Spectral Unsupervised Parsing with Additive Tree Metrics

Ankur Parikh, Shay Cohen, Eric P. Xing

Carnegie Mellon, University of Edinburgh
Overview

• **Model:** We present a novel approach to unsupervised parsing via latent tree structure learning

• **Algorithm:** Unlike existing methods, our algorithm is local-optima-free and has theoretical guarantees of statistical consistency

• **Key Ideas:**
  • Additive tree metrics from phylogenetics
  • Spectral decomposition of cross-covariance word embedding matrix
  • Kernel smoothing

• **Empirical:** Our method performs favorably to the constituent context model [Klein and Manning 2002]
Outline

• Motivation
• Intuition and Model
• Learning algorithm
• Experimental results
Supervised Parsing

**Training Set** – Given sentences with parse trees

**Test Set** – Find parse tree for each sentence

```
NN   ADV   VB   NN   CONJ   NN
Lions   quickly  chase   deer      and      antelope
```

```
S
  NP
    DT The
    NN bear
    VB likes
    NN fish

S
  NP
    VB Show
    NP
      DT the
      NN money

S
  NP
    VB the
    NP
      DT the
      NN cat
```

© Ankur Parikh 2014
Supervised Parsing

- **Modeling:** Assume tag sequence is generated by set of rules:

\[
P(tree) = P(S \rightarrow NP \ VP) \\
\times P(NP \rightarrow DT \ NN \mid NP) \\
\times P(VP \rightarrow VB \ NN \mid VP)
\]

- **Learning:** Easy to directly estimate rule probabilities from training data

- Foundation of modern supervised parsing systems.
Annotated Training Data Is Difficult to Obtain

- Annotating parse structure requires domain expertise, not easily crowdsourced.

- But sentences (and part-of-speech tags) are abundant!
Unsupervised Parsing

Training Set – Given sentences and part-of-speech tags

- DT  NN  VB  NN
  The  bear  likes  fish

- DT  NN  VB  DT  NN
  The  llama  eats  the  grass

Test Set – Find (unlabeled) parse tree for each sentence

- NN  ADV  VB  NN  CONJ  NN
  Lions  quickly  chase  deer  and  antelope

Parse tree structure now is a latent variable
Unsupervised Parsing is Much Harder

• Attempt to apply context free grammar strategy [Carroll and Charniak 1992, Pereira and Schabes 1992]

• **Modeling:** Some unknown set of rules generates the tree.

• **Learning:** Attempt to find set of rules $R$ and parameters $\theta$ that maximize data likelihood.
Unsupervised Parsing is Much Harder

- Unsupervised PCFGs perform **abysmally** and worse than trivial baselines such as right branching trees.

**Why?**

- **Modeling:** Solution that optimizes likelihood is not unique (**non-identifiability**) [Hsu et al. 2013]

- **Learning:** Likelihood function highly non-convex and search space contains **severe local optima**
Existing Approaches

• Other strategies outperform PCFGs but face similar challenges
  • objectives still NP-hard [Cohen & Smith 2012].
  • Severe local optima - accuracy can vary 40 percentage points between random restarts

• Need complicated techniques to achieve good results
  • Model/feature engineering [Klein & Manning 2002, Cohen & Smith 2009, Gillenwater et al. 2010]
  • Careful initialization [Klein & Manning 2002, Spitkovsky et al. 2010]
  • count transforms [Spitkovsky et al. 2013]

• These generally lack theoretical justification and effectiveness can vary across languages

© Ankur Parikh 2014
Existing Approaches

• Spectral techniques have led to theoretical insights for unsupervised parsing
  • Restriction of PCFG model [Hsu et al. 2013]
  • Weighted Matrix Completion [Bailly et al. 2013]

• But these algorithms not designed for good empirical performance

• Our goal is to give a first step to bridging this theory-experiment gap
Our Approach

• Formulate new model where unsupervised parsing corresponds to latent tree structure learning problem

• Derive local optima free learning algorithm with theoretical guarantees on statistical consistency

• Part of broader research theme of exploiting linear algebra for probabilistic modeling
Outline

