Help

Learning Latent-Variable Models of Natural Language

Percy Liang

Collaborators: Michael Jordan, Dan Klein, Daniel Hsu, Sham Kakade

ISL Colloquium
October 18, 2012

Some problems in natural language semantics...

1

Relation extraction

On Monday, John Doe, president of Doe Inc., signed the agreement... → president(doe_inc, john_doe_3) sign(john_doe_3, agr_17, 2012-09-26)

Phenomena:

- Facts can be expressed in many ways
- Meaning depends on context
- Facts are inter-related

Current status:

- Text (New York Times, Wikipedia: 1.5M documents)
- Database (Freebase: 600M facts, 20M entities, 50K relations)
- Learn from distant supervision (given only text and fact database)

2

Question answering

Database + *What is the capital of the largest state by area east of Mississippi?* → Lansing

Phenomena:

- Meaning is a program/logical form/database query
  $\mathbf{capital}(\mathbf{argmax}(\lambda x.\, \mathbf{state}(x) \wedge \mathbf{eastOf}(x, \mathbf{MS}), \lambda x.\, \mathbf{area}(x)))$
- Program is derived (mostly) compositionally from the words

Current status:

- Can train reliable semantic parsers in limited domains [Liang, Zettlemoyer, etc.]
- No large question-answering datasets yet, need to learn from indirect signals (raw text, search results)

3

Following instructions

Preheat oven to 350 degrees F (175 degrees C). In a large mixing bowl, cream the butter and the sugar... →
preheat(oven, 350)
b = takeOut(mixingBowl)
add(butter, b)
add(sugar, b)
...

Observations:

- Language refers to a changing world
- Words and actions at different levels of abstraction

Current status:

- Reinforcement learning to read manuals to play games [Branavan/Barzilay]
- Still need to combine compositional semantics with context-dependent interpretation

4

Desiderata

1. **Rich models** to capture all the linguistic phenomena.

2. Learn models from **weak supervision** to scale up.

5

Semantic parsing

(joint work with Michael Jordan and Dan Klein)

Semantic parsing

What is the largest city in a state bordering California?

↓

$\mathbf{argmax}(\lambda c.\, \mathbf{city}(c) \wedge \exists s.\, \mathbf{state}(s) \wedge \mathbf{loc}(c, s) \wedge \mathbf{border}(s, \mathbf{CA}), \lambda x.\, \mathbf{population}(x))$

↓

Phoenix

Forms of supervision

Expensive: logical forms

[Zelle & Mooney, 1996; Zettlemoyer & Collins, 2005]

[Wong & Mooney, 2006; Kwiatkowski et al., 2010]

What is the most populous city in California?
$\Rightarrow \mathbf{argmax}(\lambda x.\, \mathbf{city}(x) \wedge \mathbf{loc}(x, \mathbf{CA}), \lambda x.\, \mathbf{pop.}(x))$
How many states border Oregon?
$\Rightarrow \mathbf{count}(\lambda x.\, \mathbf{state}(x) \wedge \mathbf{border}(x, \mathbf{OR})$

Cheap: answers

[Clarke et al., 2010]

[Liang et al., 2011]

What is the most populous city in California?
⇒*Los Angeles*
How many states border Oregon?
⇒*3*

Learning setup

Input:

Questions $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}$
Answers $y^{(1)}, \dots, y^{(n)}$

Output:

Parameter estimate: θ

New answers: $\mathbf{x}^{(n+1)} \rightarrow y^{(n+1)}$

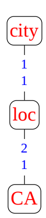
Main question: can we learn without logical forms?

Semantic parsing

Because logical forms z are latent, can choose the internal representation for z

Dependency-based compositional semantics (DCS):

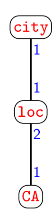
cities in California →



→ San Francisco
Los Angeles
...

Dependency-based compositional semantics

DCS tree



Constraints

$c \in \texttt{city}$

$c_1 = \ell_1$

$\ell \in \texttt{loc}$

$\ell_2 = s_1$

$s \in \texttt{CA}$

Database

city

San Francisco
Chicago
Boston
...

loc

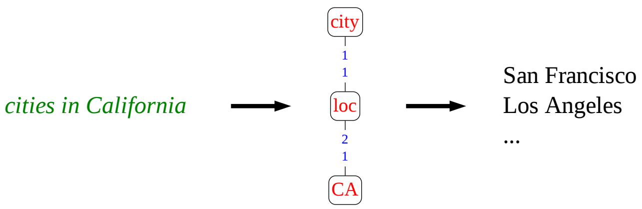
Mount Shasta	California
San Francisco	**California**
Boston	Massachusetts
...	...

