Unsupervised Learning of Probabilistic Context-Free Grammar Using Iterative Biclustering

Kewei Tu and Vasant Honavar
Artificial Intelligence Research Laboratory
Department of Computer Science
Iowa State University
www.cs.iastate.edu/~honavar/aigroup.html
www.cild.iastate.edu

Unsupervised Learning of Probabilistic Context-Free Grammar

- Greedy search to maximize the posterior of the grammar given the corpus
- Iterative (distributional) biclustering
- Competitive experimental results

Outline

- Introduction
- Probabilistic Context Free Grammars (PCFG)
- The Algorithm based on Iterative Biclustering (PCFG-BCL)
- Experimental results

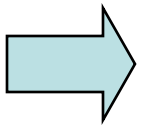
Motivation

- Probabilistic Context-Free Grammar (PCFG) find applications in many areas including:
  - Natural Language Processing
  - Bioinformatics
- Important to learn PCFG from data (training corpus)
- Labeled corpus not always available
- Hence the need for unsupervised learning

Task

- Unsupervised learning of a PCFG from a positive corpus

a square is above the triangle
the square rolls
a triangle rolls
the square rolls
a triangle is above the square
a circle touches a square
the triangle covers the circle
……



S → NP VP
NP → Det N
VP → Vt NP (0.3) | Vi PP (0.2)
| rolls (0.2) | bounces (0.1)
……

PCFG

- Context-free Grammar (CFG)
  - G = (N, Σ, R, S)
    - N: non-terminals
    - Σ: terminals
    - R: rules
    - $S \in N$: the start symbol
- Probabilistic CFG
  - Probabilities on grammar rules

P-CNF

- Probabilistic Chomsky normal form (P-CNF)
  - Two types of rules:
    - A→BC
    - A→a

The AND-OR form

- P-CNF in the AND-OR form
  - Two types of non-terminals: AND, OR
    - AND → OR1 OR2
    - OR → A1 | A2 | a1 | a2 | ……
      - with probabilities

The AND-OR form

- P-CNF in the AND-OR form

CNF

$S \rightarrow a\ (0.4) \mid AB\ (0.6)$
$A \rightarrow a\ (1.0)$
$B \rightarrow b_1\ (0.2) \mid b_2\ (0.5) \mid b_3\ (0.3)$

The AND-OR Form

$\mathrm{OR}_S \rightarrow a\ (0.4) \mid \mathrm{AND}_{AB}\ (0.6)$
$\mathrm{AND}_{AB} \rightarrow \mathrm{OR}_A \mathrm{OR}_B$
$\mathrm{OR}_A \rightarrow a\ (1.0)$
$\mathrm{OR}_B \rightarrow b_1\ (0.2) \mid b_2\ (0.5) \mid b_3\ (0.3)$

The AND-OR form

- P-CNF in the AND-OR form can be divided into two parts
  - Start rules
    - S→…
  - A set of AND-OR groups
    - Each group: AND → OR1 OR2
    - Bijection between ANDs and groups
    - An OR may appear in multiple groups

The AND-OR form

- P-CNF in the AND-OR form can be divided into two parts

CNF

$S \rightarrow a\ (0.4) \mid AB\ (0.6)$
$A \rightarrow a\ (1.0)$
$B \rightarrow b_1\ (0.2) \mid b_2\ (0.5) \mid b_3\ (0.3)$

The AND-OR Form

$\mathrm{OR}_S \rightarrow a\ (0.4) \mid \mathrm{AND}_{AB}\ (0.6)$
$\mathrm{AND}_{AB} \rightarrow \mathrm{OR}_A \mathrm{OR}_B$
$\mathrm{OR}_A \rightarrow a\ (1.0)$
$\mathrm{OR}_B \rightarrow b_1\ (0.2) \mid b_2\ (0.5) \mid b_3\ (0.3)$

Outline

- Introduction
- Probabilistic Context Free Grammars (PCFG)
- The Algorithm based on Iterative Biclustering (PCFG-BCL)
- Experimental results