• Motivation

• Intuition and Model

• Learning algorithm

• Experimental results
Intuition

• Consider the following *part-of-speech* tag sequence:

  VBD  DT  NN
  \textit{verb}  \textit{article}  \textit{noun}

• Two possible binary (unlabeled) parses
Intuition

• Consider sentences with this tag sequence:

  VBD  DT  NN
  ate  an  apple
  baked  a  cake
  hit  the  ball
  ran  the  race

• Can we uncover the parse structure based on these sentences?
Intuition

• article(DT) and noun(NN) are dependent
  • *an* = noun is singular and starts with a vowel
  • *a* = noun is singular and starts with constant
  • *the* = noun could be anything

• verb(VBD) and article(DT) not very dependent
  • Choice of article not dependent on choice of verb
Intuition

• article (DT) and noun (NN) are more dependent than verb (VB) and article (DT)
Latent Variable Intuition

\[ P(w_2, w_3 | z, x) = P(w_2 | z, x) P(w_3 | z, x) \]

**plurality/starts with vowel**

- \( w_1 \): ate
- \( w_2 \): an apple
- \( w_3 \): baked a cake
- \( z \): hit the balls
- \( x \): ran the race
Latent Variable Intuition

- Looks a lot like a *constituent parse tree*!!

\[
\begin{align*}
\mathbf{z}_1 & \quad \text{verb/noun semantic class} \\
\mathbf{z}_2 & \quad \text{plurality} \\
& \quad \text{+ noun topic}
\end{align*}
\]

\[
\begin{array}{lll}
\mathbf{w}_1 & \quad \text{ate} & \quad \text{an} & \quad \text{apple} \\
\mathbf{w}_2 & \quad \text{baked} & \quad \text{a} & \quad \text{cake} \\
\mathbf{w}_3 & \quad \text{hit} & \quad \text{the} & \quad \text{balls} \\
& \quad \text{ran} & \quad \text{the} & \quad \text{race}
\end{array}
\]
Our Conditional Latent Tree Model

- Each tag sequence \( \mathbf{x} \) associated with a latent tree

\[
p(\mathbf{w}, \mathbf{z} | \mathbf{x}) = \prod_{i=1}^{H} p(\mathbf{z}_i | \pi_x(\mathbf{z}_i)) \\
\times \prod_{i=1}^{\ell(\mathbf{x})} p(\mathbf{w}_i | \pi_x(\mathbf{w}_i))
\]

\[x = (DT, NN, VBD, DT, NN)\]

The bear ate the fish
Different Tag Sequences Have Different Trees

\[ x_1 = (DT, NN, VBD, DT, NN) \]

\[ x_2 = (DT, NN, VBD, DT, ADJ, NN) \]

The bear ate the fish
A moose ran the race

The bear ate the big fish
The moose ran the tiring race
Mapping Latent Tree To Parse Tree

- Latent tree is undirected. Direct by choosing a split point

- Result is (unlabeled) parse tree
Model Summary

- Each tag sequence \( x \) is associated with a latent tree \( u(x) \)
- \( u(x) \) generates sentences with these tags
- \( u(x) \) can be deterministically mapped to parse tree given a split point

\[
x = (DT, NN, VBD, DT, NN)
\]

The bear ate the fish
A moose ran the race
Outline

• Motivation

• Intuition and Model

• Learning algorithm

• Experimental results
A Structure Learning Problem

- Goal is to learn the most likely undirected latent tree $u(x)$ for each tag sequence $x$ given sentences

```
DT   NN   VB   DT   NN
The   llama   eats   the   grass
A     bug      likes  the  flower
An    orca     chases  the  fish
```

- Assume for now that there are many sentences for each $x$ (we deal with this problem in the paper using kernel smoothing)
Observed Case – Chow Liu Algorithm

• Compute distance matrix between variables

\[ d(w_i, w_j) \]

• Find minimum spanning tree
• Provably optimal
Latent Case

- Not all distances can be computed from data

\[
\begin{align*}
d(w_2, z_1) &\ ? \\
d(w_3, z_1) &\ ? \\
d(w_2, z_1) &\ ?
\end{align*}
\]

