Learning with Intractable Inference and Partial Supervision

Jacob Steinhardt

Stanford University

jsteinhardt@cs.stanford.edu

September 8, 2015
Company officials refused to comment. He said the company would appeal.

公司官员拒绝对此发表评论。他表示该公司将提出上诉。
An Example

Company officials refused to comment. He said the company would appeal.

公司官员拒绝对此发表评论。他表示该公司将提出上诉。
An Example

Company officials refused to comment. He said the company would appeal.

公司官员拒绝对此发表评论。他表示该公司将提出上诉。

Statistical reasoning: aggregate data across sentences to reach conclusions.
An Example

Company officials refused to comment. He said the company would appeal.

公司官员拒绝对此发表评论。他表示该公司将提出上诉。

Statistical reasoning: aggregate data across sentences to reach conclusions. Computational reasoning: focus on easily disambiguated words first.
An Example

Company officials refused to comment. He said the company would appeal.

Statistical reasoning: aggregate data across sentences to reach conclusions. Computational reasoning: focus on easily disambiguated words first.

Tension: statistics wants to expose information (aggregation), while computer science wants to hide it (abstraction, adaptivity).
An Example

Company officials refused to comment. He said the company would appeal.

Company 官员拒绝对此发表评论。
他表示该公司将提出上诉。

Statistical reasoning: aggregate data across sentences to reach conclusions.
Computational reasoning: focus on easily disambiguated words first.

Tension: statistics wants to expose information (aggregation), while computer science wants to hide it (abstraction, adaptivity).

- Statistical inference is computationally intractable.
An Example

Company officials refused to comment. He said the company would appeal.

公司官员拒绝对此发表评论。他表示该公司将提出上诉。

Statistical reasoning: aggregate data across sentences to reach conclusions. Computacional reasoning: focus on easily disambiguated words first.

Tension: statistics wants to expose information (aggregation), while computer science wants to hide it (abstraction, adaptivity).

- Statistical inference is computationally intractable.
- How can we bring these two paradigms together?
1. Motivation

2. Formal Setting

3. Reified Context Models

4. Relaxed Supervision

5. Open Questions
Setting: Structured Prediction

input $x$: 火山

output $y$: volcanic
Setting: Structured Prediction

Goal: learn $\theta$ to maximize $\mathbb{E}_{x,y \sim D} [\log p_\theta(y | x)]$
Setting: Structured Prediction

- Input $x$: volcanic
- Output $y$: volcanic

- Goal: learn $\theta$ to maximize $\mathbb{E}_{x,y \sim D} [\log p_{\theta}(y | x)]$
- Structured output space $\mathcal{Y}$ — requires inference
Supervised Learning is Easy

Recall: want to maximize $\mathbb{E}[\log p_\theta(y \mid x)]$. 
Supervised Learning is Easy

Recall: want to maximize $\mathbb{E}[\log p_\theta(y \mid x)]$.

Suppose $p_\theta(y \mid x) \propto \exp(\theta^\top \phi(x, y))$. Then:
Supervised Learning is Easy

Recall: want to maximize $\mathbb{E}[\log p_\theta(y | x)]$.

Suppose $p_\theta(y | x) \propto \exp(\theta^\top \phi(x, y))$. Then:

$$\nabla_\theta \log p_\theta(y | x) = \phi(x, y) - \mathbb{E}_{\hat{y} \sim p_\theta(\cdot | x)}[\phi(x, \hat{y})].$$

- Given
- Inference
Supervised Learning is Easy

Recall: want to maximize $\mathbb{E}[\log p_\theta(y \mid x)]$.

Suppose $p_\theta(y \mid x) \propto \exp(\theta^\top \phi(x, y))$. Then:

$$\nabla_\theta \log p_\theta(y \mid x) = \phi(x, y) - \mathbb{E}_{\hat{y} \sim p_\theta(\cdot \mid x)}[\phi(x, \hat{y})].$$

Inference errors will be corrected by supervision signal ($\phi(x, y)$) over the course of learning.
Supervised Learning is Easy

Recall: want to maximize $\mathbb{E}[\log p_\theta(y \mid x)]$.

Suppose $p_\theta(y \mid x) \propto \exp(\theta^\top \phi(x, y))$. Then:

$$\nabla_\theta \log p_\theta(y \mid x) = \phi(x, y) - \mathbb{E}_{\hat{y} \sim p_\theta(\cdot \mid x)}[\phi(x, \hat{y})].$$

Inference errors will be corrected by supervision signal $(\phi(x, y))$ over the course of learning.

- In practice, anything reasonable (MCMC, beam search) works.
Supervised Learning is Easy

Recall: want to maximize $\mathbb{E}[\log p_\theta(y \mid x)]$.

Suppose $p_\theta(y \mid x) \propto \exp(\theta^\top \phi(x, y))$. Then:

$$\nabla_\theta \log p_\theta(y \mid x) = \phi(x, y) - \mathbb{E}_{\hat{y} \sim p_\theta(\cdot \mid x)}[\phi(x, \hat{y})].$$

Inference errors will be corrected by supervision signal ($\phi(x, y)$) over the course of learning.

- In practice, anything reasonable (MCMC, beam search) works.
- Conceptually, can use Searn (Daumé III et al., 2009) or pseudolikelihood (Besag, 1975) to obviate need for inference.
Supervised Learning is Easy

Recall: want to maximize $\mathbb{E}[\log p_\theta(y \mid x)]$.

Suppose $p_\theta(y \mid x) \propto \exp(\theta^\top \phi(x, y))$. Then:

$$\nabla_\theta \log p_\theta(y \mid x) = \phi(x, y) - \mathbb{E}_{\hat{y}\sim p_\theta(\cdot \mid x)}[\phi(x, \hat{y})].$$

Inference errors will be corrected by supervision signal ($\phi(x, y)$) over the course of learning.

- In practice, anything reasonable (MCMC, beam search) works.
- Conceptually, can use Searn (Daumé III et al., 2009) or pseudolikelihood (Besag, 1975) to obviate need for inference.

Approximate inference is easy in supervised settings.
Supervised Learning is Easy

Recall: want to maximize $\mathbb{E}[\log p_\theta(y \mid x)]$.

Suppose $p_\theta(y \mid x) \propto \exp(\theta^\top \phi(x, y))$. Then:

$$\nabla_\theta \log p_\theta(y \mid x) = \phi(x, y) - \mathbb{E}_{\hat{y} \sim p_\theta(\cdot \mid x)}[\phi(x, \hat{y})].$$

Inference errors will be corrected by supervision signal ($\phi(x, y)$) over the course of learning.