PCFG-BCL: Outline

- Start with only the terminals
- Repeat the two steps
  - Learn a new AND-OR group by biclustering
  - Attach the new AND to existing ORs
- Post-processing: add start rules

- In principle, these steps are sufficient for learning any CNF grammar

PCFG-BCL: Outline

- Find new rules that yield the greatest increase in the posterior of the grammar given the corpus
- Local search, with the posterior as the objective function
- Use a prior that favors simpler grammars to avoid overfitting

PCFG-BCL

- Repeat the two steps
  - Learn a new AND-OR group by biclustering
  - Attach the new AND to existing ORs
- Post-processing: add start rules

Intuition

- Construct a table *T*
  - Index the rows and columns by symbols appearing in the corpus
  - The cell at row *x* and column *y* records the number of times the pair *xy* appears in the corpus

	is	circle	triangle	square	the	…
below					8	
above					10	
the		24	36	60		
a		12	18	30		
circle	4					
triangle	6					
⋮						

An AND-OR group corresponds to a bicluster

$$\mathrm{AND}_{NP} \rightarrow \mathrm{OR}_{Det}\mathrm{OR}_{N}$$

$$\mathrm{OR}_{Det} \rightarrow \mathrm{the}(0.67) \mid \mathrm{a}(0.33)$$

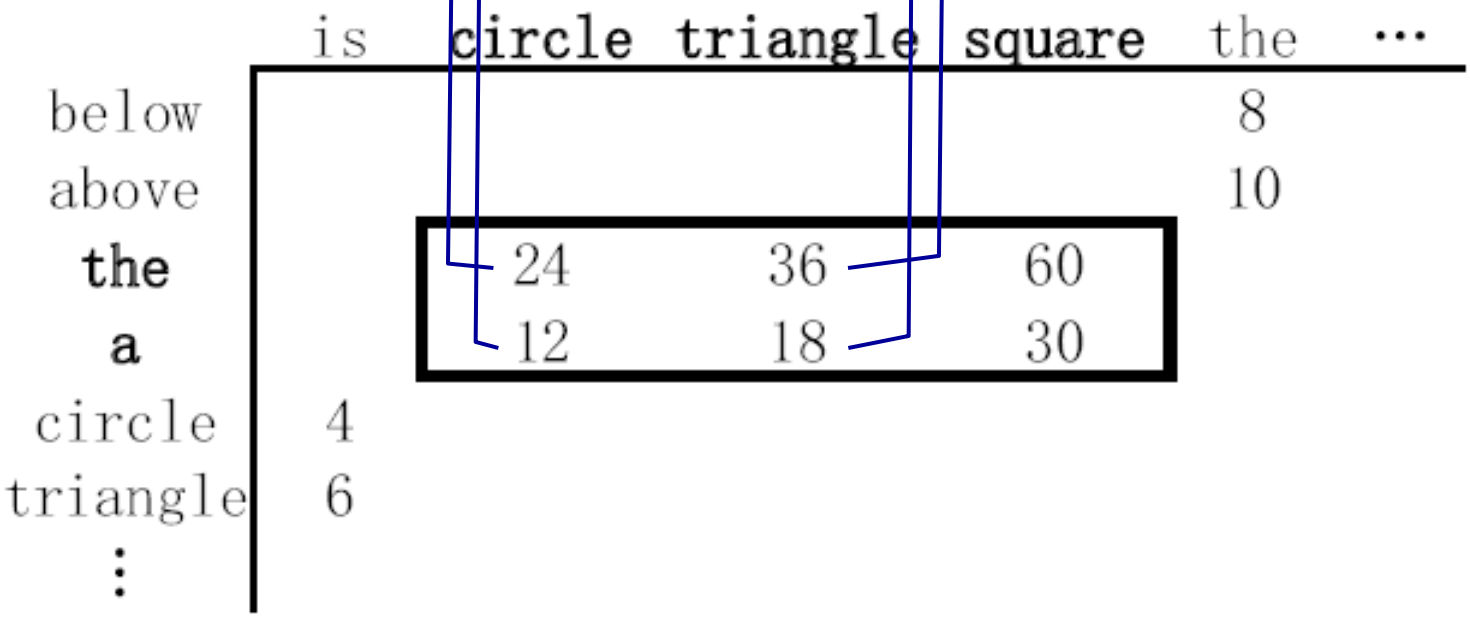
$$\mathrm{OR}_{N} \rightarrow \mathrm{circle}(0.2) \mid \mathrm{triangle}(0.3) \mid \mathrm{square}(0.5)$$

