L25 – Learning and learning!

May 12, 2025

These are lpk's informal notes
3 relevant kinds of learning

- models : "synthetic"
- speed-up search / planning / infr : "analytic"
- amortized inference avoid online interest : "analytic"

1 Model learning

Propositional logic

- given many exmaple possibly partial interpretations
- discover some clauses that are always true algorithms: "rule mining"
- generate and test (unsupervised)
- pick one proposition and try to find sparse predictor as logical function of others (e.g. as decision tree)

First-order learning Similar idea. inductive logic programming

```
child(mary, bob).
child(bob, pat).

parent(bob, mary).
grandparent(pat, bob).

grandparent(A, B) :-
  parent(A, C),
  parent(C, B).

parent(A, B) :-
  child(B, A)
```

Can also do things like sorting algorithms!

```
sorted([1, 4, 6]).
not sorted([4, 2, 9]).
```

Top-down v bottom-up Bottom-up : Consume single data point at a time, move upward in a lattice
Unification : P(X, Y, X), P(a, Z, H) yields most general specialization P(a, Y, a)
Inverse unification : P(a, z, a) , P(b, 1, b) yields least general generalizer P(X, Y, X)
Top-down : first-order decision-tree learning e.g. FOIL Tests can be quantified: Ex. P(X, a)

Bayes Nets

1. Given structure, learn parameters : counting + laplace correction
2. Learn structure
   - Try to detect conditional independence relations
   - Local search in structure space

   Max likelihood of data but pressure complexity

   $$argmax_M P(D|M) - \lambda|M|$$

   Local greedy search:

   - don't have to recompute all the tables
   - respect equivalence classes of structures

   Chow/Liu
3. Learn parameters when some values are unobserved optimize $\sum_h P(\theta, h|D)$

   EM: $P(h|\theta, D)$, $\theta = argmax P(D, h|\theta)$

   Parameter estimation for a Hidden Markov Model

   Can also do gradient descent

   Can also add hidden structure, nodes, etc.

Markov Random Fields Counting doesn't work. Really just do gradient descent on likelihood.

Learning PDDL

- Roughly an ILP problem
- Look at what fluents changed (these have to be results)
- Find examples with conforming results
- Intersect the preconditions

Something about NSRTs if we have time

- grounding classifiers
- continuous parameters
- samplers

Learning an MDP

- If everything is small, it's just counting (like BN)
- If big, then we can use a neural network, but how to learn to predict a probability distribution?
  - Parametric representation with an "calibrated" loss function: logistic regression if discrete; or output $\mu\Sigma$. Train with samples of (x, y)
  - Generators: map noise and x into samples of P(Y — x) GAN, VAE, diffusion model
- If we only have a generator, then we could use it to make a simulator and do RL.
- Learn a big joint generative model (s, a, r, s, a, r, s, a, r, s) ...
  - Generate from it, conditioned on high reward, and s = s0
  - Need to do some kind of search at generation time EBM or diffusion
- Value iteration networks (instance of general strategy)

- Find an algorithm that works well, but models hard to specify
- Rather than estimating models using likelihood, but it all into a network and train end-to-end to perform well, coming up, implicitly, with parameters

VIN: input is map and your location, output is action Box includes transition model + VI algorithm Differentiate the whole thing! (T is convolutional in that case)

2 Speed-up learning

We have the model (learned or hand-written). We can do inference / search, but it's slow So: let's learn to reason more efficiently

- heuristic function / value function
- successor ordering in search (action distribution)
- value / var ordering in CSP
- search control in FOL theorem provers
- generating decompositions / abstractions (PLOI / CAMP)
- Alpha Go / Zero

Amortized inference Learn to answer queries of a specialized kind (e.g. always same query and evidence variables)

Can frame as supervised learning, but may be useful to keep some aspects of the inference algorithm, to help generalization

neural logic machines: Learn to infer truth value of complex FOL expression

3 What classes to take next?

Not exhaustive! Lots of options!

Logic

- 24.141 Logic I
- 24.242 Logic II
- 24.251 Modal Logic
- 24.251 Intro to Philosophy of Language
- 24.253 Philosophy of Mathematics
- 18.510 Intro to Mathematical Logic and Set Theory
- 18.515 Mathematical Logic

Probabilistic inference and statistics

- 6.7480 Information Theory: From Coding to Learning
- 6.7800 Inference and Information
- 6.7810 Algorithms for Inference
- 6.7830 Bayesian Modeling and Inference
- 14.38 Inference on Causal and Structural Parameters
- 14.39 Large-Scale Decision-Making and Inference
- 15.C08 Causal Inference

Decision and Game thy

- 6.3950 AI, Decision Making, and Society
- 16.410 Principles of Autonomy and Decision-Making (maybe too much overlap)
- 16.412 Cognitive Robotics
- 16.420 Planning Under Uncertainty (maybe too much overlap)
- 6.5340 Topics in Algorithmic Game Theory
- 14.12 Economic Applications of Game Theory
- 14.16 Strategy and Information
- 14.126 Game Theory
- 6.3260 Networks
- 14.163 Algorithms and Behavioral Science

Cognition and Philsophy

- 9.66 Computational Cognitive Science
- 24.233 Rationality
- 24.XXX AI and Rationality (to be approved); to be offered by lpk and Brian Hedden in Fall 25

Robotics

- 2.160 Identification, Estimation, an dLearning
- 2.165 Robotics
- 6.8200 Sensorimotor learning
- 6.4210 Robotic Manipulation
- 16.412 Cognitive Robotics

Optimization

- 6.C57 Optimization Methods