the in flow and outflow of people in each grid cell over time. Now, given this series of grid-based-heat-maps, want to predict the grid at the next time-frame. Most off the shelf deep learning models don't capture the relevant spatio-temporal structure. Instead, develop a new architecture that looks for specific structure relevant to the domain [44].

Problem: Air pollution is a global concern. Monitoring air quality is extremely difficult.

Solution: infer real-time and fine-grained air quality through a city [46]. Provide a user interface for zooming in and out of a city to check the air quality at different regions of a city. Can even identify the source of solution.

→ Deployed in over 300 cities in china(!). Even offers prediction over the next 48 hours–

6.1.4 Challenge 4: Providing Services

Developed: urban computing platform that can accommodate the 6 types of data discussed previously, specially tuned for spatiotemporal data.

Opens up the opportunity for cities to share data and tools to improve inference and analytics.

Deployed: JD iCity, can play with it here: `http://ucp.jd.com` (Dave: just a heads up it's in chinese).
Summary:

- Framework for urban computing
- Many research challenges, some effective solutions so far
- Urban computing platform as an OS for cities.

.....

6.2 Reasoning Under Uncertainty

Now for some reasoning under uncertainty.

6.2.1 On Testing of Uniform Samplers [6]

Paper by Sourav Chakraborty and Kuldeep Meep.

Andrew Ng: "AI is the new electricity"

And yet, it fails basic tasks: "Im a huge metal fan" $\rightarrow$ (translate to french) "I'm a large ventilated object."

This Work: Verification.

Given a model M, like a neural network to label images, and a specification, φ, that specifies the goals of M.

Idea of verification: We can check whether there exists an execution of M that violates φ.

Q: Yes, but so what?

A: Well, samplers form the core of the state of the art probabilistic reasoning techniques. Can use Markov Chain Monte Carlo (MCMC). Use statistical test to argue for quality of the sampling distribution.

Example: suppose we're uniformly sampling from a domain from 1 to N. Distance is total variation distance (ℓ_1). How many samples do we need before getting a collision?

Theorem 6.1. *Testing whether a distribution is ε-close to uniform has query complexity $\Theta(\sqrt{S}/\varepsilon^2)$, with S the size of the domain(?) Dave: missed citation – from another paper.*

Definition 13 (Conditional Sampling): *Given a distribution D on domain S, one can specify:*

- *Specify a set $T \subset D$*
- *Draw samples according to the distr. D_T, that is, D under the condition that the samples belong to T.*

Clearly, such a sampling technique is *as powerful* by setting T to the full support. But, what else can we do?

Sampling algorithm (for the two "true" uniform and non-uniform case – how can we determine which distr we are sampling from?):

- Pick two elements uniformly at random x, y.
- In the case of the "far" distribution, one of the elements will have Pr 0, and the other Pr > 0.
- Now a constant number of conditional samples is enough to identify that the distribution is not uniform.

Q: What about other distributions?

A: Need a few more tests, but largely same idea as the above.

Consider: uniform sampler for CNF formulas: Given a CNF formula ϕ and a CNF sampler A, that outputs a random solution of ϕ.

Definition 14 (CNF Sampler): *A CNF-Sampler A is a randomized algorithm that, given a ϕ, outputs a random element of the set S, such that for anyx $\in S$:*

$$Pr(A(\phi) = x) = \frac{1}{|S|}$$

Main problem: come up with a good CNF sampler.

Algorithm: Similar idea to before, yields the following main result:

Theorem 6.2. *Given $\varepsilon, \eta, \delta$, the number of samples the above their algorithm needs to accept/reject the formula,*

$$K = \tilde{O}\left(\frac{1}{(n-\varepsilon)^4}\right),$$

samples for any input formula ϕ.

Experiments: compare different CNF sampling algorithms on a variety of benchmarks.

Conclusion:

- Need methodological approach for verification of AI systems.
- Need to go beyond qualitative verification to probabilistic verification.
- Sampling is a crucial component of the state of the art probabilistic reasoning systems.
- This work: property testing meets verification, promise strong theoretical guarantees.

.....

6.2.2 Finding All Bayes Net Structures Near-Optimally [23]

Paper by Zhenyu Liao, Charupriya Sharma, James Cussens, and Peter van Beek.