	is	**circle**	**triangle**	**square**	the	…
below					8	
above					10	
the		24	36	60		
a		12	18	30		
circle	4					
triangle	6					
⋮						

The bicluster is multiplicatively coherent

$$\frac{a_{ik}}{a_{jk}} = \frac{a_{il}}{a_{jl}}$$

for any two rows i,j
and two columns k,l



Expression-context matrix of a bicluster

- Each row: a symbol pair contained in the bicluster
- Each column: a context in which the symbol pairs appear in the corpus

	… covers (.)	… touches (.)	… is above (.)	… is below (.)	…
(a, circle)	1	2	1	1	
(a, triangle)	1	2	1	3	
(a, square)	3	4	2	4	…
(the, circle)	2	3	1	3	
(the, triangle)	3	5	2	5	
(the, square)	5				

It's also multiplicatively coherent.

Intuition

- If there's a bicluster that is multiplicatively coherent and has a multiplicatively coherent expression-context matrix
- Then an AND-OR group can be learned from the bicluster

Probabilistic Justification

- Change in likelihood as a result of adding an AND-OR group to a PCFG

$$\max_{P_r} LG(BC) =$$

$$\frac{\prod_{x \in A} {r_x}^{r_x} \prod_{y \in B} {c_y}^{c_y}}{s^s \prod_{\substack{x \in A \\ y \in B}} {a_{xy}}^{a_{xy}}} \times \frac{\prod_{p \in \text{EC-row}} {r'_p}^{r'_p} \prod_{q \in \text{EC-col}} {c'_q}^{c'_q}}{{s'}^{s'} \prod_{\substack{p \in \text{EC-row} \\ q \in \text{EC-col}}} {a'_{pq}}^{a'_{pq}}}$$

Bicluster multiplicative coherence

Expression-context matrix multiplicative coherence

Prior

- To prevent overfitting, use a prior that favors simpler grammars
  - $P(G) \propto 2^{-DL(G)}$
  - DL(G) is the description length of the grammar

Learning a new AND-OR group by biclustering

- find in the table T a bicluster that leads to the maximal posterior gain
- create a new AND-OR group from the bicluster
- reduce the corpus using the new rules
  - E.g., “the circle” is rewritten to the new AND symbol
- update T
  - A new row and column are added for the new AND symbol

PCFG-BCL

- Repeat the two steps
  - Learn a new AND-OR group by biclustering
  - Attach the new AND to existing ORs
- Post-processing: add start rules

Attaching the new AND under existing ORs

- For the new AND symbol N …
  - There may exist OR symbols in the learned grammar, s.t. O→N is in the target grammar
  - Such rules can't be learned in the biclustering step
    - When learning O, N doesn’t exist
    - When learning N, only learn N→AB
- We need an additional step to find such rules
  - Recursion is learned in this step

Intuition

- Adding rule O→N

  = adding a new row/column to the bicluster
- If O→N is true, then
  - the expanded bicluster is multiplicatively coherent
  - the expanded expression-context matrix is multiplicatively coherent

- If we find an OR symbol s.t. the expanded bicluster has this property
- Then a new rule O→N can be added to the grammar

Probabilistic Justification

- Likelihood gain

$$\frac{P(D|G_{k+1})}{P(D|G_k)} \approx LG(\widetilde{BC}')$$

$\widetilde{BC}'$ is an approximation of the expanded bicluster

- To prevent overfitting, the prior is also considered

Attaching the new AND under existing ORs

- Try to find OR symbols that lead to large posterior gain
- When found
  - add the new rule O→N to the grammar
  - do a maximal reduction of the corpus
  - update the table T