- In practice, anything reasonable (MCMC, beam search) works.
- Conceptually, can use Searn (Daumé III et al., 2009) or pseudolikelihood (Besag, 1975) to obviate need for inference.

Approximate inference is easy in supervised settings.

- Unless we care about estimating uncertainty (calibration, precision/recall)
Partially Supervised Structured Prediction

input $x$: Company officials refused to comment.
latent $z$: 公司 官员 拒绝 对此 发表评论。
output $y$:
Partially Supervised Structured Prediction

input $x$: Company officials refused to comment.

latent $z$: 公司 官员 拒绝 对此 发表评论。

output $y$: 公司 官员 拒绝 对此 发表评论。

Goal: learn $\theta$ to maximize $\mathbb{E}_{x,y \sim D}[\log p_{\theta}(y \mid x)]$
Partially Supervised Structured Prediction

- **input** $x$: Company officials refused to comment.
- **latent** $z$: 
- **output** $y$: 公司 官员 拒绝 对此 发表评论。

- Goal: learn $\theta$ to maximize $\mathbb{E}_{x,y \sim \mathcal{D}}[\log p_{\theta}(y | x)]$
- Where $p_{\theta}(y | x) = \sum_z p_{\theta}(y, z | x)$

Inference errors on $z$ get reinforced during learning. Inference often hardest (and most consequential) at beginning of learning!
Partially Supervised Structured Prediction

input $x$: Company officials refused to comment.

latent $z$: 公司 官员 拒绝 对此 发表评论。

output $y$: 公司 官员 拒绝 对此 发表评论。

- Goal: learn $\theta$ to maximize $\mathbb{E}_{x,y \sim D}[\log p_\theta(y | x)]$
- Where $p_\theta(y | x) = \sum_z p_\theta(y, z | x)$

Again assume $p_\theta(y, z | x) \propto \exp(\theta^\top \phi(x, z, y))$. Then

$$\nabla_\theta \log p_\theta(y | x) = \mathbb{E}_{\hat{z} \sim p_\theta(\cdot | x, y)}[\phi(x, \hat{z}, y)] - \mathbb{E}_{\hat{z}, \hat{y} \sim p_\theta(\cdot | x)}[\phi(x, \hat{z}, \hat{y})].$$

Inference errors on $z$ get reinforced during learning. Inference often hardest (and most consequential) at beginning of learning!
Partially Supervised Structured Prediction

- **input** $x$: Company officials refused to comment.
- **latent** $z$:
- **output** $y$: 公司 官员 拒绝对此发表评论。

Goal: learn $\theta$ to maximize $\mathbb{E}_{x,y \sim D} [\log p_{\theta}(y \mid x)]$

Where $p_{\theta}(y \mid x) = \sum_z p_{\theta}(y, z \mid x)$

Again assume $p_{\theta}(y, z \mid x) \propto \exp(\theta^\top \phi(x, z, y))$. Then

$$\nabla_{\theta} \log p_{\theta}(y \mid x) = \mathbb{E}_{\hat{z} \sim p_{\theta}(\cdot \mid x, y)}[\phi(x, \hat{z}, y)] - \mathbb{E}_{\hat{z}, \hat{y} \sim p_{\theta}(\cdot \mid x)}[\phi(x, \hat{z}, \hat{y})].$$

Inference errors on $z$ get **reinforced** during learning.
Partially Supervised Structured Prediction

input $x$: Company officials refused to comment.
latent $z$: 
output $y$: 公司官员拒绝对此发表评论。

- Goal: learn $\theta$ to maximize $\mathbb{E}_{x,y \sim \mathcal{D}}[\log p_{\theta}(y | x)]$
- Where $p_{\theta}(y | x) = \sum_z p(y,z | x)$

Again assume $p(y, z | x) \propto \exp(\theta^\top \phi(x, z, y))$. Then

$$\nabla_{\theta} \log p_{\theta}(y | x) = \mathbb{E}_{\hat{z} \sim p_{\theta}(\cdot | x, y)}[\phi(x, \hat{z}, y)] - \mathbb{E}_{\hat{z}, \hat{y} \sim p_{\theta}(\cdot | x)}[\phi(x, \hat{z}, \hat{y})].$$

Inference errors on $z$ get **reinforced** during learning.
Inference often hardest (and most consequential) at beginning of learning!
Two thrusts:

1. How can we *reify* computation as part of a statistical model?

2. How can we *relax* the supervision signal to aid computation while still maintaining consistent parameter estimates?
Related Work

Learning tractable models / accounting for approximations

- **sum-product networks** (Poon & Domingos, 2011)
- **max-violation perceptron** (Huang, Fayong, & Guo, 2012; Zhang et al., 2013; Yu et al., 2013)
- **fast-mixing Markov chains** (S. & Liang, 2015)
- **many others** (Barbu, 2009; Daumé III, Langford, & Marcu, 2009; Domke, 2011; Stoyanov, Ropson, & Eisner, 2011; Niepert & Domingos, 2014; Li & Zemel, 2014; Shi, S., & Liang, 2015)
Related Work

Learning tractable models / accounting for approximations

- **sum-product networks** (Poon & Domingos, 2011)
- **max-violation perceptron** (Huang, Fayong, & Guo, 2012; Zhang et al., 2013; Yu et al., 2013)
- **fast-mixing Markov chains** (S. & Liang, 2015)
- **many others** (Barbu, 2009; Daumé III, Langford, & Marcu, 2009; Domke, 2011; Stoyanov, Ropson, & Eisner, 2011; Niepert & Domingos, 2014; Li & Zemel, 2014; Shi, S., & Liang, 2015)

Improving expressivity of variational inference

- **combining with MCMC** (Salimans, Kingma, & Welling, 2015)
- **using neural networks** (Kingma & Welling, 2013; Mnih & Gregor, 2014)
Related Work

Learning tractable models / accounting for approximations

- **sum-product networks** (Poon & Domingos, 2011)
- **max-violation perceptron** (Huang, Fayong, & Guo, 2012; Zhang et al., 2013; Yu et al., 2013)
- **fast-mixing Markov chains** (S. & Liang, 2015)
- many others (Barbu, 2009; Daumé III, Langford, & Marcu, 2009; Domke, 2011; Stoyanov, Ropson, & Eisner, 2011; Niepert & Domingos, 2014; Li & Zemel, 2014; Shi, S., & Liang, 2015)

Improving expressivity of variational inference

- combining with MCMC (Salimans, Kingma, & Welling, 2015)
- using neural networks (Kingma & Welling, 2013; Mnih & Gregor, 2014)