Definition 15 (Bayes Nets): *A directed acyclic graph (DAG) that model a joint distribution over a set of random variables*

Bayes Nets can model conditional independence and causation, they can learn and model structure in data.

Structure Learning: learn a Bayes Net from data using a score-and-search approach, given training data of N instances.

$\rightarrow$ Lots of approaches to structure learning – 1) consider space of all DAGs, 2) restrict your structure, or 3) consider only the best k-scoring DAGs.

Problem: restricting structure too much might limit the Bayes Net, if you don't limit it won't scale.

This Work: Structure learning that fixes both of the above problems.

Main Results:

- Propose a novel approach to model averaging inspired by approximation algorithms
- Approach only consider models that are optimal or near-optimal (in score).
- Prove this approach is efficient and can scale to much larger networks than the SOTA.

Definition 16 (ε-BNSL): *Given $\varepsilon > 0$, a dataset I over variables V and a scoring function σ, the ε-Bayes Net Structure Learning (BNSL) problem finds all networks:*

$$OPT \leq score(G) \leq OPT + \varepsilon,$$

where $\varepsilon = (\rho - 1)OPT$.

Problem closely related to Bayes Factor (BF), which can be interpreted as a measure for the relative success of a model to predict data.

Other problem: scaling.

$\rightarrow$ Solution: prune the search space –

Theorem 6.3. *(From Teyssier and Koller [38]) Given a vertex and two parent sets Π and Π', we can prune the if $\Pi \subset \Pi'$ and $\sigma(\Pi) \leq \sigma(\Pi')$, then Π' can be safely pruned.*

This work extends this theorem to the ε-optimal case:

Theorem 6.4. *Given a vertex and two parent sets Π and Π' and $\varepsilon \geq 0$, we can prune the if $\Pi \subset \Pi'$ and $\sigma(\Pi) + \varepsilon \leq \sigma(\Pi')$, then Π' can be safely pruned.*

Experiments show that their approach both scales and achieves near-optimal scores.

.....

6.2.3 Rethinking the Discount Factor in RL [31]

Paper by Silviu Pitis.

Original title: "The MDP is all you need!" – we think that MDPs can sufficiently account for all the modeling power that we need.

Set out to show that MDPs are sufficient for general intelligence by defining some axioms and deriving rational behavior in MDPs. But, I couldn't do it. So, instead I'll be talking about rethinking the discount factor.

Why like the MDP as a model?

- Preferences induced by the discounted value function satisfy several notions of consistency (our axioms!)
- Fundamental theorem of Inverse RL – any abirtrary behavior can be represented as the optimal policy in some MDP [27].

Q: So, why might the MDP fail to model preferences?

A1: Human preferences are complex – maybe agent can't learn the "optimal policy".

A2: We have good reason to model preferences with respect to suboptimal policies.

A3: Cliff example in this paper gives a numerical example of the above.

MDPs can't model *arbitrary* preferences. Let A and B be two events. Then:

$$ABAAAA > AAABAAAA > AABAAAA, \tag{5}$$

can't be captured. But, it's irrational to do that anyway.

→ What we care about is capturing *rational* behavior.

Definition 17 (Rationality): *Characterized by axioms that we agree preferences should satisfy, as in the Von Neuman and Morgenstern axioms (Completeness, Transitivity, Independence, Continuity, and some time-axioms Irrelevance, Dynamic Consistency, and Impatience)*

This work: define rational preferences over actions, states, and policies. MDPs induce preferences according to:

$$\text{if } V^1(s_1) > V^2(s_2) \quad \text{ then } (s_1, \pi_1) > (s_2, \pi_2)$$

Main result:

Theorem 6.5. *There exist* $\mathcal{R} : S \times A \to \mathbb{R}$ *and* $\Gamma : S \times A \to \mathbb{R}^+$ *such that for all* s, a, π*:*

$$U(s, a, \pi) = R(s, a) + \Gamma(s, a)\mathbb{E}_{s' \sim T(s,a)}[U(s', \Pi)]. \tag{6}$$

Closest thing to the usual rationally result – basically, we need to think about discount factor as a function of (s, a) instead of being fixed.

Q: Given the observed behavior, what utility function (parameterized by both reward and discount), is being optimized?

A: Nice direction for future work!