PCFG-BCL

- Repeat the two steps
  - Learn a new AND-OR group by biclustering
  - Attach the new AND to existing ORs
- Post-processing: add start rules

Postprocessing

- For each sentence in the corpus:
  - If it's fully reduced to a single symbol x, then add S→x
  - If not, a few options…
- Return the grammar

Outline

- Introduction
- Probabilistic Context Free Grammars (PCFG)
- The Algorithm based on Iterative Biclustering (PCFG-BCL)
- Experimental results

Experiments

- Measurements
  - weak generative capacity
    - precision, recall, F-score
- Test data
  - artificial, English-like CFGs

Grammar Name	Size (in CNF)	Recursion	Source
Num-agr	19 Terminals, 15 Nonterminals, 30 Rules	No	Boogie[12]
Langley1	9 Terminals, 9 Nonterminals, 18 Rules	Yes	Boogie[12]
Langley2	8 Terminals, 9 Nonterminals, 14 Rules	Yes	Boogie[12]
Emile2k	29 Terminals, 15 Nonterminals, 42 Rules	Yes	EMILE[1]
TA1	47 Terminals, 66 Nonterminals, 113 Rules	Yes	ADIOS[5]

Experiment results

[Adriaans, et al., 2000] [Solan, et al., 2005]

Grammar Name	PCFG-BCL			EMILE			ADIOS		
	P	R	F	P	R	F	P	R	F
Num-agr (100)	100 (0)	100 (0)	100 (0)	50 (4)	100 (0)	67 (3)	100 (0)	92 (6)	96 (3)
Langley1 (100)	100 (0)	100 (0)	100 (0)	100 (0)	99 (1)	99 (1)	99 (3)	94 (4)	96 (2)
Langley2 (100)	98 (2)	100 (0)	99 (1)	96 (3)	39 (7)	55 (7)	76 (21)	78 (14)	75 (14)
Emile2k (200)	85 (3)	90 (2)	87 (2)	75 (12)	68 (4)	71 (6)	80 (0)	65 (4)	71 (3)
Emile2k (1000)	100 (0)	100 (0)	100 (0)	76 (7)	85 (8)	80 (6)	75 (3)	98 (3)	85 (3)
TA1 (200)	82 (7)	73 (5)	77 (5)	77 (3)	14 (3)	23 (4)	77 (24)	55 (12)	62 (14)
TA1 (2000)	95 (6)	100 (1)	97 (3)	98 (5)	48 (4)	64 (4)	50 (22)	92 (4)	62 (17)

P=Precision, R=Recall, F=F-score

Number in the parentheses: standard deviation

- PCFG-BCL outperforms EMILE and ADIOS
  - with lower standard deviations

Summary

- An unsupervised PCFG-learning algorithm
  - It acquires new grammar rules by iterative biclustering on a table of symbol pairs
  - In each step it tries to maximize the increase of the posterior of the grammar
  - Competitive experimental results

Work in progress

- Alternative strategies for optimizing the objective function
- Evaluation on and adaptation to real world applications (e.g., natural language), wrt. both weak and strong generative capacity

Thank you~

Backup…

Step 1

Posterior gain:

$$\max_{P_r} LPG(BC) = \max_{P_r} \log \frac{P(G_{k+1}|D)}{P(G_k|D)}$$

$=$

Likelihood Gain

$+$

$+$ **Prior gain** (bias towards large BC)

Step 2 Intuition

- Remember O is learned by extracting a bicluster
- adding rule O→N

  = adding a new row/column to the bicluster

Expanding the bicluster

$$\mathrm{AND} \rightarrow \mathrm{OR}_1\,\mathrm{OR}_2$$
$$\mathrm{OR}_1 \rightarrow \mathrm{big}\ (0.6) \mid \mathrm{old}\ (0.4)$$
$$\mathrm{OR}_2 \rightarrow \mathrm{dog}\ (0.6) \mid \mathrm{cat}\ (0.4)$$