- Need a distance function such that the observed distances can be used to recover the latent distances
Problem Traces Back to Phylogenetics

- Existing species like words
- Latent ancestors like bracketing states
Additive Tree Metrics \([\text{Buneman 1974}]\)

\[
d(i, j) = \sum_{(a, b) \in \text{path}(i, j)} d(a, b)
\]

\[
d(w_1, w_3) = d(w_1, z_2) + d(z_1, z_2) + d(w_3, z_1)
\]

*Computable from data*

*not computable from data*
Why Additive Metrics Are Useful

• Given tree structure, we can compute latent distances as a function of observed distances.

\[
d(i, j) = \frac{1}{2} (d(g, b) + d(h, a) - d(g, h) - d(a, b))
\]
Find Minimum Cost Tree

\[ \widehat{u} = \min_u \sum_{(i,j) \in E_u} d(i,j) \]

- This strategy recovers correct tree [Rzhetsky and Nei, 1993]
- Objective is NP-hard in general
- But for special case of projective parse trees, we show tractable dynamic programming algorithm exists [Eisner and Satta 1999].
Spectral Additive Metric For Our Model

• Following distance function is an additive tree metric for our model (adapted from Anandkumar et al. 2011)

\[ d_{\text{spectral}}^x(i,j) = - \log \Lambda_m \left( E[w_i w_j^T | x] \right) \]

where \( \Lambda_m(A) = \prod_{k=1}^{m} \sigma_k(A) \)

• Each \( w_i \) represented by \( p \)-dimensional word embedding
(1) For each tag sequence $\mathbf{x}$, estimate distances $d_{\mathbf{x}}^{\text{spectral}}(i, j) \forall w_i, w_j$

(2) Use dynamic programming to recover minimum cost undirected latent tree

(3) Transform into a parse tree by directing it using the split point $\mathbf{R}$
Theoretical Guarantees

• Our learning algorithm is statistically consistent

• If sentences are generated according to our model then

\[ \text{as } \#\text{sentences } \to \infty, \hat{u}(x) = u(x) \ \forall x \]

with high probability
Outline

• Motivation

• Intuition and Model

• Learning algorithm

• Experimental results
Experiments

• Primary comparison is the Constituent Context Model (CCM) [Klein and Manning 2002].

• We evaluate on three languages
  • English – PennTreebank
  • German – Negra corpus
  • Chinese – Chinese Treebank

• Use heuristic to find split point $R$ to direct our latent trees
English Results

![Graph showing F1 score vs. max sentence length for English results. The graph compares Spectral, Spectral-Oracle, and CCM methods. The Spectral method shows a consistently higher F1 score than the other two methods, especially at higher sentence lengths.]
German Results

![Graph showing F1 score vs max sentence length for German results with Spectral, Spectral-Oracle, and CCM lines.]
Chinese Results

![Graph showing F1 score vs. max sentence length for Spectral, Spectral-Oracle, and CCM methods in Chinese.](image)
Across Languages

Spectral-Oracle

Spectral

CCM

English  German  Chinese

English  German  Chinese

English  German  Chinese
CCM – Random Restarts

CCM Random Restarts (Length <= 10)

Accuracy

Frequency

Random
PCFG
Right branching
Conclusion

• We approach unsupervised parsing as a structure learning problem

• This enables us to develop a local optima free learning algorithm with theoretical guarantees

• Part of a broader research theme that aims to exploit linear algebra perspectives for probabilistic modeling.
Thanks!
| Differences |
|-------------|
| **Unsupervised PCFGs** |
| - Trees are generated by probabilistically combining rules. |
| - Set of rules and rule probabilities *(the grammar)* must be learned from data |
| - Not only **NP-hard**, but also severely **non-identifiable** |

| **Our Model** |
| - There is no grammar. |
| - Each tag sequence deterministically maps to a latent tree. |
| - Intuition is that word correlations can help us uncover the latent tree for each tag sequence. |
| **Identifiable and provable learning algorithm exists** |