AAAI-19

박종의

January 2019

1 January 31st

1.1 Bayesian Learning and Probabilistic Graphical Model 2

1.1.1 Bayesian Graph Convolutional Neural Networks for Semi-Supervised Classification

- Represent uncertainty of the underlying data during graph building
- Bayesian framework to account graph uncertainty
- Bayesian Neural Networks:
  - Treat network weights as RVs
  - Compute posterior by variational inference
- High-level Algorithm
  - Train a graph generation model given the currently observed graph
  - Sample graphs from the learned graph generation model
  - Compute posterior of GCNN weights
  - Average over multiple samples
- Performance boost on the citation dataset
- Tested robustness by attacking the graphs(slightly perturbing the graph topology)
- Conclusion
  - Works well even With limited training data
  - Resilient to graph attacks
  - Can represent uncertainty
  - Can incorporate a variety of other graph generation/learning algorithm

1.1.2 Learning Logistic Circuits

- PSDD + SPN great at learning densities
- Just want to classify. No need to compute the joint distribution
- Probabilistic Circuits = Probabilistic analogue to logical circuits
- Logistic Circuit = Probabilistic Circuits + logistic function on final output
- How to learn the parameters of logistic circuits?
- First compute the active wires. Not all of them are active.
- Only consider them to reduce the amount of computation.
- Every LC can be reduce to an equivalent logistic regression $\Rightarrow$ Global Circuit Flow
- Even outperform CNNs in MNIST
- Converting PC to LC
  - The true and false weights are combined into one weight
  - The weight value is the log of the original ones.
- Highly interpretable: logical sentences become nodes
- Scalability: bottleneck in structure learning

1.1.3 Poster Spotlights

- Deep Convolutional SPN
  - SPN can be represented using CNN
- Understanding VAEs in Fisher–Shannon Plane
  - Fisher information and Shannon information are complementary to each other
- InfoVAE
  - Modified the VAE objective to address few problems that VAE has

1.2 Multiagent Systems 1

1.2.1 Fair Knapsack

- Mind experiment
  - Airline company has to choose what movies to ship in their in-plane entertainment system
  - Ask the customers what they prefer

- Formal definition
  - Voters $V = \{v_1, v_2, \ldots, v_n\}$
  - Items $A = \{a_1, a_2, \ldots, a_m\}$
  - Fixed budget B
  - Each item a has a cost $c(a)$
  - The voters have their own utility u_i for each item
  - Find a subset(knapsack) S that $c(S) \leq B$ and $u(S)$ is maximized
- Valuation Functions
  - Fair knapsack: tries to be proportionally fair

$$u_{\text{Fair}}(S) = \prod_{v_i \in V} (1 + \sum_{a \in S} u_i(a))$$

  - Diverse Knapasack

$$u_{\text{Div}}(S) = \sum_{v_i \in V} \max_{a \in A} u_i(a)$$

  - Individually Best Knapsack

$$u_{\text{IB}}(S) = \sum_{v_i \in V} \sum_{a \in S} u_i(a)$$

- Fairness comes with surprisingly high computational complexity.

1.2.2 Poster Spotlights

- Leveraging Observations in Bandits: Between Risks and Benefits
  - Multiple agents plaing the same bandit problems
  - Each agent can observe neighbours' actions but not rewards
  - Target-UCB: social-based optimism
- Learning to Teach in Cooperative MARL
  - How to coordinate and selectively share local knowledge
  - Phase I: Fixed teaching policy
  - Phase II: Train teaching policy based on learning rate(?)
- Multi-Winner Contests for Strategic Diffusion in Social Networks
  - Solicit as many efforts as possible from users in SN
  - Credit to each player that contributed task efforts and has made successful referrals

- Diffusion rewards
  - Allocate rewards proportionally
- General Robustness Evaluation of Incentive Mechanism Against Bounded Rationality Using Continuum-Armed Bandits
  - Incentive mechanisms
  - Robustness against bounded rationality of agents
  - Margin calculator
- Message-Dropout: An Efficient Training Method for MA-DRL
  - Drops messages and compensate it during execution phase
- Overcoming Blind Spots in the Real World
  - Mismatch between simulation and real world
  - Human feedback to solve blindspots
  - Human demonstration data to identify influential features
  - Deviation behaviors