New rule: $\mathrm{OR}_2 \rightarrow \mathrm{AND}$

	dog	cat	AND
big	27	18	15
old	18	12	10

The expanded bicluster should still be multiplicatively coherent

Step 2 Intuition

- Expression-context matrix
  - adding rule O→N

    = adding a set of new rows to the E-C matrix

Expanding the expression-context matrix

	the (.) slept.	the big (.) slept.	the old (.) slept.	the old big (.) slept.	… heard the (.)	… heard the old (.)	…
(old, dog)	6	1	1	0	3	1	
(big, dog)	9	2	1	1	4	1	…
(old, cat)	4	1	0	0	2	1	
(big, cat)	6	1	1	0	4	1	
(old, AND)	3	1	0	0	2	1	…
(big, AND)	5	1	1	0	2	1	

The expanded expression-context matrix should still be multiplicatively coherent.

Step 2

- Likelihood gain:

$$\frac{P(D|G_{k+1})}{P(D|G_k)} \approx LG(\widetilde{BC}')$$

$\widetilde{BC}'$: the expected numbers of appearance of the symbol pairs when applying the current grammar to expand the current partially reduced corpus.

Grammar selection/averaging

- Run the algorithm for multiple times to get multiple grammars
- Use the posterior of the grammars to do model selection/averaging
- Experimental results:
  - Improved the performance
  - Decreased the standard deviations

Time Complexity

$$O(N^2k^2c + h^{d+1}c\omega m^2)$$

- N: # of ANDs
- k: average # of rules headed by an OR
- c: average column# of Expr-Cont Matrix
- h: average # of ORs that produce an AND or terminal
- d: a recursion depth limit
- ω: sentence# in the corpus
- m: average sentence length

biclustering vs. distributional clustering

	John (.) tea	John (.) coffee	John (.) eating	John makes (.)	John likes (.)	John is (.)
makes	x	x				
likes	x	x	x			
is			x			
tea				x	x	
coffee				x	x	
eating					x	x

Figure from [Adriaans, et al., 2000]

V1 → makes | likes
V2 → likes | is

biclustering vs. substitutability heuristic

	John (.) tea	John (.) coffee	John (.) eating	John makes (.)	John likes (.)	John is (.)
makes	x	x				
likes	x	x	x			
is			x			
tea				x	x	
coffee				x	x	
eating					x	x

Figure from [Adriaans, et al., 2000]

N1 → tea | coffee
N2 → eating

size: 10*12

	the	is	a	below	circle	bounces	covers	triangle	square	rolls	above	touches
the					16			39	61			
is				9							16	
a					12			15	26			
below	8		1									
circle		3				1	6			2		4
covers	11		8									
triangle		8				3	4			9		9
square		14				3	9			13		12
above	15		1									
touches	17		8									

size: 10*12

	the	is	a	below	circle	bounces	covers	triangle	square	rolls	above	touches
the					16			39	61			
is				9							16	
a					12			15	26			
below	8		1									
circle		3				1	6			2		4
covers	11		8									
triangle		8				3	4			9		9
square		14				3	9			13		12
above	15		1									
touches	17		8									

A set of multiplicatively coherent biclusters, which represent a set of AND-OR groups in the grammar.

Related work

- Unsupervised CFG learning
  - EMILE [Adriaans et al., 2000]
  - ABL [Zaanen, 2000]
  - [Clark, 2001; 2007]
  - ADIOS [Solan et al., 2005]
- Main difference
  - Distributional biclustering
  - A unified method for different types of rules

Related work

- Unsupervised PCFG learning
  - Inside-outside
  - [Stolcke&Omohundro, 1994]
  - [Chen 1995]
  - [Kurihara&Sato, 2004; 2006]
  - [Liang et al., 2007]
- Main difference
  - Different prior
  - Structure search method

Related work

- Unsupervised parsing (*not CFG*)
  - [Klein&Manning, 2002; 2004]
  - U-DOP [Bod, 2006]