CA

California

Basic DCS defines a constraint satisfaction problem

Semantic parsing

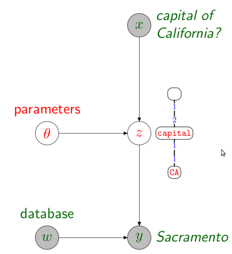


Two types of features $\phi(\mathbf{x}, z) \in \mathbb{R}^d$:

Translation (e.g., count of *in* mapping to loc)
Parsing (e.g., count of city connected to loc via 11)

12

Probabilistic model



Semantic parsing:

$$\mathbb{P}_\theta(z \mid \mathbf{x}) = \frac{\exp\{\phi(\mathbf{x}, z)^\top \theta\}}{\sum_{z' \in \mathcal{Z}(\mathbf{x})} \exp\{\phi(\mathbf{x}, z')^\top \theta\}}$$

Execution:

$y = \mathbf{RunOnDatabase}(z)$

13

Two algorithmic challenges

Maximum likelihood is non-convex:

$$\log \mathbb{P}_\theta(y \mid x) = \log \sum_{z:\mathbf{RunOnDatabase}(z)=y} \mathbb{P}_\theta(z \mid x)$$

Space of logical forms is exponentially large:

$$\mathcal{Z}(x) = \mathbf{PossiblePredicatesForWord}(x)$$
$$\mathcal{Z}(\mathbf{x}) = \cup_{\mathbf{x}=\mathbf{st}} \cup_{s \in \mathcal{Z}(\mathbf{s}), t \in \mathcal{Z}(\mathbf{t})} \mathbf{Combine}(s, t)$$

14

Alternating algorithm

Iterate:

Use beam search with parameters $\boldsymbol{\theta}$ to approximate $\mathcal{Z}(\mathbf{x})$

Use L-BFGS to optimize approx. likelihood to update $\boldsymbol{\theta}$

beam $\tilde{Z}_\theta(\mathbf{x})$ ⟷ parameters $\boldsymbol{\theta}$

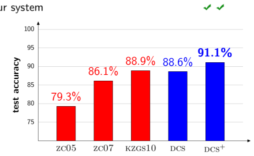
15

Experiments

GeoQuery dataset (600 training, 280 test)

On GEO, 600 training examples, 280 test examples

System	Description	Lexicon	Logical forms
ZC05	CCG [Zettlemoyer & Collins, 2005]	✗ ✗	✓
ZC07	relaxed CCG [Zettlemoyer & Collins, 2007]	✗ ✗	✓
KZGS10	CCG w/unification [Kwiatkowski et al., 2010]	✗ ✗	✓
DCS	our system	✓ ✗	✗
DCS+	our system	✓ ✓	✗



Differences: less supervision, different representation

16

Learning and search

beam $\tilde{Z}_\theta(\mathbf{x})$ ⟷ parameters $\boldsymbol{\theta}$

Bootstrapping effect:

- Initially, beam has no z with $\mathbf{RunOnDatabase}(z) = y$
- As parameters get better, beam search improves

17

Semantic parsing summary

What is the largest city in a state bordering California?

↓

$\mathbf{argmax}(\lambda c.\,\mathbf{city}(c) \wedge \exists s.\,\mathbf{state}(s) \wedge \mathbf{loc}(c, s) \wedge \mathbf{border}(s, \mathrm{CA}), \lambda x.\,\mathbf{population}(x))$

↓

Phoenix

- Answering complex questions requires deep representations (programs)
- Need to learn from weak natural supervision to scale up
- Empirical result: model latent programs, learning from just answers gets comparable results to logical forms

18

Unsupervised learning of latent-variable models

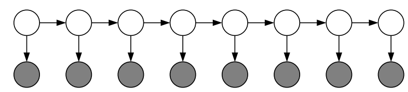
(joint work with Daniel Hsu, Sham Kakade)

19

Question: Can we develop theoretically-justifiable methods for learning complex latent-variable models?

20

Learning HMMs



$$\mathbb{P}_\theta(Z, X) = \prod_{j=1}^{\ell} T(Z_j, Z_{j-1}) O(X_j, Z_j)$$

k hidden states (Z_j), d observations (X_j)

Learning problem:

- Input: samples $X^{(1)}, \dots, X^{(n)} \sim \mathbb{P}_\theta(X)$
- Output: estimate of $\theta = (T, O)$

21

Unsupervised learning

Maximum likelihood estimator:

$$\hat\theta_{\mathrm{ml}} = \arg\max_\theta \log \mathbb{P}_\theta(X)$$

- NP-hard in general
- In practice, use EM with careful initializations, annealing, converges to local optima

Method of moments estimator:

- Moment equations: $\mu = M(\theta) = \mathbb{E}_\theta[m(X)]$
- Estimate the moments from observed data: $\hat\mu = \hat{\mathbb{E}}[m(X)]$
- Solve moment equations: $\hat\theta_{\mathrm{mom}} = M^{-1}(\hat\mu)$

22

[Anandkumar/Hsu/Kakade 2012]

Method of moments estimator for HMMs

Assume T and O have full rank. Then the following suffice:

- Observe $\mathbb{E}[X_1 X_2^\top] = OTO^\top$
- Observe $\mathbb{E}[X_1 X_2^\top (X_3^\top \eta)] = OT\mathbf{diag}(T^\top O^\top \eta) O^\top$

Solve moment equations using eigendecomposition to get (O, T).