.....

6.3 Invited Talk: Tuomas Sandholm on Solving Imperfect Information Games

Video Here: `https://vimeo.com/313942390`

Note: most real-world applications are imperfect information games.

Recently achieved superhuman AI performance in imperfect information games.

Q: How do we do it?

A: Techniques for perfect information games won't work. But, can rely on application/task specific techniques!

Challenges: 1) Uncertainty about what others and chance will do, 2) Hidden state, so we need to interpret signals and *use game theory.*

$\rightarrow$ The game theory is critical! But, hard to scale computationally. So, major challenge is scaling the game theory.

Libratus overview pictured in Figure 14.

Definition 18 (Extensive Form Game (EFG)): *Game with some chance and N players is an extensive form game. Also defines a game tree of possibilities over plays, with payoff at the leaves (general sum).*

Strategies for EFG: behavioral strategy σ_i with i an information set. Specifies a distribution over actions.

Player i's utility of strategy profile (σ_1, σ_2).

ε-Nash Equilibrium: strategy profile such that no player can improve behavior by more than ε by changing strategy.

Talk Roadmap:

- Abstraction: unifed framework for game abstraction with solution quality bounds.

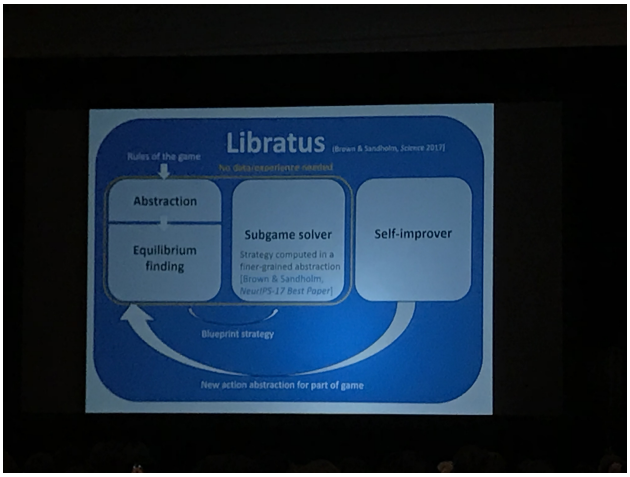


Figure 14: Overview of Libratus.

- Finding Nash Equilibrium: fast algorithms for solving games.
- Equilibrium Refinement

Abstraction
Started with lossless absstraction [14] but, found we needed to move to lossy abstraction. Turns out abstraction in games in non-monotonic [40]

Q: Can we get bounds on solution quality anyway?

A: Yes! [32]. Also applies to abstraction, since modeling is just abstraction.

Abstraction theorem:

Theorem 6.6. *Given a perfect-recall game, an acylic abstract game, a mapping between the two games that satisfies mild assumptions, an ε-Nash equilibrium in the abstract game, then: any lifted strategy is an ε-Nash equilibrium in the original game:*

$$\varepsilon' = \varepsilon + \ \textit{mapping error} \ + \ \textit{refinement error}.$$

Rest of the talk: theorems apply to 2-player, 0-sum games, but they're intended to apply to more general games.

Counterfactual regret minimization (CFR): used by every top poker AI in the past 5 years. Used a tabular form of CFR and abstraction before equilibrium finding.

→ Next up they introduce a function approximator for use in CFR (leads to "Deep CFR").

Chicken-and-egg problem with abstractions: hard to pick an abstraction without knowing the equilibria, but hard to find the equilbiria without using an abstraction.

→ Solution: do both simultaneously!

Monto Carlo CFR [22]:

- Action regret is how much better we would have done had we always picked this action in this situation in the past.
- Do Monte Carlo roll outs, always pick action proportional to *positive regret*.

Finding Nash Equilibria
Motivation: Previous fastest solver is CFR+, but has some sever limitations (namely: requires a huge number of iterations to pick the right action early).

→ Gives rise to linear CFR, which is a dramatically more efficient method than CFR (from 500,000 iterations to around a few hundred).

Two ideas:

1. Linear CFR: Weight iteration by recency for efficiency gains
2. CFR+: floor regrets at zero

Q: Can we combine them?

A: Theory → yes! Practice → no.

But, less aggressive combinations do well. This yields discounted CFR, which combines good aspects of each of these things.

