A Fast Variational Approach for Learning Markov Random Field Language Models

Yacine Jernite, Alexander Rush and David Sontag
Let’s talk about language modeling

● What is language modeling?
  ● Probability distribution over sentences
  ● Text generation, machine translation, speech recognition
  ● Useful parameters
Let’s talk about language modeling

- Language modeling: size $K$ context

- Language model:
  \[ p(\vec{w}) = \prod_{i=1}^{n} p(w_i|w_{i-1}, \ldots, w_{i-K}) \]

- Multinomial $p(w_i|w_{i-1}, \ldots, w_{i-K})$: n-gram models

Let's talk about language modeling

The rats scared the cat.
Let’s talk about language modeling

- **Neural language models**
  - Based on embeddings of the vocabulary $V$ into $\mathbb{R}^D$
  - Expects distributional similarities, same idea as Latent Semantic Analysis
Let’s talk about language modeling

- Neural language models
  - Distance dependent context embedding:
    \[ U^{2}_{\text{scared}}, U^{1}_{\text{the}} \in \mathbb{R}^{D} \]
  - Target word embedding:
    \[ W_{\text{cat}} \in \mathbb{R}^{D} \]
  - Distribution:
    \[
p(\text{cat}| \text{scared, the}) = \frac{\exp((U^{2}_{\text{scared}} + U^{1}_{\text{the}})W_{\text{cats}}^{T})}{Z(\text{scared, the})}
\]
Bi-directional embedding systems

- How to get better embeddings

Word2Vec: take advantage of right context

\[
p(w_i | w_{ci}) = \frac{\exp \left( \left( \sum_{l=-K}^{K} U^l_{wi+l} + \sum_{l=1}^{K} U^l_{wi+l} \right) W^T_{wi} \right)}{Z(w_{ci})}
\]

- Does not define a distribution over sentences, maximizes

\[
\sum_{i=1}^{n} \log p(w_i | w_{ci})
\]
Our contributions

In this work:

• We propose a new language model, with similarities to recent embedding learning algorithms

• We provide a fast learning algorithm, independent of the corpus size
Markov random fields

- Graph structure $G = (V, E)$, cliques $c \in G$:

$$p(X_1, \ldots, X_n) = \exp(\sum_{c \in G} \theta_c(x_c) - A(\theta))$$
Markov random fields

- Graph structure $G = (V, E)$, cliques $c \in G$:
  
  $$p(X_1, \ldots, X_n) = \exp(\sum_{c \in G} \theta_c(x_c) - A(\theta))$$

- Log-partition function:
  
  $$A(\theta) = \log \left( \sum_x \exp(\sum_{c \in G} \theta_c(x_c)) \right)$$
Low rank Markov sequence model

- Word distribution depends on size $K$ context:

\[
\theta_{i,j}(w_i, w_j) = U_{w_i}^{(j-i)} W_{w_j}^T
\]

- Low rank log-potentials:

\[
p(w_i \mid w_{C_i}) = \frac{\exp((\Sigma_{l=-K}^{1} U_{w_{i+l}}' l + \Sigma_{l=1}^{K} U_{w_{i+l}}' l) W_{w_i}' T))}{Z(w_{C_i})}
\]

- Pseudo-likelihood equivalent to Word2Vec objective!

\[
\psi(\overrightarrow{w}) = \Sigma_{i=1}^{N} \log p(w_i \mid w^{-i}) = \Sigma_{i=1}^{N} \log p(w_i \mid w_{C_i})
\]
Running time to learn language models

• N-gram model: $O(N \times K)$

• Neural Language Models: $O(N \times |V|)$
  o Approximations such as hierarchical softmax can reduce this further

• MRF likelihood learning: $O(N \times |V|^{K+1})$
  o MRF is treewidth $K$
  o We need an approximation of $A(\theta)$
Obtaining a tractable approximation for $A(\theta)$

1. Lifted inference
   - Deriving a symmetrical model
   - Complexity: $O(|V|^K + 1)$
   - Independent of number of words N

2. Tree Re-Weighted approximation
   - Wainwright et al., 2005
   - Complexity: $O(K \times |V|^2)$
What is lifting?

- Symmetric graphs
What is lifting?

- Lifted loopy belief propagation:
- Lifted TRW (Bui et al., UAI ‘14)
1. Getting symmetries: cyclic model

- Some regularities

The rats scared the cat.