Computational-statistical tradeoffs

- huge body of recent work (Berthet & Rigollet, 2013; Chandrasekaran & Jordan, 2013; Zhang et al., 2013; Zhang, Wainwright, & Jordan, 2014; Christiano, 2014; Daniely, Linial, & Shalev-Shwartz, 2014; Garg, Ma, & Nguyen, 2014; Shamir, 2014; Braverman et al., 2015; S. & Duchi, 2015; S., Valiant, & Wager, 2015)
Structured Prediction Task

input $x$: volcanic

output $y$: volcanic
Contexts Are Key
Contexts Are Key

DP: v o l c a
Contexts Are Key

DP:

beam search:
Key idea: contexts!

\[
\*o \overset{\text{def}}{=} \begin{cases} 
ao \\
bo \\
co \\
\vdots
\end{cases}
\]
Desiderata

- coverage (short contexts)
  - better uncertainty estimates (precision)
  - stabler partially supervised learning updates
Desiderata

- coverage (short contexts)
  - better uncertainty estimates (precision)
  - stabler partially supervised learning updates
Desiderata

- coverage (short contexts)
  - better uncertainty estimates (precision)
  - stabler partially supervised learning updates
Desiderata

- coverage (short contexts)
  - better uncertainty estimates (precision)
  - stabler partially supervised learning updates

- expressivity (long contexts)
  - capture complex dependencies
Desiderata

- coverage (short contexts)
  - better uncertainty estimates (precision)
  - stabler partially supervised learning updates

- expressivity (long contexts)
  - capture complex dependencies
Desiderata

- coverage (short contexts)
  - better uncertainty estimates (precision)
  - stabler partially supervised learning updates

- expressivity (long contexts)
  - capture complex dependencies
Desiderata

- coverage (short contexts)
  - better uncertainty estimates (precision)
  - stabler partially supervised learning updates

- expressivity (long contexts)
  - capture complex dependencies

← best of both worlds
Reifying Contexts

input $x$: $\text{v o l c a n i c}$

output $y$: $\text{v o o l c a n i c}$

Challenge: how to trade off contexts of different lengths?

⇒ Reify contexts as part of model!
Reifying Contexts

input $x$: VOLCANIC

output $y$: VOLCANIC

Challenge: how to trade off contexts of different lengths?

⇒ Reify contexts as part of model!
Reifying Contexts

Reified Context Models

input $x$: Y D I C A N I C

output $y$: v o l c a n i c

context $c$: v *o *ol *olc ......
Reifying Contexts

input $x$:  
\[
\begin{array}{cccccc}
V & o & I & C & A & N & I & C
\end{array}
\]

output $y$:  
\[
\begin{array}{cccccc}
V & o & l & c & a & n & i & c
\end{array}
\]

context $c$:  
\[
\begin{array}{cccccc}
v & *o & *ol & *olc & \ldots
\end{array}
\]
\[
\begin{array}{cccccc}
r & ro & rol & *olc
\end{array}
\]
\[
\begin{array}{cccccc}
v & ra & ral & ***c
\end{array}
\]
\[
\begin{array}{cccccc}
y & *o & *ol & ***r
\end{array}
\]
\[
\begin{array}{cccccc}
* & ** & *** & ****
\end{array}
\]

$C_1 \quad C_2 \quad C_3 \quad C_4$

Challenge: how to trade off contexts of different lengths?

$\Rightarrow$ Reify contexts as part of model!
Reifying Contexts

input $x$: v D I C CANIC

output $y$: vol c anic

context $c$: v *o *ol *olc ......
  r ro rol *olc
  v ra ral ***c
  y *o *ol ***r
  * ** *** ****

$C_1$ $C_2$ $C_3$ $C_4$

Challenge: how to trade off contexts of different lengths?
Reifying Contexts

input $x$: $\text{V D I C A N I C}$

output $y$: $\text{v o l c a n i c}$

context $c$: $\text{v *o *ol *olc}$

$\text{r ro rol *olc}$

$\text{v ra ral ***c}$

$\text{y *o *ol ***r}$

$\text{* *** **** ****}$

$C_1 \quad C_2 \quad C_3 \quad C_4$

“context sets”

Challenge: how to trade off contexts of different lengths?

$\rightarrow$ Reify contexts as part of model!
Reified Context Models

Given:

- context sets $C_1, \ldots, C_L$
Given:

- context sets $C_1, \ldots, C_L$
- features $\phi_i(c_{i-1}, y_i)$
Reified Context Models

Given:
- context sets $C_1, \ldots, C_L$
- features $\phi_i(c_{i-1}, y_i)$

Define the model

$$p_\theta(y_{1:L}, c_{1:L-1}) \propto \exp \left( \sum_{i=1}^{L} \theta^\top \phi_i(c_{i-1}, y_i) \right) \cdot \kappa(y, c)$$

 inconvenience
Reified Context Models

Given:
- context sets $C_1, \ldots, C_L$
- features $\phi_i(c_{i-1}, y_i)$

Define the model

$$p_\theta(y_{1:L}, c_{1:L-1}) \propto \exp \left( \sum_{i=1}^{L} \theta^\top \phi_i(c_{i-1}, y_i) \right) \cdot \kappa(y, c)$$

consistency
Reified Context Models

Given:
- context sets $C_1, \ldots, C_L$
- features $\phi_i(c_{i-1}, y_i)$

Define the model

$$p_\theta(y_{1:L}, c_{1:L-1}) \propto \exp \left( \sum_{i=1}^{L} \theta^\top \phi_i(c_{i-1}, y_i) \right) \cdot \kappa(y, c)$$

Graphical model structure:
Reified Context Models

Given:
- context sets \( C_1, \ldots, C_L \)
- features \( \phi_i(c_{i-1}, y_i) \)

Define the model

\[
p_\theta(y_{1:L}, c_{1:L-1}) \propto \exp \left( \sum_{i=1}^{L} \theta^\top \phi_i(c_{i-1}, y_i) \right) \cdot \kappa(y, c)
\]

Graphical model structure:
Reified Context Models

Given:
- context sets $C_1, \ldots, C_L$
- features $\phi_i(c_{i-1}, y_i)$

Define the model

$$p_\theta(y_1:L, c_1:L-1) \propto \exp \left( \sum_{i=1}^L \theta^\top \phi_i(c_{i-1}, y_i) \right) \cdot \kappa(y, c)$$

Graphical model structure:

inference via forward-backward!
Adaptive Context Selection

- Select context sets $C_i$ during forward pass of inference
Adaptive Context Selection

- Select context sets $C_i$ during forward pass of inference
- Greedily select contexts with largest mass
Adaptive Context Selection

- Select context sets $C_i$ during forward pass of inference
- Greedily select contexts with largest mass
Adaptive Context Selection