Conducted experiments in Heads-Up No-Limit Texas hold'em (henceforth "texas") → DCFR and LCFR both dominate nearly all other strategies.

Q: What about Monte-Carlo variants?

A: Doesn't mix well with DCFR, but does mix with linear CFR quite well.

Limitations of CFR:

- Usually for 2 player games
- Only works with linear loss
- Doesn't support behavioral constraints

Next up: they'll fix all three of these.

Consider the "sequential decision making" strategy space [10].

Idea: define a local notion of regret $\hat{R}$ at each decision point of the player.

Theorem 6.7. *We can get an extension of the usual result of CFR but for any convex loss function (not just linear) based on the local regret:*

$$R^T = \ldots \leq \max_x \sum_{j \in J} \pi_j(x) \hat{R}_j^T$$

This new framework is much more general and can be applied to a variety of new applications.

Hot Off the Press:

- *Regret Circuits:* compositional calculus of regret minimizers.

  → Can support strategy constraints that cut across "infoset"s.

- *Breaking the complexity barrier:* new decomposition yields a CFR-like algorithm but converges in $O\left(\frac{1}{T^{\frac{3}{4}}}\right)$ instead of $O\left(\frac{1}{T^{\frac{1}{2}}}\right)$. First ever CFR like algorithm to break this barrier!

Perfect information games and single agent search: take some steps, then solve remaining subtree. If tree is too big, we break off a small chunk, and try to solve that subset.

Idea: depth-limited solving (DLS) [5]. In Libratus, when we solve a game, we solve it *all the way to the end of the game.* In things like "deep stack" they do this depth limited solving, but in an expensive way (random subgames solved in advance).

→ New Approach for solving DLS:

- At the leaf nodes at the depth limited tree, we allow the other player P_1 to choose a continuation policy.
- Solve the subgame with the current set of continuation policies
- Calculate best response for P_2.
- Add best response to set of leaf-node policies.

Dave: Not sure I got the player numbers right in the above..

Theorem 6.8. *The above approach converges to Nash equilibrium.*

Moreover, the above reaches very low exploitibility in a small number of iterations.

Key Takeaways:

- Planning is important in imperfect information gatmes
- In real time planning you must consider how the oppeonent can adapt to changes in your policy
- States don't have well defined values in imperfect info games
- New bot developed end-to-end on a single computer is 2nd best AI bot after Libratrus.

6.3.1 Equilibrium Refinement

Major Point: An issue with Nash: assumes strongest possible opponent! So, it doesn't capitalize on opponents' mistakes.

→ Enter, refinements, which can help to improve.

Idea: Mandate that users play each action at each information set. This enables reasonable strategies and beliefs to be computed even in non-standard parts of the game tree. Enables refinement!

Can formulate this problem as a Linear Program (LP), parameterized by $\varepsilon > 0$. Theory says there is an $\varepsilon^* \in \mathbb{R}_{\geq}$ such that the LP can be computed in polynomial time and achieve good performance (finding a good "basis").

But, the above induces an unusably slow algorithm for refinement (that has desirable properties). Instead, don't look for ε^*, look at something a bit weaker → gives rise to a practical algorithm.

Bottom line: refinements of Nash Eq matter when some players are not fully rational.

Consider Stackelberg games: extensive form games *with commitment.*

- One player commits to a public mixed strategy ("leader")
- Another play ("follower")

Goal is another equilibrium: "strong" stackelberg equilibrium) assume follower breaks ties in best possible way for leader).

→ Same issue as Nash: assumes fully rational opponent.

Q: Is trembling-hand perfection meaningful in Stackelberg games?

A: From the following theorem:

Theorem 6.9.
- *Trembling-hand Stackelberg equilibrium are Stackelberg equilibrium*
- *Finding a Stackelberg equilibrium is NP-Hard*
- *Finding a τ-Stackelberg equilibrium is NP-Hard.*

Conclusion:

1. Imperfect info games are important but different
2. Game-theoretic techniques are required to be robust against all opponents
3. Modern techniques for these (=imperfect info games) should be taught in AI courses
4. Talked about abstraction and its role in CFR
5. Finding Nash equilibria can be done scalably

6. May want to refine equilibria sometimes.

Future Work:

- Merge practical abstraction algorithm with new abstraction theory
- Even bigger games! Deeper/infinite, large branching factors
- Techniques that work with blackbox access to the world

And that's Thursday! Now onto the poster session.

