UBC MLRG (Fall 2018): Reinforcement Learning

Machine Learning Reading Group (MLRG)

- Machine learning reading group (MLRG) format:
  - Each semester we pick a general topic.
  - Each week someone leads us through a tutorial-style lecture/discussion.
  - So it’s organized a bit more like a “topics course” than reading group.
- We use this format because ML has become a huge field.
- The last few years the topics have been organized by me.
  - This year it is going to be student-run.
  - We may also drift a bit more between different topics.

Machine Learning Reading Group (MLRG)

- I've tried to pack as much as possible into the two ML courses:
  - CPSC 340 covers most of the most-useful methods.
  - CPSC 540 covers most of the background needed to read research papers.

- This reading group covers topics that aren't yet in these course.
  - Aimed at people who have taken CPSC 340,
    and are comfortable with 540-level material.

- This may change now that we have more ML faculty.

Recent MLRG History

- Topics covered in recent tutorial-style MLRG sessions:
  - Summer 2015: Probabilistic graphical models.
  - Fall 2015: Convex optimization.
  - Winter 2016: Bayesian statistics.
  - Summer 2016: Miscellaneous.
  - Fall 2016: Deep learning.
  - Winter 2016: Reinforcement learning.
  - Summer 2017: Online, active, and causal learning.
  - Fall 2017: Deep learning meets graphical models.
  - Winter 2018: Parallel and distributed machine learning.
  - Fall 2018: Reinforcement learning.

Why Reinforcement Learning?

https://www.youtube.com/watch?v=Ih8EfvOzBOY
https://www.youtube.com/watch?v=SH3bADiB7uQ
https://www.youtube.com/watch?v=nUQsRPJ1dYw

DeepLoco

Building up to Reinforcement Learning

- Reinforcement learning (RL) is very general/difficult:
  - It includes many other machine learning problems as special cases.
- Good introductory book:
  - "Introduction to Reinforcement Learning" by Sutton & Barto.
- Other names for reinforcement learning:
  - Approximate dynamic programming.
  - Neurodynamic programming.
  - Control theory.
- To build up to RL, let's start with supervised learning:
  - Introduce notation, and discuss ways RL is harder.

Supervised Learning

- Supervised learning notation:
  - We have input features x^t.
  - There are possible outputs y^t.
  - We have a loss function $L(x^t,y^t)$.
    - E.g., loss of 0 if you classify correctly and loss of 1 is you classify incorrectly.

- Reinforcement learning notation:
  - The features are referred to as states s^t.
  - The outputs are referred to as actions a^t.
  - The (negative) loss function is called the reward $r(s^t,a^t)$.
    - E.g., reward of 0 if you classify correctly and reward of -1 if you classify incorrectly.

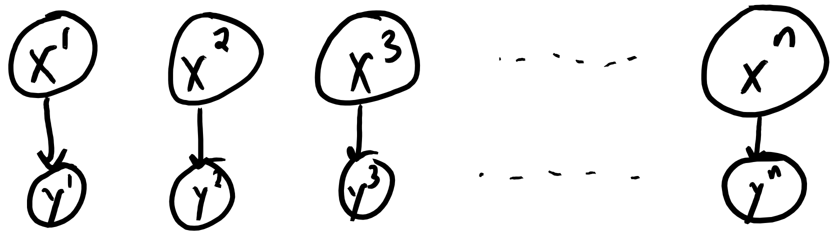
Supervised Learning

- Supervised learning training phase:
  - We have ‘n’ training examples, we can do whatever we want with them.
  - The output of training is a classifier: maps from x^t to y^t.
  - This is called a policy in RL: policies map from s^t to a^t.
- Goal: classifier minimizes loss ⇔ policy maximizes reward

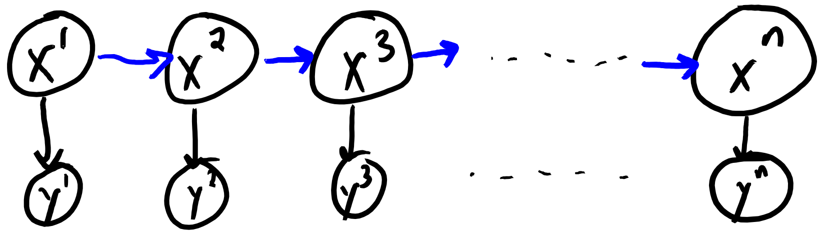
- Some models give score for each label:
  - For example, softmax gives probability of each y^t given x^t.
  - This is a Q function: $Q(s^t,a^t)$ is “value” of action a^t in state s^t.
  - Given a policy, we can define the value function $V(s^t)$ as “value” given policy (which may be deterministic or stochastic).