Then they stole its milk.
1. Getting symmetries: cyclic model

- Border effects: sentences

The rats scared the cat.
The rats stole its milk.
1. Getting symmetries: cyclic model

- Border effects: sentences
1. Getting symmetries: cyclic model

- Broken symmetry: conditioning
1. Getting symmetries: cyclic model

- Broken symmetry: conditioning

\[ p(<S>) p(w | <S>) = p(w) \]
2. Tree Re-Weighted approximation

- $A(\theta)$ is convex in $\theta$:

\[ A(\theta_G) \leq \frac{1}{2} A(\theta_{T_1}) + \frac{1}{2} A(\theta_{T_2}) \]
2. Tree Re-Weighted approximation

\[ \forall (i, j) \in E, \rho_{i,j} = \frac{1}{K + 1} \]
Deriving the bound

\[ A(\theta) = \max_{\mu \in \mathcal{M}} \langle \theta, \mu \rangle + H(\mu) \]
\[ \leq \max_{\mu \in \mathcal{M}} \langle \theta^{\text{cycl}}, \mu \rangle + H(\mu) \]
\[ \leq \max_{\mu \in \mathcal{L}} \sum_{i=1}^{N} \left( \langle \theta_{i}^{\text{cycl}}, \mu_{i} \rangle + H(\mu_{i}) \right) + \sum_{(i,j) \in E} \left( \langle \theta_{i,j}^{\text{cycl}}, \mu_{i,j} \rangle - \rho_{i,j} I(\mu_{i,j}) \right) \]
\[ = \frac{N}{K + 1} \max_{\mu \in \mathcal{L}} \sum_{i=0}^{K} \left( \langle \theta_{i}^{\text{cycl}}, \mu_{i} \rangle + H(\mu_{i}) \right) + \sum_{j=1}^{K} \left( \langle \theta_{0,j}^{\text{cycl}}, \mu_{0,j} \rangle - I(\mu_{0,j}) \right) \]
\[ \text{s.t. } \mu_{0} = \mu_{1} = \cdots = \mu_{K} \]

Symmetrical version

Lambda-Rubinstein-Winter (TRW)

Lifted TRW, \( \rho = \frac{1}{K+1} \)
Deriving the bound

\[
\max_{\mu \in \mathcal{L}} \sum_{i=0}^{K} \left( \left\langle \theta_i^{\text{cycl}}, \mu_i \right\rangle + H(\mu_i) \right) + \sum_{j=1}^{K} \left( \left\langle \theta_{0,j}^{\text{cycl}}, \mu_{0,j} \right\rangle - I(\mu_{0,j}) \right)
\]
\[
\text{s.t. } \mu_0 = \mu_1 = \cdots = \mu_K
\]
\[
= \min_{\delta} \max_{\mu \in \mathcal{L}} \sum_{i=0}^{K} \left( \left\langle \theta_i^{\text{cycl}} + \delta_i, \mu_i \right\rangle + H(\mu_i) \right) + \sum_{j=1}^{K} \left( \left\langle \theta_{0,j}^{\text{cycl}}, \mu_{0,j} \right\rangle - I(\mu_{0,j}) \right)
\]

\[
= \min_{\delta} A_K(\theta^\delta)
\]
Deriving the bound

\[ A(\theta) = \max_{\mu \in \mathcal{M}} \langle \theta, \mu \rangle + H(\mu) \]

\[ \leq \max_{\mu \in \mathcal{M}} \langle \theta^{\text{cycl}}, \mu \rangle + H(\mu) \]

\[ \leq \max_{\mu \in \mathcal{L}} \sum_{i=1}^{N} \left( \langle \theta_i^{\text{cycl}}, \mu_i \rangle + H(\mu_i) \right) + \sum_{(i,j) \in E} \rho_{i,j} \left( \langle \theta_{i,j}^{\text{cycl}}, \mu_{i,j} \rangle - I(\mu_{i,j}) \right) \]

\[ = \frac{N}{K + 1} \max_{\mu \in \mathcal{L}} \sum_{i=0}^{K} \left( \langle \theta_i^{\text{cycl}}, \mu_i \rangle + H(\mu_i) \right) + \sum_{j=1}^{K} \left( \langle \theta_{0,j}^{\text{cycl}}, \mu_{0,j} \rangle - I(\mu_{0,j}) \right) \]

\[ \text{s.t. } \mu_0 = \mu_1 = \cdots = \mu_K \]

\[ = \frac{N}{K + 1} \min_{\mu \in \mathcal{L}} \max_{\delta} \sum_{i=0}^{K} \left( \langle \theta_i^{\text{cycl}} + \delta_i, \mu_i \rangle + H(\mu_i) \right) + \sum_{j=1}^{K} \left( \langle \theta_{0,j}^{\text{cycl}}, \mu_{0,j} \rangle - I(\mu_{0,j}) \right) \]

\[ = \tilde{A}(\theta) \]

**Dual decomposition**
Algorithm

● Objective: \( \max_{\theta} \langle \theta, \mu \rangle - \tilde{A}(\theta) \)

● Collect moments \( \mu \)

*Whereas great* delays have been used by sheriffs, gaolers and other officers, to whose custody, any of the King's subjects have been committed for criminal or supposed criminal matters, in making returns of writs of habeas corpus…
Algorithm

- Objective: $\max_{\theta} \langle \theta, \mu \rangle - \tilde{A}(\theta)$

- Collect moments $\mu$

*Whereas* great *delays* have been used by sheriffs, gaolers and other officers, to whose custody, any of the King's subjects have been committed for criminal or supposed criminal matters, in making returns of writs of habeas corpus…
Algorithm