2 Feburary 1st

2.1 Reinforcement Learning 2

2.1.1 State Abstractions as Compression in Apprenticeship Learning

- Abstraction: create a simple model of the environment, but perform well
- Compression vs. Value
- How small can we make the state space while preserving the performance?
- Computation complexity and sample complexity is proportional to the state space size
- Rate–Distortion theory
  - Source $\rightarrow$ Encoder $\rightarrow$ Decoder $\rightarrow$ Destination
  - Rate: number of bits used in your representation
  - Distortion: measured by some distortion metric
  - Lower bound on the product of rate and distortion
  - Can be solved by Blahut–Arimoto algorithm
- Information Bottleneck Theory
  - Relevance variable

- Latent representation must capture important features

- Given expert demonstrations, what's the best state abstraction?
- Devised an objective easier to optimize

2.1.2 Towards Better Interpretability in DQNs

- Interpretability
  - Local explanation: Given input
  - Global explanation: Regardless of input
- Keys initialized randomly and learned via backprop
- Loss functions
  - Bellman Error
  - Distributional Error
  - Reconstruction Error
  - Diversity Error

2.1.3 On Reinforcement Learning for Full-length Game of Starcraft

- Not much previous works on full-length SC games
- Huge state and action space + Long term sparse rewards $\Rightarrow$ Makes things super difficult
- Leveraged different level of abstractions
- Examples of low-level abstractions
  - Build(what, where)
  - Produce(what)
- Controller chooses what sub-policy to execute
- Macro-actions learned from human demonstrations

2.2 Reinforcement Learning 3

2.2.1 Fully Convolutional Network with Multi-Step RL for Image Processing

- Contributions
  - RL with pixel-wise rewards
  - Novel approach for image processing tasks
  - Comparable with SOTA

- Each pixel and its neighbors' value is considered as a state
- Actions are conventional tools for image processing, e.g. box filter
- Number of agents too large to deploy existing MARL approaches
- Modified A3C to be fully convolutional(?)
- Reward map convolution(?) to consider future states of neighbor pixels

2.2.2 Model-Free IRL using MLE

- Bayesian IRL
- Contributions
  - Model-Free
  - Q-averaging, Q-softmax
  - Real-world freeway merging problems
- Stationary policy assumption

$$\log L(\theta;\tau) = \log \prod_{t=1}^{N} \pi_\theta(s_t, a_t) = \sum_{t=1}^{N} \log \pi_\theta(s_t, a_t)$$

2.2.3 SUM: Self-Supervised Mixture-of-Experts by Uncertainty Estimation

- Key ideas
  - Uncertainty Estimation
  - Mixture-of-Experts
  - Self-supervision
  - Decayed Mask ER
- Uncertainty-Enhanced Muti-head DDPG
  - Multi-head: Q-value + Q-variance
  - Trained by optimizing negative-log-likelihood
  - Robust to task shift
- Mixture-of-Experts
  - Gating function to select the expert
  - Trained in a self-supervised manner
- Self-supervised learning
  - Ground truth constructed by softmax gating(?)

- Train gating function using MSE
- Activate experts with high uncertainty
- Stopped when an expert masters a task: super high mean and low variance

2.2.4 Off-Policy DRL by Bootstrapping the Covariate Shift

- Off-policy TD esitmation operators may diverge (work by Baird)
- Covariate shift
- Hallak proposed an operator whose fixed point is the covariant shift
  - Every scalar multiple is a fixed point
  - Projection into the simplex to make it unique
- Discounted covariate shift that has single nondegenerate FP
- If γ is small, we have convergence guarantees
- Experiments with random behavior policy
- Modified C51 to have two heads: one for value, another for covariate shift
- Used covariate shift to modify the sampling rates

2.2.5 Model Learning for Lookahead Exploration in Continuous Control

- Random exploration can be unsafe and data collections may be expensive
- Problems of HRL
  - low-level skill set may not contain key low-level policies
  - modularization degrades performance
- Look-ahead exploration
- Coarse skill dynamics model: predict terminal state using current state and goal
- Able to recover from bad skill sets
- Can perform well even when the state dynamics model is bad