- Select context sets $C_i$ during forward pass of inference
- Greedily select contexts with largest mass
Adaptive Context Selection

- Select context sets $C_i$ during forward pass of inference
- Greedily select contexts with largest mass

\[ C_1 \]
Adaptive Context Selection

- Select context sets $C_i$ during forward pass of inference
- Greedily select contexts with largest mass

```
C

a

b
c

d
e

... ⋮

C_1
```
Adaptive Context Selection

- Select context sets $C_i$ during forward pass of inference
- Greedily select contexts with largest mass

$C_1$
Adaptive Context Selection

- Select context sets $C_i$ during forward pass of inference
- Greedily select contexts with largest mass
Adaptive Context Selection

- Select context sets $C_i$ during forward pass of inference
- Greedily select contexts with largest mass

The diagram shows a network of contexts $C_1$ and $C_2$. The selection process involves greedily choosing contexts with the largest mass, indicated by arrows pointing to contexts with stars. Biases towards short contexts unless there is high confidence are noted.
Adaptive Context Selection

- Select context sets $C_i$ during forward pass of inference
- Greedily select contexts with largest mass

Biases towards short contexts unless there is high confidence.
Precision

input $x$:  $\text{V o l c a n i c}$

output $y$:  $\text{v o l c a n i c}$
**Precision**

input $x$: \[ \text{v o l c a n i c} \]

output $y$: \[ \text{v o l c a n i c} \]

Model assigns probability to each prediction, so can predict on most confident subset.
Precision

input $x$:  V  O  L  C  A  N  I  C
output $y$:  v  o  l  c  a  n  i  c

Model assigns probability to each prediction, so can predict on most confident subset.

Measure precision (# of correct words) vs. recall (# of words predicted).
Precision

Model assigns probability to each prediction, so can predict on most confident subset.

Measure precision (# of correct words) vs. recall (# of words predicted).

- comparison: beam search
Precision

Measure precision (# of correct words) vs. recall (# of words predicted).

![Graph showing precision vs. recall for Word Recognition with two lines: one for Beam search (blue) and one for RCM (red).]
Partially Supervised Learning

Decipherment task:

\[
\text{cipher} \quad \text{am} \mapsto 5, \quad \text{l} \mapsto 13, \quad \text{what} \mapsto 54, \ldots
\]
Partially Supervised Learning

Decipherment task:

- cipher \( \mapsto \) 5, I \( \mapsto \) 13, what \( \mapsto \) 54, ...
- latent \( z \)  I  am  what  I  am

Output \( y \)

Goal: determine cipher

Fit 2nd-order HMM with EM, using RCMs for approximate E-step.

Use learned emissions to determine cipher.

Again compare to beam search (Nuhn et al., 2013)

Fraction of correctly mapped words:

<table>
<thead>
<tr>
<th>Training passes</th>
<th>Mapping accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>0.1</td>
</tr>
<tr>
<td>10</td>
<td>0.2</td>
</tr>
<tr>
<td>15</td>
<td>0.3</td>
</tr>
<tr>
<td>20</td>
<td>0.4</td>
</tr>
</tbody>
</table>

J. Steinhardt (Stanford)

Learning and Inference

September 8, 2015
Decipherment task:

cipher   am $\mapsto$ 5, l $\mapsto$ 13, what $\mapsto$ 54, ...
latent z   l   am   what   l   am
output y   13  5  54  13  5
Partially Supervised Learning

Decipherment task:

\[
\begin{align*}
\text{cipher} & \quad \text{am} \mapsto 5, \quad \text{l} \mapsto 13, \quad \text{what} \mapsto 54, \ldots \\
\text{latent } z & \quad \text{l} \quad \text{am} \quad \text{what} \quad \text{l} \quad \text{am} \\
\text{output } y & \quad 13 \quad 5 \quad 54 \quad 13 \quad 5
\end{align*}
\]

Goal: determine cipher
Partially Supervised Learning

Decipherment task:

\[
\begin{align*}
\text{cipher} & \quad \text{am} \rightarrow 5, \quad \text{l} \rightarrow 13, \quad \text{what} \rightarrow 54, \ldots \\
\text{latent } z & \quad \text{l} \quad \text{am} \quad \text{what} \quad \text{l} \quad \text{am} \\
\text{output } y & \quad 13 \quad 5 \quad 54 \quad 13 \quad 5
\end{align*}
\]

Goal: determine cipher

Fit 2nd-order HMM with EM, using RCMs for approximate E-step.
Decipherment task:

\[
\begin{align*}
\text{cipher} & \rightarrow \text{am} \mapsto 5, \text{l} \mapsto 13, \text{what} \mapsto 54, \ldots \\
\text{latent } z & \quad \text{l} \quad \text{am} \quad \text{what} \quad \text{l} \quad \text{am} \\
\text{output } y & \quad 13 \quad 5 \quad 54 \quad 13 \quad 5
\end{align*}
\]

Goal: determine cipher

Fit 2nd-order HMM with EM, using RCMs for approximate E-step.

- use learned emissions to determine cipher.
Partially Supervised Learning

Decipherment task:

\[
\begin{align*}
\text{cipher} & \quad \text{am} \leftrightarrow 5, \ l \leftrightarrow 13, \ \text{what} \leftrightarrow 54, \ldots \\
\text{latent } z & \quad l \quad \text{am} \quad \text{what} \quad l \quad \text{am} \\
\text{output } y & \quad 13 \quad 5 \quad 54 \quad 13 \quad 5
\end{align*}
\]

Goal: determine cipher

Fit 2nd-order HMM with EM, using RCMs for approximate E-step.

- use learned emissions to determine cipher.
- again compare to beam search (Nuhn et al., 2013)
Partially Supervised Learning

Fraction of correctly mapped words:

![Graph showing mapping accuracy over training passes for RCM and beam methods.](image-url)
Contexts During Training

Context lengths increase smoothly during training:
Context lengths increase smoothly during training:

Contexts During Training

Start of training: little information, short contexts.
End of training: lots of information, long contexts.
Contexts During Training

Context lengths increase smoothly during training:

Start of training: little information, short contexts.