.....

7 Friday February 1st

The final day! My talk is in the RL session this morning so I unfortunately didn't make it to the keynote.

7.1 Reinforcement Learning

First, some RL.

7.1.1 Diversity-Driven Hierarchical RL [36]

Paper by Yuhang Song, Jianyi Wang, Thomas Lukasiewicz, Zhengua Xu, and Mai Xu.

Code at `github.com/YuhangSong/DEHRL`

Focus: Hierarchical RL (HRL)

→ Idea: recombine sequences of basic actions to form sub-policies [37].

HRL can speed up learning, transfer, add explainability.

Game of Focus: Overcooked. Cooking game where agents move around a grid and place objects in different locations subject to a timer. Need to place ingredients on certain objects (bread in toaster, etc.) with the goal of making food based on (randomly generated) orders given.

Task Features:

- Primitive actions
- Abstract goals
- Sparse extrinsic reward (only received when agent collects the right ingredients based on the order

Using HRL on Overcooked: can use subpolicies to move agent in each direction, then higher level policies to move agent to each destination where ingredients can be collected/placed.

Main Idea: Exploit and learn diversity driven policies for use in HRL. Comes from the following assumption:

Assumption 1. *Learning different sub-policies is beneficial (popular in literature).*

Their extension:

Assumption 2. *With limited capability, learning sub policies far away from each one as possible is even better.*

So, with this assumption, their solution:

- Diversity of sub-policies can be measured by the **distance** between the resulting states of sub-policies.
- Diversity-driven solution:

  → Transition model memorize resulting states of different sub-policies

  → Intrinsic reward generated based on if a sub-policy leads to a state that is far away from the resulting states if choosing another sub-policies.

Form a neural network that trains the many aspects of the above solution (models, policies, and so on).

Experiments (in Overcooked):

- Sub-policy discovery at level 1 goes to the "five most diverse/useful states" (out of around 600).
- At level 2, sub-policies fetch different ingredients at the four corners.

Experiments (in Minecraft) with no reward/supervision: goal is to break a block, build a block, or jump on a block.
→ Random policies can't do anything, but their HRL approach can actually build some nice structures.

Dave: I'm up!

.....

7.1.2 Towards Better Interpretability in DQN [1]

Paper by Raghuram Mandyam Annasamy and Katia Sycara.

Goal: Interpretability!

→ Important for Deep RL since most neural nets used are treated as a black box, but naturally these systems will be deployed in crucial areas soon.

Lots of examples of seeking interpretability in supervised learning. But, it's harder in Deep RL

Proposed Method: "interpretable"-DQN. Keys initialized randomly and learned via backprop.

Four loss functions:

- Bellman Error (for Q Learning)
- Distributional Error (force good representations/robustness Dave: I think, missed this one)
- Reconstruction Error (force interpretable)
- Diversity Error (force attention)

Intuition:

- Use key-value pairs to enforce interpretability.
- Keys act like cluster-centers
- Can then evaluate/visualize these cluster centers through things like t-sne.
- Pass keys through conlutional network to generate canonical states

Evaluation:

- "Agreement metric" used for evaluating the approach.
- Induce new distribution in image space.
- Random agreement will be around $1/|A|$.
- In Pac-Man they find 30% agreement vs the random (which is roughly 10%).

Examples of memorization: Create small permutations to existing states and test what the agent does.

Provide an interface for visualizing the reconstructions when small changes are made to the image. Find their i-DQN can do a good job of ignoring the perturbations.

Problem: Lots of Deep RL methods overfit! [7, 43]. Can use i-DQN to explore this a bit.

.....

7.1.3 On RL for Full-Length Game of Starcraft [28]

Paper by Zhen-Jia Pang, Ruo-Ze Liu, Zhou-Yu Meng, Yi Zhang, Yang Yu, and Tong Lu.

Why Starcraft?

- Game is ideal for RL
- Most successful real time strategy of all time
- Really hard for human and AI to play
- If SC is solved, most games can be solved

Definition 19 (Starcraft): *Starcraft is a game where you control a base, units, and manage resources. You must build up a base and good units to destroy the other enemies.*

Difficulties: Starcraft can be multi-agent, is partially observable, and requires both macro level reasoning and low level control over many units.