- **Objective:** \( \max_{\theta} \langle \theta, \mu \rangle - \tilde{A}(\theta) \)

- Collect moments \( \mu \)

Whereas great delays have been used by sheriffs, gaolers and other officers, to whose custody, any of the King's subjects have been committed for criminal or supposed criminal matters, in making returns of writs of habeas corpus…

Distance 1
Algorithm

- Objective: \( \max_{\theta} \langle \theta, \mu \rangle - \tilde{A}(\theta) \)

- Collect moments \( \mu \)

Whereas great **delays** have **been** used by sheriffs, gaolers and other officers, to whose custody, any of the King's subjects have been committed for criminal or supposed criminal matters, in making returns of writs of habeas corpus…
Algorithm

- Objective: $\tilde{\mathcal{L}}(\overline{w}; U, V) = \langle \theta, \mu \rangle - \tilde{A}(\theta)$

- Gradient descent
  - Compute $\tilde{A}(\theta)$
  - $\min_\delta$ ...
  - $\nabla_{\delta}$: belief propagation
  - until convergence, or $\tilde{A}_\delta(\theta)$
Algorithm

- Objective: \( \tilde{\mathcal{L}}(\overrightarrow{w}; U, V) = \langle \theta, \mu \rangle - \tilde{A}(\theta) \)

- Gradient descent
  - Compute \( \tilde{A}(\theta) = \min_{\delta} \)
  - Compute \( \nabla_{\theta} \tilde{A} \)
    - Belief propagation with \( \delta^* \)
Algorithm

- Objective: $\tilde{\mathcal{L}}(\overrightarrow{w}; U, V) = \langle \theta, \mu \rangle - \tilde{\mathcal{A}}(\theta)$

- Gradient descent
  - Compute $\tilde{\mathcal{A}}(\theta) = \min_\delta$
  - Compute $\nabla_\theta \tilde{\mathcal{A}}$
  - Compute $\nabla_\theta \tilde{\mathcal{L}}, \nabla_{U,V} \tilde{\mathcal{L}}$ (chain rule) $\theta^l = U^l W^T$

Complexity: $O(N \times K + K \times |V|^2)$
Comparison to exact inference

- Toy dataset for exact inference
  - \( V = \{a, b, c, d\} \)
  - \( N = 14 \)

\(<S>abdbbcaabdcdac<S>\)
Comparison to exact inference

Learn bound, test bound
Comparison to exact inference

Learn likelihood, test likelihood

Learn bound, test bound
Comparison to exact inference

![Graph showing objective values vs. LBFGS iterations. The graph compares exact log-likelihood and lower bound, with annotations indicating learning and testing likelihood and bound.]
Language modeling

- Penn Treebank dataset
  - $|V| = 10,000$
  - $N = 1,000,000$
Language modeling

![NLM test log-likelihood graph]

Objective values vs. LBFGS iterations
Language modeling

Lower bound on test log-likelihood
Language modeling

Lower bound on test log-likelihood

Learning log-potentials

Objective values

NLM test log-likelihood

Learning word embeddings

LBFGS iterations
Language modeling

<table>
<thead>
<tr>
<th>Word</th>
<th>MRF Lifted</th>
<th>Word2Vec</th>
</tr>
</thead>
<tbody>
<tr>
<td>elected</td>
<td>named appointed assistant</td>
<td>named appointed bank-holding</td>
</tr>
<tr>
<td>company</td>
<td>firm industry group</td>
<td>holding anacomp uniroyal</td>
</tr>
<tr>
<td>red</td>
<td>conservative freedom black</td>
<td>cross tape delicious</td>
</tr>
<tr>
<td>has</td>
<td>had is was</td>
<td>had been have</td>
</tr>
<tr>
<td>dollar</td>
<td>currency economy government</td>
<td>currency pound stabilized</td>
</tr>
<tr>
<td>dollars</td>
<td>francs lows highs</td>
<td>us$ millions billions</td>
</tr>
<tr>
<td>jack</td>
<td>richard david carl</td>
<td>kemp porter timothy</td>
</tr>
<tr>
<td>coffee</td>
<td>food network business</td>
<td>flat-rolled sport recycling</td>
</tr>
</tbody>
</table>
Part-of-Speech Tagging

Diagram showing the tagging process with 'PDT', 'DT', 'JJ', and 'NNS' labels for different parts of speech.
## Part-of-Speech Tagging

<table>
<thead>
<tr>
<th>Model</th>
<th>Total Acc</th>
<th>Unknown Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>95.8</td>
<td>65.4</td>
</tr>
<tr>
<td>Lifted MRF</td>
<td>96.0</td>
<td>76.0</td>
</tr>
</tbody>
</table>
Take away points

● New language model

The scared cat.
Take away points

● New language model

● Fast learning algorithm

\[ O(K \times |V|^2) \]
Take away points

● New language model

● Fast learning algorithm

● Wider applicability
Take away points

● Find the code at: https://github.com/srush/MRF-LM

● Questions?