End of training: lots of information, long contexts.
Contexts During Training

Context lengths increase smoothly during training:

Start of training: little information, short contexts.
End of training: lots of information, long contexts.
Discussion

RCMs provide both expressivity and coverage, which enable:
RCMs provide both expressivity and coverage, which enable:

- More accurate uncertainty estimates (precision)
Discussion

RCMs provide both expressivity and coverage, which enable:

- More accurate uncertainty estimates (precision)
- Better partially supervised learning updates
Discussion

RCMs provide both expressivity and coverage, which enable:

- More accurate uncertainty estimates (precision)
- Better partially supervised learning updates

Reproducible experiments on Codalab: codalab.org/worksheets
1. Motivation

2. Formal Setting

3. Reified Context Models

4. Relaxed Supervision

5. Open Questions
Intractable Supervision

Sometimes, even supervision is intractable:

\[
\text{input } x: \quad \text{What is the largest city in California?}
\]
\[
\text{latent } z: \quad \arg\max(\lambda x.\text{CITY}(x) \land \text{LOC}(x, \text{CA}), \lambda x.\text{POPULATION}(x))
\]
\[
\text{output } y: \quad \text{Los Angeles}
\]
Intractable Supervision

Sometimes, even supervision is intractable:

input $x$: What is the largest city in California?

latent $z$: \[ \text{argmax}(\lambda x.\text{CITY}(x) \land \text{LOC}(x, \text{CA}), \lambda x.\text{POPULATION}(x)) \]

output $y$: Los Angeles

Intractable no matter how simple the model is!
Intractable Supervision

Sometimes, even supervision is intractable:

- **input** $x$: What is the largest city in California?
- **latent** $z$: $\arg\max (\lambda x.\text{CITY}(x) \land \text{LOC}(x, \text{CA}), \lambda x.\text{POPULATION}(x))$
- **output** $y$: Los Angeles

Intractable no matter how simple the model is!

- but likely statistical relationships (e.g. between \text{CITY} and \text{Los Angeles})
Sometimes, even supervision is intractable:

**input** $x$: What is the largest city in California?

**latent** $z$: $\text{argmax}(\lambda x. \text{CITY}(x) \land \text{LOC}(x, \text{CA}), \lambda x. \text{POPULATION}(x))$

**output** $y$: Los Angeles

Intractable no matter how simple the model is!

- but likely statistical relationships (e.g. between \text{CITY} and \text{Los Angeles})

Need a way to relax the **likelihood**.
Sometimes, even supervision is intractable:

- **input** $x$: What is the largest city in California?
- **latent** $z$: $\text{argmax}(\lambda x.\text{CITY}(x) \land \text{LOC}(x, \text{CA}), \lambda x.\text{POPULATION}(x))$
- **output** $y$: Los Angeles

Intractable no matter how simple the model is!

- but likely statistical relationships (e.g. between \text{CITY} and \text{Los Angeles})

Need a way to relax the **likelihood**.

- while maintaining good statistical properties (asymptotic consistency)
Approach

Start with intractable likelihood $q(y \mid z)$, model family $p_\theta(z \mid x)$. 

$\theta$
Approach

Start with intractable likelihood \( q(y \mid z) \), model family \( p_\theta(z \mid x) \).

Replace \( q(y \mid z) \) with family of likelihoods \( q_\beta(y \mid z) \) (some very easy).
Approach

Start with intractable likelihood $q(y \mid z)$, model family $p_\theta(z \mid x)$.

Replace $q(y \mid z)$ with family of likelihoods $q_\beta(y \mid z)$ (some very easy).

Derive constraints on $(\theta, \beta)$ that ensure tractability.
Approach

Start with intractable likelihood $q(y \, | \, z)$, model family $p_\theta(z \, | \, x)$.

Replace $q(y \, | \, z)$ with family of likelihoods $q_\beta(y \, | \, z)$ (some very easy).

Derive constraints on $(\theta, \beta)$ that ensure tractability.

Learn within the tractable region.
Relaxed Supervision: Example

- **input** $x$: Company officials refused to comment.
- **latent** $z$: 公司官员拒绝对此发表评论。
- **output** $y$: 公司官员拒绝对此发表评论。
Relaxed Supervision: Example

input $x$: Company officials refused to comment.

latent $z$: 

output $y$: 公司 官员 拒绝 对此 发表评论。

Idea: instead of requiring $y$ to match observed output, penalize based on some weighted distance $\text{dist}_\beta(\tilde{y}, y)$. 
Relaxed Supervision: Example

input $x$: Company officials refused to comment.
latent $z$: 
output $y$: 公司官员拒绝对此发表评论。

Idea: instead of requiring $y$ to match observed output, penalize based on some weighted distance $\text{dist}_\beta(\tilde{y}, y)$.

$$
\ell(\theta, \beta; x, y) = -\log \left( \sum_z p_\theta(z, y | x) \right)
$$

As $\beta \to \infty$, recover original objective. But optimizing will send $\beta \to 0$!

Two questions: How to create natural pressure to increase $\beta$? How to define distances for general problems?
Relaxed Supervision: Example

input $x$: Company officials refused to comment.

latent $z$: 

output $y$: 公司官员拒绝对此发表评论。

Idea: instead of requiring $y$ to match observed output, penalize based on some weighted distance $\text{dist}_\beta(\tilde{y}, y)$.

$$
\ell(\theta, \beta; x, y) = -\log \left( \sum_{z, \tilde{y}} p_{\theta}(z, \tilde{y} \mid x) \exp(- \text{dist}_\beta(\tilde{y}, y)) \right)
$$

As $\beta \to \infty$, recover original objective. But optimizing will send $\beta \to 0$!
Relaxed Supervision: Example

input $x$: Company officials refused to comment.

latent $z$: 

output $y$: 公司官员拒绝对此发表评论。

Idea: instead of requiring $y$ to match observed output, penalize based on some weighted distance $\text{dist}_\beta(\tilde{y}, y)$.

$$\ell(\theta, \beta; x, y) = -\log \left( \sum_{z, \tilde{y}} p_\theta(z, \tilde{y} \mid x) \exp(-\text{dist}_\beta(\tilde{y}, y)) \right)$$

As $\beta \to \infty$, recover original objective.
Relaxed Supervision: Example

input $x$: Company officials refused to comment.

latent $z$: 公司官员拒绝对此发表评论。

output $y$: 公司官员拒绝对此发表评论。

Idea: instead of requiring $y$ to match observed output, penalize based on some weighted distance $\text{dist}_\beta(\tilde{y}, y)$.

$$
\ell(\theta, \beta; x, y) = -\log \left( \sum_{z, \tilde{y}} p_{\theta}(z, \tilde{y} | x) \exp(-\text{dist}_\beta(\tilde{y}, y)) \right)
$$

As $\beta \to \infty$, recover original objective.

- but optimizing will send $\beta \to 0$!
Relaxed Supervision: Example

input $x$: Company officials refused to comment.
latent $z$: 
output $y$: 公司 官员 拒绝 对此 发表评论。

Idea: instead of requiring $y$ to match observed output, penalize based on some weighted distance $\text{dist}_\beta(\tilde{y}, y)$.

$$
\ell(\theta, \beta; x, y) = -\log \left( \sum_{z, \tilde{y}} p_\theta(z, \tilde{y} \mid x) \exp(- \text{dist}_\beta(\tilde{y}, y)) \right)
$$

As $\beta \to \infty$, recover original objective.