Main focus: massive state-action space *and* long horizon until receiving reward. So, how can we solve this?

Approach:

- Low-level abstraction: build a building (select unit → build building → go back to work), or produce a unit (select building, etc.).

  → Can be inferred from mirroring human demonstrations, gives rise to macro-actions.

- High-level abstraction: Learn high level policies over these macro-actions (the "low-level" abstractions).

Experiments:

- Map: simple64
- Enemy: Terrain (built in AI)
- Agent: Protoss
- Fixed types of units/buildings for agent (but not enemy).
- High level policy that is decomposed into:
  1. a "base" policy responsible for handling base building and resource management.
  2. a "battle" policy responsible for dealing with battle/focused on individual control.

Conclusions: 1) Investigate hierarchical architecture for SC, 2) Simple yet effective training algorithm, and 3) Achieves SOTA results on SC.

.....

7.2 Reasoning under Uncertainty

Next some reasoning under uncertainty.

7.2.1 Collecting Online Learning of GPs in Multi-Agent Systems

Paper by Nghia Hoang, Quang Minh Hoang, Bryan Kian Hsiang Low, and Jonathon How.

Collecting Learning Motivation:

- Local agents that can upload local statistics
- Old approach: Central server combine and broad cast → But, limitation of typical approaches: centralized risk of failure and communication/computational bottleneck.

- Their approach fixes these shortcomings by removing central server.
  → No centralized risk of failure and no computational/communication bottlenecks.

Challenges: decentralization! Not clear how to do this (especially with resource constraints).

Goal: Develop efficient local model representation for online update with data.

Gaussian Process (GP prediction is *not efficient*:

- Computation is cubis
- Representation is quadratic
- Update is cubic

Solution: exploit sparse coding $\boldsymbol{u} = u(Z)$ is distributed by a standard GP.

Q: How can agent share models without communicating raw data?

A: Do local (bayesian/GP) updates, then share a representation that merges these updates. Main idea is to exploit additive structure to make this shared representation from a few messages.

Experiments: test on (real-world) traffic data set, characterizes different traffic phenomena.

→ Finding: they can reduce error with more data efficiently.

.....

7.2.2 Weighted Model Ingeration using Knowledge Compilation

Paper by Pedro Zuidberg Dos Martiers, Anton Dries, and Luc de Raedt.

Setting: Probabilistic inference. Goal is to combine the best of continuous and discrete inference

Knowledge compilation: take a boolean formula, compile it offline line so that you can do quick inference online.

New Method that can handle all of 1) knowledge compilation, 2) inference of density functions, 3) exact and approximate inference, and 4) can handle poly-nomial or non-linear constraints.

Contribution: Can handle probability density functions while applying state of the art knowledge compilation techniques, along with two new solvers: Symbo and Sampo.

Symbo does: 1) abstract theory, 2) compile formula, 3) to arithmetic circuit, 4) label leaves, 5) evaluate, 6) multiply by weight of continuous variables, 7) integrate.

Sampo does: approximate Monte Carlo inference for doing approximate probabilistc inference with linear time dependency. First sampling based algorithm for "WMI"

→ Both samplers beat state of the art.
Dave: Back to RL

7.2.3 Off-Policy Deep RL by Bootstrapping the Covariate Shift

Paper by Carles Gelada and Marc Bellemare.

TD Policy Evaluation with Linear models: samples $s \sim d_\pi, a \sim \pi, r = R(s,a), s' \sim P(\cdot \mid s,a)$.

Linear value approximation $V_k(s) = \phi(s)\theta_k$.

Weight update:

$$\theta_{k+1} \leftarrow \theta_k + \alpha(r + \gamma V_k(s') - V_k(s))\phi(s).$$

Can treat the above weight update as application of an operator.

Idea: also do *projection* operator this way: $\Pi_{d_\pi} x$. Well known result that Bellman operator converges.

Off Policy Problem: End up with a *different* operator that does not lead to convergence. Well know difficulty from Baird (Baird's counter example) [3].

Definition 20 (Covariate Shift): *Sample from another distribution μ, induces the covariate shift:* $\frac{d_\pi(s)}{d_\mu(s)}$.

