Best-First Rao-Blackwellization

J Chiu

March 5, 2022

Gradient Estimation

- Goal: optimize

$$\max_{\theta} \sum_{x} p_{\theta}(x) f(x)$$

  (optimizing wrt parameters of f is easy)
- Discrete x, expensive evaluation of $f(x)$
  - eg f is a giant neural network

Gradient Estimators

- Score-function estimator (SFE)

$$\nabla_\theta \sum_x p_\theta(x) f(x) = \mathbb{E}_{p_\theta(x)} \left[f(x) \nabla \log p_\theta(x) \right]$$

  - High variance, consistent
  - Requires multiple evaluations, better with more compute
- Continuous relaxation, $x \approx g(\theta, \epsilon)$

$$\nabla_\theta \sum_x p_\theta(x) f(x) \approx \mathbb{E}_{p(\epsilon)} \left[\nabla_\theta f(g_\theta(\theta, \epsilon)) \right]$$

  - Biased, lower variance
  - Requires low number of evals, less benefit from more compute
  - Requires relaxable f
  - Stochastic softmax tricks (SST)
- Focus on improving SFE
  - SFE under-explored
  - Better with more compute is attractive

Rao-Blackwellization

- Too expensive to enumerate over all x (f is expensive)
- Reduce effective width of distribution via sub-sampling (red)
- Enumerate over green portion

p(1)	p(2)	p(3)	p(4)
f(1)	f(2)	f(3)	f(4)

Rao-Blackwellization of score function estimator

- Condition on values of $f(x)$ for $x \in S$

$$
\begin{aligned}
&\nabla_\theta \sum_x p_\theta(x) f(x) \\
&= \mathbb{E}_{p_\theta(x)} \left[f(x) \nabla \log p_\theta(x) \right] \\
&= \sum_{x \in S} p(x) f(x) \nabla \log p_\theta(x) + \mathbb{E}_{p_\theta(x \notin S)} \left[f(x) \nabla \log p_\theta(x) \right]
\end{aligned}
$$

- Conditioning on all values $f(x)$ = no variance
- Want S to have x with high $p(x)f(x)$
  - Need to compromise with high $p(x)$
  - Use heuristic estimate of $f(x)$?
  - What if x is structured?

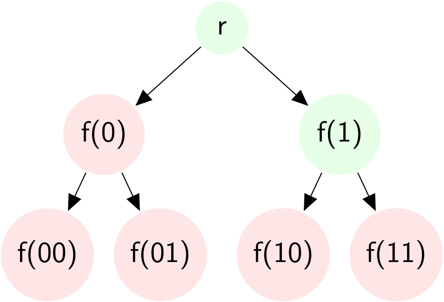
Structured Setting

- In the flat setting, only cared about width
- In structured setting, can also approximate depth

Flat

p(1)	p(2)	p(3)	p(4)
f(1)	f(2)	f(3)	f(4)

Structured



Rao-Blackwellization of structured SFE

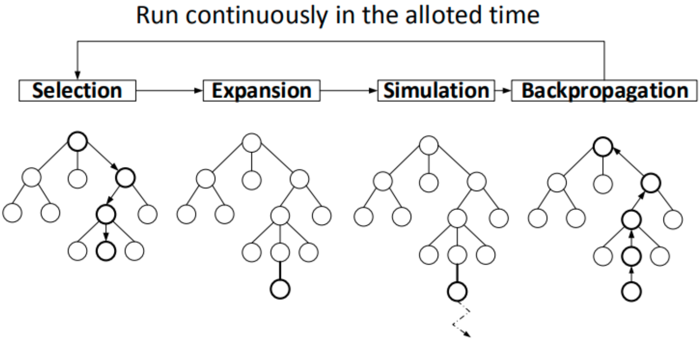
- Maybe f first-order additively decomposable over $x = x_{1:T}$?
- Cost-to-go estimate $\mathbb{E}_{x_{t+1:T}}\left[f(x_{t+1:T})\right] \approx \tilde{V}(x_{1:t})$
- Condition on values of $f(x_{1:t})$ for $x_{1:t} \in S$ (set of prefixes)

$$
\begin{aligned}
&\nabla_\theta \sum_x p_\theta(x) f(x) \\
&= \mathbb{E}_{p_\theta(x)}\left[f(x) \nabla \log p_\theta(x)\right] \\
&\approx \sum_{x_{1:t} \in S} p(x)(f(x_{1:t}) + \tilde{V}(x_{1:t})) \nabla \log p_\theta(x_{1:t}) \\
&\qquad + \mathbb{E}_{p_\theta(x \notin S)}\left[f(x) \nabla \log p_\theta(x)\right]
\end{aligned}
$$

Best-first Rao-Blackwellization

- Modern MCTS
  - Prior $p(x)$ (limits width)
  - Cost-to-go estimate $\tilde{V}(x)$ (limits depth)
  - Search procedure that links the two
- Proposal
  - In many cases, already have the prior $p(x)$
  - Estimate cost-to-go with continuous relaxation or cheaper problem-dependent weaker model (learned value function approx)
  - Link the two by marginalization

BF RB


Run continuously in the alloted time
Selection
Expansion
Simulation
Backpropagation

Minimal Experiment

Approximate DP when exact is tractable

- Depth approx: Learn cost-to-go estimate to bound depth in 32k state HMM
  - HMM with forward algo that doesnt go all the way to end
  - Bounded width = prior not important
  - Experiment with continuous relaxation + learned $\tilde{V}$ for suffix
- Width approx: Utilize prior in large-state HMM
  - Limit num states considered at each time step
- Approximations for width + depth
  - Probably more useful in PCFGs or other more interesting models (massive switching, AR latent)

Random thoughts

- For exp fam, is this an instance of hypernetworks (ie predicting gradient)?

Citations I