Learning Graphical Concepts

Kevin Ellis, Eyal Dechter, Ryan P. Adams, Joshua B. Tenenbaum

December 11, 2013
What do humans construct?
What *graphical models* do humans construct?

![Diagram of graphical models](image)

*Figure 1. iHMM Graphical Model*
Graphical Models as Programs

Humans construct graphical models with a compositional structure and repeating motifs. We learn these structured motifs using by learning distributions over programs, given a set of example graphical models.

*Distribution over programs* → *Distribution over graphical models*
The CEC Program Induction Algorithm

Two key ideas:
- **Bootstrap learning**: reuse parts of solutions to easier problems to solve harder problems.
- **Incremental program synthesis**: create program $\rho_n : \tau$ by composing subprograms $e^i_n : \tau \rightarrow \tau$

Formalize as inference via EM in a hierarchical Bayesian model. Updates give iterative Explore/Compress procedure

- $G$: Stochastic grammar over programs
- $\rho_n$: Program for $n^{th}$ graphical model
- $\ell_n$: # of subprograms in $\rho_n$
- $t_n$: $n^{th}$ graphical model
Explore Step (E of EM)

Objective: Estimate
\[ q_n(\rho_n = e_n^\ell \circ \cdots \circ e_n^1) = P(t_n|\rho_n) \prod_{j \leq \ell_n} P(e_n^j|G) \]

Option 1: MCMC with transition kernel
\[ T(e_{n+1}^{i+1} \circ \rho_n^i|e_n^i \circ \rho_n^i) \propto P(t_n|e_{n+1}^{i+1} \circ \rho_n^i)P(e_{n+1}^{i+1}|G) \]
which serves as a proposal for \( q_n \) and the proposal weights are given by
\[ w(e_n^i \circ \cdots \circ e_n^1) = \left[ \prod_{j<i} P(t_n|e_n^j \circ \cdots \circ e_n^1) \right]^{-1} \]

Option 2: Beamed search
Compress Step (M of EM)

Minimize

\[
-\ln P(G) + \sum_n \mathbb{E}_{q_n} \left[ -\ln P(\rho_n | G) \right]
\]

MDL of grammar + Expected MDL of \(n^{th}\) program

\[\rightarrow Compression!\]

\(G\) is a stochastic grammar that consists of a weighted set of constant subtrees (like an Adaptor Grammar). Optimal \(G\) given if subtree \(t\) is in \(G\)

iff

\[\sum_n \mathbb{E}_{q_n} [c(t, \rho_n)] \geq \lambda\]

where \(c(t, \rho_n)\) is the number of times subtree \(t\) occurs in \(\rho_n\) and \(\lambda\) is a regularization coefficient.
Results on Graphical Models

- Set of twelve synthetic test cases (ising models, rings, Hidden Markov Models, etc.). CEC completely solved eight tasks and partially solved the other four.
- Simpler graphs bootstrap learning of complex graphs. For example, learning a Markov Chain bootstraps the learning of the ising model.
Future work: Building towers

Use program induction to construct plausible buildings. Programs evaluate to instructions for building a tower, likelihood is a surrogate for tower’s objective function (height, stability, etc)
Conclusion

- Programs offer an expressive, compositional representation for constructive machine learning.
- Bootstrap learning + incremental program synthesis makes program induction more tractible

Future directions:
- Better program induction algorithms; currently very brittle (use techniques from program analysis?)
- Further applications of program induction to constructive machine learning