Yields an update rule defined by this ratio, gives an operator:

$$C_{k+1} = \Pi_{d_\mu} Y c_k.$$

which *does* converge. But, it might converge to bad behaviour.

Idea: Can treat this as a projection into a simplex, which lets them translate the above new update rule to cooperate with neural networks.

Interpretation of the discount here: discounted of value function relxas the operator P_π to γP_π. Similarly, treat operator Y as relaxing P_π to $\gamma P_\pi + (1-\gamma) e d_\mu$.

Result: N step contraction factor under the discounted operator γY.

Theorem 7.1. *For any n, for any γ, can find an operator Y that converges to a fixed point.*

Experiment with the C51 agent (the distr. RL agent). Evaluate in "extreme off policy tasks"), like seaquest.

Dave: Back to reasoning under uncertainty.

.....

7.2.4 Compiling Bayes Net Classifiers into Decision Graphs [35]

Paper by Andy Shih, Arthur Choi, and Adnan Darwiche.

Goal: Compile a (Bayes Net) classifier into a decision graph to better explain a classifier.

Case study: win95pts.

$\rightarrow$ consider a printer with some symptons (slow printing, low toner, etc.).

Two kinds of explanations: can you fix a set of features such that the remaining features are rir-relevant, or can we remove erreoneous features.

Compilation:

1. Insight 1: recursively decompose into subclassifiers.
2. Insight 2: Identify subclasiffiers that are equivalent to save computation time.

Q: How can we. given two subclassifiers B_1 and B_2, identify if $\forall_{v \in V} B_1(v) = B_2(v)$.

A: Goal is to compute $\Pr(C = 1 \mid v) \geq T$, which we can rewrite as a linear inequality. Problem becomes separating points on a line, so we binary search in time $O(|V|)$.

Experiments: use classifiers from literature that we can compile successfully into small ODDs. Try win95pts, Andes, cpcs54 $\rightarrow$ they can all be merged and reduced quite dramatically.

Scaled this to networks with over 40 features.

Takeaways:

- Classifiers have an underlying decision function
- Decision functions can be compiled into an ODD
- Explanations and verification are efficient on ODDs (which are otherwise intractable on Bayesian network classifiers).

References

[1] Raghuram Mandyam Annasamy and Katia Sycara. Towards better interpretability in deep q-networks. *AAAI*, 2019.

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

[3] Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[5] Noam Brown, Tuomas Sandholm, and Brandon Amos. Depth-limited solving for imperfect-information games. *NeurIPS*, 2018.

[6] Sourav Chakraborty and Kuldeep S Meel. On testing of uniform samplers. *AAAI*, 2019.

[7] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. *arXiv preprint arXiv:1812.02341*, 2018.

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

[9] Theodore Eisenberg and Charlotte Lanvers. What is the settlement rate and why should we care? *Journal of Empirical Legal Studies*, 6(1):111–146, 2009.

[10] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Online convex optimization for sequential decision processes and extensive-form games. *arXiv preprint arXiv:1809.03075*, 2018.

[11] Vincent François-Lavet, Yoshua Bengio, Doina Precup, and Joelle Pineau. Combined reinforcement learning via abstract representations. *arXiv preprint arXiv:1809.04506*, 2018.

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[13] Yaroslav Ganin, Tejas Kulkarni, Igor Babuschkin, SM Eslami, and Oriol Vinyals. Synthesizing programs for images using reinforced adversarial learning. *arXiv preprint arXiv:1804.01118*, 2018.

[14] Andrew Gilpin and Tuomas Sandholm. Lossless abstraction of imperfect information games. *Journal of the ACM (JACM)*, 54(5):25, 2007.

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

[16] Tianfu He, Jie Bao, Ruiyuan Li, Sijie Ruan, Yanhua Li, Chao Tian, and Yu Zheng. Detecting vehicle illegal parking events using sharing bikes trajectories. *KDD*, 2018.

[17] Brendan Juba and Hai S Le. Precision-recall versus accuracy and the role of large data sets. *AAAI*, 2019.

[18] Ata Kabán. Dimension-free error bounds from random projections. *AAAI*, 2019.