- but optimizing will send $\beta \to 0$!

Two questions:
- How to create natural pressure to increase $\beta$?
Relaxed Supervision: Example

input $x$: Company officials refused to comment.

latent $z$: 公司官员拒绝对此发表评论。

output $y$: 公司官员拒绝对此发表评论。

Idea: instead of requiring $y$ to match observed output, penalize based on some weighted distance $\text{dist}_\beta(\tilde{y}, y)$.

$$\ell(\theta, \beta; x, y) = -\log \left( \sum_{z, \tilde{y}} p_{\theta}(z, \tilde{y} | x) \exp(-\text{dist}_\beta(\tilde{y}, y)) \right)$$

As $\beta \to \infty$, recover original objective.

- but optimizing will send $\beta \to 0$!

Two questions:

- How to create natural pressure to increase $\beta$?
- How to define distances for general problems?
Relaxed Supervision: Formal Framework

- Assume (WLOG) that $z \rightarrow y$ is deterministic: $y = f(z)$. 
Assume (WLOG) that $z \rightarrow y$ is deterministic: $y = f(z)$.
Let $S(z, y) \in \{0, 1\}$ encode the constraint $[f(z) = y]$. 

Relaxed Supervision: Formal Framework

- Assume (WLOG) that \( z \rightarrow y \) is deterministic: \( y = f(z) \).
- Let \( \mathcal{S}(z, y) \in \{0, 1\} \) encode the constraint \([f(z) = y]\).
- Take projections \( \pi_j : \mathcal{Y} \rightarrow \mathcal{Y}_j, j = 1, \ldots, k \).
Relaxed Supervision: Formal Framework

- Assume (WLOG) that $z \rightarrow y$ is deterministic: $y = f(z)$.
- Let $S(z, y) \in \{0, 1\}$ encode the constraint $[f(z) = y]$.
- Take projections $\pi_j : \mathcal{Y} \rightarrow \mathcal{Y}_j$, $j = 1, \ldots, k$.
- Let $S_j(z, y) = [\pi_j(f(z)) = \pi_j(y)]$ be the projected constraint.
Relaxed Supervision: Formal Framework

- Assume (WLOG) that $z \rightarrow y$ is deterministic: $y = f(z)$.
- Let $S(z, y) \in \{0, 1\}$ encode the constraint $[f(z) = y]$.
- Take projections $\pi_j : \mathcal{Y} \rightarrow \mathcal{Y}_j$, $j = 1, \ldots, k$.
- Let $S_j(z, y) = [\pi_j(f(z)) = \pi_j(y)]$ be the projected constraint.
- Define distance function:

\[
\text{dist}_\beta(z, y) = \sum_{j=1}^k \beta_j \cdot (1 - S_j(z, y)).
\]
Relaxed Supervision: Formal Framework

- Assume (WLOG) that $z \rightarrow y$ is deterministic: $y = f(z)$.
- Let $S(z, y) \in \{0, 1\}$ encode the constraint $[f(z) = y]$.
- Take projections $\pi_j : \mathcal{Y} \rightarrow \mathcal{Y}_j$, $j = 1, \ldots, k$.
- Let $S_j(z, y) = [\pi_j(f(z)) = \pi_j(y)]$ be the projected constraint.
- Define distance function:

\[
\text{dist}_\beta(z, y) = \sum_{j=1}^k \beta_j \cdot (1 - S_j(z, y)).
\]

Note: can featurize $\text{dist}_\beta$ as $-\beta^\top \psi(z, y)$, where $\psi_j = S_j - 1$. 
Relaxed Supervision: Formal Framework

- Assume (WLOG) that $z \rightarrow y$ is deterministic: $y = f(z)$.
- Let $\mathbb{S}(z, y) \in \{0, 1\}$ encode the constraint $[f(z) = y]$.
- Take projections $\pi_j : \mathcal{Y} \rightarrow \mathcal{Y}_j, j = 1, \ldots, k$.
- Let $\mathbb{S}_j(z, y) = [\pi_j(f(z)) = \pi_j(y)]$ be the projected constraint.
- Define distance function:

$$
\text{dist}_\beta(z, y) = \sum_{j=1}^{k} \beta_j \cdot (1 - \mathbb{S}_j(z, y)) .
$$

Note: can featurize $\text{dist}_\beta$ as $-\beta^\top \psi(z, y)$, where $\psi_j = \mathbb{S}_j - 1$.

Lemma

Suppose that $\pi_1 \times \cdots \times \pi_k$ is injective. Then

$$
\mathbb{S}(z, y) = \bigwedge_{j=1}^{k} \mathbb{S}_j(z, y)
$$
Example: Unordered Supervision

input $x$:  a  b  a  a  
latent $z$:  d  c  d  d  
output $y$:  \{c : 1, d : 3\}

Let $\text{count}(\cdot, j)$ count number of occurrences of character $j$.

Decomposition:

\[
\begin{align*}
y &= f(z) \\
S(z, y) &= \Rightarrow V \bigwedge_j 1 [\text{count}(z, j) = \text{count}(y, j)]
\end{align*}
\]
Example: Unordered Supervision

input $x$: a b a a
latent $z$: d c d d
output $y$: \{c : 1, d : 3\}

Let $\text{count}(\cdot, j)$ count number of occurrences of character $j$. 
Example: Unordered Supervision

input $x$: a b a a
latent $z$: d c d d
output $y$: $\{c : 1, d : 3\}$

Let $\text{count}(\cdot, j)$ count number of occurrences of character $j$.

Decomposition:

$$f(z) \quad \left\{ \begin{array}{c}
[y = \text{multiset}(z)] \\
S(z, y)
\end{array} \right.$$
Example: Unordered Supervision

Input $x$: a b a a
Latent $z$: d c d d
Output $y$: $\{c : 1, d : 3\}$

Let $\text{count}(\cdot, j)$ count number of occurrences of character $j$.

Decomposition:

$$\mathcal{S}(z, y) \quad \implies \quad \bigwedge_{j=1}^{v} \left[\text{count}(z, j) = \text{count}(y, j)\right]$$
Example: Unordered Supervision

input $x$: a b a a
latent $z$: d c d d
output $y$: $\{c : 1, d : 3\}$

Let $\text{count}(\cdot, j)$ count number of occurrences of character $j$.