State-Space Models

- In standard setup, the x^t are IID samples:



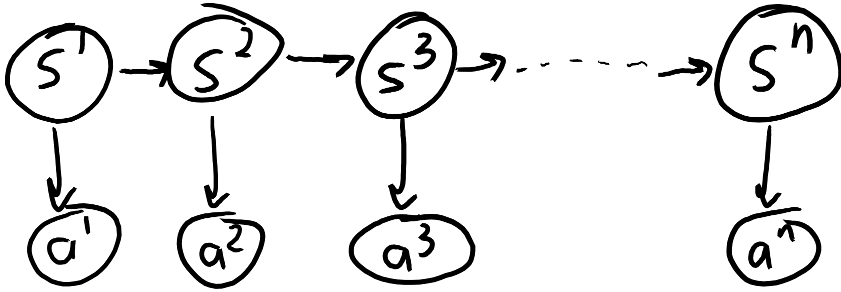
- In state-space models, the x^t come from a Markov chain:



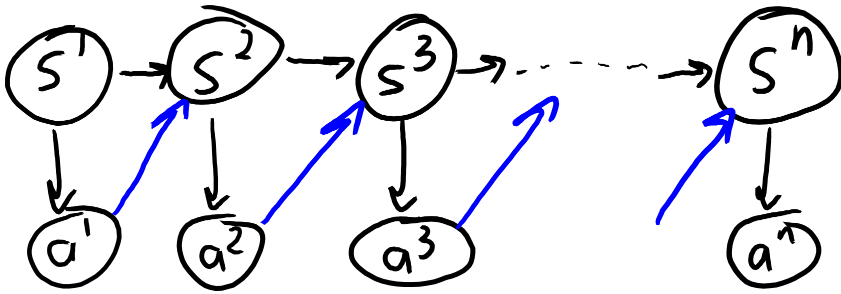
  - Value of x^t depends on the value of x^{t-1}.
  - We obtain IID samples in the special case of no dependencies.
  - Learning in this fully-observed DAG is pretty similar.

Markov Decision Processes (MDPs)

- State-space model in RL notation



- In Markov decision processes (MDPs), s^t also depends on a^{t-1}.
  - The action affects the value of the next state.
    - Here we need planning:
      - Choose actions that will lead to future states with high reward.



  - In MDPs we assume we have the "model":
    - Know all rewards $r(s^t,a^t)$ and transition probabilities $p(s^t \mid s^{t-1}, a^{t-1})$.
  - Given "model", we can find optimal values/policy by dynamic programming:
    - Value iteration and policy iteration.

Reinforcement Learning

- Reinforcement learning is MDPs when we don't know the "model".
  - All we can do is take actions and observe states/rewards that result.

- We need to simultaneously solve three problems:
  - We need to solve a supervised learning problem, $r(s^t,a^t)$.
  - We need to discover dynamics of a state-space model, $p(s^t \mid s^{t-1}, a^{t-1})$.
  - We need to plan an MDP policy maximizing long-term reward, $s^t \rightarrow a^t$.
- All while working with simulations.

- Unfortunately, this combination gives a few more challenges…

Active Learning

- Let's go back to the basic supervised learning setting:
  - Features s^t are just IID samples.

- Active learning considers the following variation:
  - The training examples are unlabeled.
  - The learner can query the user to label a training example s^t.
  - Goal is to do well with a fixed budget of queries.

- The fixed budget means we can't visit all features/states.
  - Here we need exploration: which states do we visit to learn the most?

Online Learning and Bandit Feedback

- In online learning there is no separate training/testing phase:
  - We receive a sequence of features/states s^t.
  - We have to choose prediction/action a^t on each example as it arrives.
  - Our “score” is the average loss/reward over time.
  - Here we need to predict well as we go (not at the end).
    - You pay a penalty for trying bad actions as you are learning.

- A common variation is with bandit feedback:
  - We only observe the reward function $r(s^t,a^t)$ for actions a^t that we choose.
  - Here we have an exploration vs. exploitation trade-off:
    - Should we explore by picking an a^t we don’t know much about?
    - Should we exploit by picking an a^t that gives high reward?

Causal Learning

- Causal learning:
  - Observational prediction:
    - Do people who take Cold-FX have shorter colds?
  - Causal prediction:
    - Does taking Cold-FX cause you to have shorter colds?
  - Counter-factual prediction:
    - You didn’t take Cold-FX and had long cold, would taking it have made it shorter?
- Here we need to learn effects of actions.
  - Including predicting effects of new actions.
- We may not control the actions: off-policy learning.
  - Actions are often randomized, but still want to find best actions.

Reinforcement Learning

- Reinforcement learning is MDPs when we don't know the "model".
  - All we can do is take actions and observe states/rewards that result.
- We need to consider:
  - Modeling how (s^t,a^t) combinations affects reward (supervised learning)
  - Learning how (s^t,a^t) affects s^{t+1} (state-space models, causality).
  - Planning for long-term reward (MDPs).
  - Exploring space of states and actions (active learning, bandit feedback).
- Two common frameworks:
  - Monte Carlo methods collects a lot of simulations to turn it into an MDP.
  - Temporal-difference learning considers online prediction as you go.
    - Need to consider exploration vs. exploitation, penalties for trying bad actions.

Related Problems

- Inverse reinforcement learning, apprenticeship learning, etc.:
  - Learning from an expert without an explicit reward function.
- Hidden state-space models:
  - The actual state is hidden, and x^t is just an observation based on the state.
  - Hidden Markov models, Kalman filters, LQR control.
- Partially-observed MDPs (POMDPs):
  - MDPs and reinforcement with hidden state-space model.
  - Hard even when you know the "model".

Schedule

Date	Topic	Presenter
Oct 15	Motivation/Overview	Mark
Oct 22		Yifan
Oct 29	Bayesian GANs 4 RL	Christian?
Nov 5		Sharan
Nov 12	Causality	Jason
Nov 19	(Bayes RL?)	Aaron
Nov 26	... Verification	Boyan
Dec 3		Wilder
Dec 10	Bandit Stuff (prob)	Mehrdad
Dec 17	Alpha Go	Vaden

3 NIPS

something something