[19] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[20] Julian Katz-Samuels and Clayton Scott. Nonparametric preference completion. *arXiv preprint arXiv:1705.08621*, 2017.

[21] Sarah Keren, Avigdor Gal, and Erez Karpas. Goal recognition design for non-optimal agents. In *AAAI*, pages 3298–3304, 2015.

[22] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte carlo sampling for regret minimization in extensive games. In *Advances in neural information processing systems*, pages 1078–1086, 2009.

[23] Zhenyu A Liao, Charupriya Sharma, James Cussens, and Peter van Beek. Finding all bayesian network structures within a factor of optimal. *AAAI*, 2019.

[24] Ao Liu, Qiong Wu, L Zhenming, and Lirong Xia. Nearneighbor methods in random preference completion. In *Proceedings of 33rd AAAI Conference on Artifical Intelligence (AAAI-19)*, 2019.

[25] Johnathan Mell and Jonathan Gratch. Iago: interactive arbitration guide online. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 1510–1512. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

[26] Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[27] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, pages 663–670, 2000.

[28] Zhen-Jia Pang, Ruo-Ze Liu, Zhou-Yu Meng, Yi Zhang, Yang Yu, and Tong Lu. On reinforcement learning for full-length game of starcraft. *AAAI*, 2019.

[29] Hae Won Park, Mirko Gelsomini, Jin Joo Lee, and Cynthia Breazeal. Telling stories to robots: The effect of backchanneling on a child's storytelling. In *Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction*, pages 100–108. ACM, 2017.

[30] Luis Enrique Pineda and Shlomo Zilberstein. Planning under uncertainty using reduced models: Revisiting determinization. In *ICAPS*, 2014.

[31] Silviu Pitis. Rethinking the discount factor in reinforcement learning: A decision theoretic approach. *AAAI*, 2019.

[32] Tuomas Sandholm and Satinder Singh. Lossy stochastic game abstraction with bounds. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 880–897. ACM, 2012.

[33] Guni Sharon, Roni Stern, Ariel Felner, and Nathan R Sturtevant. Meta-agent conflict-based search for optimal multi-agent path finding. *SoCS*, 1:39–40, 2012.

[34] Jing-Cheng Shi, Yang Yu, Qing Da, Shi-Yong Chen, and An-Xiang Zeng. Virtual-taobao: Virtualizing real-world online retail environment for reinforcement learning. *arXiv preprint arXiv:1805.10000*, 2018.

[35] Andy Shih, Arthur Choi, and Adnan Darwiche. Compiling bayesian network classifiers into decision graphs. *AAAI*, 2019.

[36] Yuhang Song, Jianyi Wang, Thomas Lukasiewicz, Zhenghua Xu, and Mai Xu. Diversity-driven extensible hierarchical reinforcement learning. *AAAI*, 2019.

[37] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2): 181–211, 1999.

[38] Marc Teyssier and Daphne Koller. Ordering-based search: A simple and effective algorithm for learning bayesian networks. *arXiv preprint arXiv:1207.1429*, 2012.

[39] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[40] Kevin Waugh, David Schnizlein, Michael Bowling, and Duane Szafron. Abstraction pathologies in extensive games. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 781–788. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[41] Jacqueline M Kory Westlund, Hae Won Park, Randi Williams, and Cynthia Breazeal. Measuring young children's long-term relationships with social robots. In *Proceedings of the 17th ACM Conference on Interaction Design and Children*, pages 207–218. ACM, 2018.

[42] Sung Wook Yoon, Alan Fern, and Robert Givan. Ff-replan: A baseline for probabilistic planning. In *ICAPS*, volume 7, pages 352–359, 2007.

[43] Amy Zhang, Nicolas Ballas, and Joelle Pineau. A dissection of overfitting and generalization in continuous reinforcement learning. *arXiv preprint arXiv:1806.07937*, 2018.

[44] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI*, pages 1655–1661, 2017.

[45] Shangtong Zhang, Borislav Mavrin, Hengshuai Yao, Linglong Kong, and Bo Liu. Quota: The quantile option architecture for reinforcement learning. *arXiv preprint arXiv:1811.02073*, 2018.

[46] Yu Zheng, Furui Liu, and Hsun-Ping Hsieh. U-air: When urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1436–1444. ACM, 2013.

[47] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017.

[48] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.