Decomposition:

$$f(z) \leftarrow y = \text{multiset}(z) \quad \Rightarrow \quad \bigwedge_{j=1}^{\pi_j(y)} \{\text{count}(z, j) = \text{count}(y, j)\}$$
Relaxed Supervision

**Example: Unordered Supervision**

<table>
<thead>
<tr>
<th>input $x$:</th>
<th>a b a a</th>
</tr>
</thead>
<tbody>
<tr>
<td>latent $z$:</td>
<td>d c d d</td>
</tr>
<tr>
<td>output $y$:</td>
<td>${c: 1, d: 3}$</td>
</tr>
</tbody>
</table>

Let $\text{count}(\cdot, j)$ count number of occurrences of character $j$.

Decomposition:

\[
\underbrace{f(z)}_{S(z,y)} \iff \bigwedge_{j=1}^{\pi_j(y)} \left[ \underbrace{\text{count}(z, j) = \text{count}(y, j)}_{S_j(z,y)} \right]
\]
Example: Conjunctive Semantic Parsing

Side information: *predicates* \( \{ Q_1, \ldots, Q_m \} \).

- e.g. \( Q_6 = [\text{DOG}] = \text{set of all dogs} \)
Example: Conjunctive Semantic Parsing

Side information: *predicates* \{Q_1, \ldots, Q_m\}.

- e.g. \(Q_6 = [\text{DOG}] = \text{set of all dogs}\)

input \(x\): brown dog (input utterance)
Example: Conjunctive Semantic Parsing

Side information: *predicates* \( \{ Q_1, \ldots, Q_m \} \).

- e.g. \( Q_6 = [\text{DOG}] = \text{set of all dogs} \)

input \( x \): 
brown dog 
(input utterance)

latent \( z \): 
\((Q_{11}, Q_6)\) 
(set of all brown objects, set of all dogs)
Example: Conjunctive Semantic Parsing

Side information: *predicates* $\{Q_1, \ldots, Q_m\}$.
- e.g. $Q_6 = [\text{DOG}] = \text{set of all dogs}$

- **input $x$:** brown dog (input utterance)
- **latent $z$:** $(Q_{11}, Q_6)$ (set of all brown objects, set of all dogs)
- **output $y$:** $Q_{11} \cap Q_6$ (denotation, observed as a set)
Example: Conjunctive Semantic Parsing

Side information: *predicates* \{Q_1, \ldots, Q_m\}.

- e.g. \( Q_6 = [\text{DOG}] = \text{set of all dogs} \)

  \[
  \begin{align*}
  \text{input } x: & \quad \text{brown dog} & \text{(input utterance)} \\
  \text{latent } z: & \quad (Q_{11}, Q_6) & \text{(set of all brown objects, set of all dogs)} \\
  \text{output } y: & \quad Q_{11} \cap Q_6 & \text{(denotation, observed as a set)}
  \end{align*}
  \]

For \( z = (Q_{j_1}, \ldots, Q_{j_L}) \), define the denotation \([z] = Q_{j_1} \cap \cdots \cap Q_{j_L}\).
Relaxed Supervision

Example: Conjunctive Semantic Parsing

Side information: \textit{predicates} \{Q_1, \ldots, Q_m\}.

\begin{itemize}
  \item e.g. \(Q_6 = \text{[DOG]} = \text{set of all dogs}\)
\end{itemize}

\begin{itemize}
  \item input \(x\): brown dog \hspace{1cm} \text{(input utterance)}
  \item latent \(z\): \((Q_{11}, Q_6)\) \hspace{1cm} \text{(set of all brown objects, set of all dogs)}
  \item output \(y\): \(Q_{11} \cap Q_6\) \hspace{1cm} \text{(denotation, observed as a set)}
\end{itemize}

For \(z = (Q_{j_1}, \ldots, Q_{j_L})\), define the denotation \(\llbracket z \rrbracket = Q_{j_1} \cap \cdots \cap Q_{j_L}\).

Decomposition:

\[
y = \underbrace{\llbracket z \rrbracket}_{\mathcal{S}(z,y)}
\]
Example: Conjunctive Semantic Parsing

Side information: *predicates* \{ \(Q_1, \ldots, Q_m\)\}.
- e.g. \(Q_6 = [\text{DOG}] = \text{set of all dogs}\)

input \(x\): brown dog (input utterance)

latent \(z\): \((Q_{11}, Q_6)\) (set of all brown objects, set of all dogs)

output \(y\): \(Q_{11} \cap Q_6\) (denotation, observed as a set)

For \(z = (Q_{j_1}, \ldots, Q_{j_L})\), define the denotation \([z] = Q_{j_1} \cap \cdots \cap Q_{j_L}\).

Decomposition:

\[
y = [z] \iff \bigwedge_{j=1}^{\infty} \mathbb{I}[[z] \subseteq Q_j] = \mathbb{I}[y \subseteq Q_j]
\]
Normalization Constant

Create pressure to increase $\beta$ by adding normalization constant:

$$q_\beta(y \mid z) = \exp\left(\beta^\top \psi(z, y) - A(\beta)\right) - \text{dist}_\beta(z, y)$$
Normalized Constant

Create pressure to increase $\beta$ by adding normalization constant:

$$q_\beta(y \mid z) = \exp(\beta^\top \psi(z, y) - A(\beta)) - \text{dist}_\beta(z, y)$$

$$\ell(\theta, \beta; x, y) = -\log \left( \sum_z p_\theta(z \mid x) q_\beta(y \mid z) \right).$$
Normalization Constant

Create pressure to increase $\beta$ by adding normalization constant:

$$q_\beta(y \mid z) = \exp\left(\beta^\top \psi(z, y) - A(\beta) - \text{dist}_\beta(z, y)\right)$$

$$\ell(\theta, \beta; x, y) = -\log\left(\sum_z p_\theta(z \mid x) q_\beta(y \mid z)\right).$$

Lemma

Given $\pi_1, \ldots, \pi_k$, let $A(\beta) \overset{\text{def}}{=} \sum_{j=1}^k \log \left(1 + (|Y_j| - 1) \exp(-\beta_j)\right)$. Then, $\sum_y \exp(-\text{dist}_\beta(z, y)) \leq A(\beta)$ for all $z$. 
Normalization Constant

Create pressure to increase $\beta$ by adding normalization constant:

$$q_{\beta}(y \mid z) = \exp(\beta^\top \psi(z, y) - A(\beta))$$

$$- \text{dist}_\beta(z, y)$$

$$\ell(\theta, \beta; x, y) = -\log \left( \sum_z p_\theta(z \mid x) q_{\beta}(y \mid z) \right).$$

**Lemma**

Given $\pi_1, \ldots, \pi_k$, let $A(\beta) \overset{\text{def}}{=} \sum_{j=1}^k \log (1 + (|Y_j| - 1) \exp(-\beta_j))$. Then,

$$\sum_y \exp(-\text{dist}_\beta(z, y)) \leq A(\beta) \text{ for all } z.$$  

**Lemma**

Jointly minimizing $L(\theta, \beta) = \mathbb{E}[\ell(\theta, \beta; x, y)]$ yields a consistent estimate of the true parameters $\theta^*$.  