Maximum likelihood:

- Statistically efficient
- Computationally inefficient

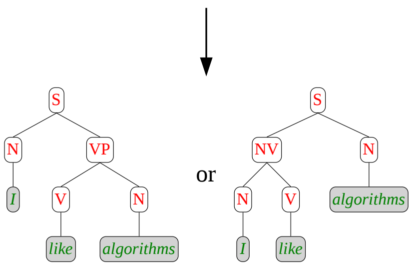
Method of moments:

- Statistically inefficient
- Computationally efficient

23

Random structures

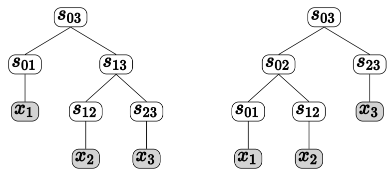
$\mathbf{x}$: *I like algorithms.*



z:

24

Parse trees



Observed $\mathbf{x}$:

- x_i: i-th word in the sentence

Latent z:

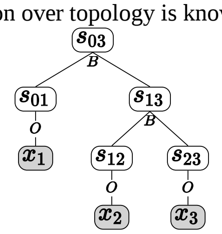
- **Topology**: tree topology (random)
- s_{ij}: state over span $[i : j]$

25

PCFG

Parameters θ:

- Emissions: $O \in \mathbb{R}^{d \times k}$
- Binary productions: $B \in \mathbb{R}^{k^2 \times k}$
- Assume distribution over topology is known



26

Standard PCFG is non-identifiable

Can we recover parameters given infinite data: $\mathbb{P}_{\theta^*}(\mathbf{x}) \Rightarrow \theta^*$?

Consider equivalence class:

$$\mathcal{S}(\theta_0) = \{\theta : \mathbb{P}_\theta(\mathbf{x}) \equiv \mathbb{P}_{\theta_0}(\mathbf{x})\}$$

Trivial: $|\mathcal{S}(\theta_0)| \geq k!$

Theorem: $|\mathcal{S}(\theta_0)|$ is infinite for PCFGs

"Proof" technique:

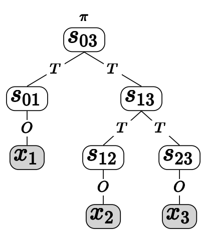
Sample a random parameter setting θ_0
Compute dimension of tangent space (rank of Jacobian)
Identifiable if dimension equals number of parameters

27

Restricted PCFG

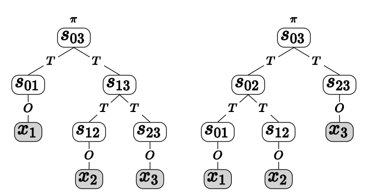
Parameters θ:

- Emissions: $O \in \mathbb{R}^{d \times k}$
- Left/right productions: $T \in R^{k \times k}$



28

Method of moments



Fixed tree:

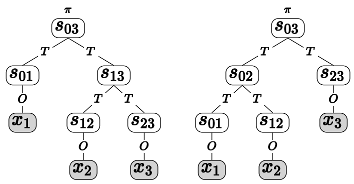
- eigendecomposition [Anandkumar/Hsu/Kakade 2012]

Random tree:

- unmixing + eigendecomposition [Hsu/Kakade/Liang 2012]

29

Random tree



Each moment matrix is mixture over topologies:

$$\mu_{12} = \mathbb{E}[x_1 \otimes x_2] = 0.5\Psi_1 + 0.5\Psi_2$$
$$\mu_{13} = \mathbb{E}[x_1 \otimes x_3] = 0.5\Psi_3 + 0.5\Psi_2$$
$$\mu_{23} = \mathbb{E}[x_2 \otimes x_3] = 0.5\Psi_3 + 0.5\Psi_1$$

Compound parameters Ψ are moments for fixed topologies (fingerprint of path through tree; much fewer than # topologies)

Random tree

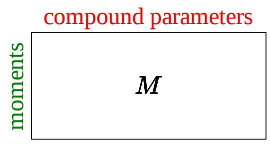
General mixing formula:

$$\mu = M\Psi$$

Unmixing algorithm:

- Estimate mixed moment matrices μ
- Compute mixing matrix M (using $\mathbb{P}(\text{Topology})$)
- Unmix compound parameters $\Psi = M^{-1}\mu$
- Call fixed tree algorithm on Ψ to recover $\theta = (T, O)$

More complex models?



- Technique relies on M having more constraints (moments) than variables (compound parameters)
- If allow different left/right production probabilities
  - $\Rightarrow$ too many compound parameters
  - $\Rightarrow$ mixing matrix M becomes rank deficient
- Algorithm not making efficient use of moments (know model identifiable via randomized identifiability checker)

Summary

1. Want to learn rich models of semantics from weak supervision (models with latent programs).

2. Learning is hard in latent-variable models, but heuristics can work empirically.

3. Want to develop more principled algorithms / understanding computational and statistical properties.

Thank you!