Constraints for Efficient Inference

Inference task:

$$
\nabla_\theta \log \rho_\theta (y \mid x) = \mathbb{E}_{\hat{z} \sim \rho_\theta (\cdot \mid x, y)} [\phi (x, \hat{z}, y)] - \mathbb{E}_{\hat{z}, \hat{y} \sim \rho_\theta (\cdot \mid x)} [\phi (x, \hat{z}, \hat{y})].
$$

sample $z$ given $x, y$

sample $z$ given $x$
Constraints for Efficient Inference

Inference task:

\[
\nabla_\theta \log p_\theta(y \mid x) = \underbrace{\mathbb{E}_{\hat{z} \sim p_\theta(\cdot \mid x, y)}[\phi(x, \hat{z}, y)]} - \underbrace{\mathbb{E}_{\hat{z}, \hat{y} \sim p_\theta(\cdot \mid x)}[\phi(x, \hat{z}, \hat{y})]}.
\]

sample \( z \) given \( x, y \)

sample \( z \) given \( x \)

\[
p_{\theta, \beta}(z \mid x, y) \propto p_\theta(z \mid x)q_\beta(y \mid z) \propto p_\theta(z \mid x) \exp(\beta^\top \psi(z, y)).
\]
Constraints for Efficient Inference

Inference task:

\[ \nabla_{\theta} \log p_{\theta}(y \mid x) = \mathbb{E}_{z \sim p_{\theta}(\cdot \mid x, y)}[\phi(x, \hat{z}, y)] - \mathbb{E}_{\hat{z}, \hat{y} \sim p_{\theta}(\cdot \mid x)}[\phi(x, \hat{z}, \hat{y})]. \]

\( \text{sample } z \text{ given } x, y \)
\( \text{sample } z \text{ given } x \)

\[ p_{\theta, \beta}(z \mid x, y) \propto p_{\theta}(z \mid x)q_{\beta}(y \mid z) \]
\[ \propto p_{\theta}(z \mid x)\exp(\beta^{\top}\psi(z, y)). \]

Rejection sampler:

- sample from \( p_{\theta}(z \mid x) \)
- accept with probability \( \exp(\beta^{\top}\psi(z, y)) \).
Constraints for Efficient Inference

Inference task:

\[
\nabla_\theta \log p_\theta(y | x) = \underbrace{\mathbb{E}_{\hat{z} \sim p_\theta(\cdot | x, y)}[\phi(x, \hat{z}, y)]}_\text{sample } z \text{ given } x, y - \underbrace{\mathbb{E}_{\hat{z}, \hat{y} \sim p_\theta(\cdot | x)}[\phi(x, \hat{z}, \hat{y})]}_\text{sample } z \text{ given } x
\]

\[
p_{\theta, \beta}(z | x, y) \propto p_\theta(z | x) q_\beta(y | z)
\]

\[
\propto p_\theta(z | x) \exp(\beta^\top \psi(z, y)).
\]

Rejection sampler:

- sample from \( p_\theta(z | x) \)
- accept with probability \( \exp(\beta^\top \psi(z, y)) \).

Bound expected number of samples:

\[
\sum_{x,y \in \text{Data}} \left( \sum_z p_\theta(z | x) \exp(\beta^\top \psi(z, y)) \right)^{-1} \leq \tau. \quad (1)
\]
Constraints for Efficient Inference

Inference task:

\[ \nabla_{\theta} \log p_\theta(y \mid x) = \mathbb{E}_{\hat{z} \sim p_\theta(\cdot \mid x, y)}[\phi(x, \hat{z}, y)] - \mathbb{E}_{\hat{z}, \hat{y} \sim p_\theta(\cdot \mid x)}[\phi(x, \hat{z}, \hat{y})]. \]

- sample \( z \) given \( x, y \)
- sample \( z \) given \( x \)

\[ p_{\theta, \beta}(z \mid x, y) \propto p_\theta(z \mid x)q_\beta(y \mid z) \]
\[ \propto p_\theta(z \mid x)\exp(\beta^\top \psi(z, y)). \]

Rejection sampler:

- sample from \( p_\theta(z \mid x) \)
- accept with probability \( \exp(\beta^\top \psi(z, y)) \).

Bound expected number of samples:

\[ \sum_{x, y \in \text{Data}} \left( \sum_z p_\theta(z \mid x)\exp(\beta^\top \psi(z, y)) \right)^{-1} \leq \tau. \quad (1) \]

Ratio of normalization constants: can optimize subject to (1) (similar to CCCP).
Experiments

Conjunctive semantic parsing:

![Diagram showing accuracy and number of samples over iterations for different FixedBeta values.]
Experiments

Conjunctive semantic parsing:

Accuracy and number of samples over iteration for different Beta values.
1. Motivation
2. Formal Setting
3. Reified Context Models
4. Relaxed Supervision
5. Open Questions
Scale up to larger tasks
  - semantic parsing, reinforcement learning, program induction
- Scale up to larger tasks
  - semantic parsing, reinforcement learning, program induction
- Extend to Bayesian models
- Scale up to larger tasks
  - semantic parsing, reinforcement learning, program induction
- Extend to Bayesian models
- Understand non-convex optimization
Open Questions

- Scale up to larger tasks
  - semantic parsing, reinforcement learning, program induction
- Extend to Bayesian models
- Understand non-convex optimization
- Metacomputation
  - using Reified Context Models?
- Scale up to larger tasks
  - semantic parsing, reinforcement learning, program induction
- Extend to Bayesian models
- Understand non-convex optimization
- Metacomputation
  - using Reified Context Models?
- Probabilistic abstract interpretation
Open Questions

- Scale up to larger tasks
  - semantic parsing, reinforcement learning, program induction
- Extend to Bayesian models
- Understand non-convex optimization
- Metacomputation
  - using Reified Context Models?
- Probabilistic abstract interpretation
- Statistics & Computation: still a long way to go

Thanks!
- Scale up to larger tasks
  - semantic parsing, reinforcement learning, program induction
- Extend to Bayesian models
- Understand non-convex optimization
- Metacomputation
  - using Reified Context Models?
- Probabilistic abstract interpretation
- Statistics & Computation: still a long way to go

Thanks!
谢谢