Parsing

Weighted Deductive Parsing

Laura Kallmeyer

Heinrich-Heine-Universität Düsseldorf

Winter 2017/18

Table of contents

Idea (1)

Idea of weighted deductive parsing Nederhof (2003):

- Give a deductive definition of the probability of a parse tree.
- Use Knuth's algorithm to compute the best parse tree for category S and a given input w.

Advantage:

- Yields the best parse without exhaustive parsing.
- Can be used to parse any grammar formalism as long as an appropriate weighted deductive system can be defined.

Idea (2)

Reminder:

- Parsing Schemata understand parsing as a deductive process.
- Deduction of new items from existing ones can be described using inference rules.
- General form:

$$\frac{\textit{antecedent}}{\textit{consequent}}\ \textit{side conditions}$$

  antecedent, consequent: lists of items
- Application: if antecedent can be deduced and side condition holds, then the consequent can be deduced as well.

Idea (3)

A parsing schema consists of

- deduction rules;
- an axiom (or axioms), can be written as a deduction rule with empty antecedent;
- and a goal item.

The parsing algorithm succeeds if, for a given input, it is possible to deduce the goal item.

Idea (4)

Example: Deduction-based definition of bottom-up CFG parsing (CYK) with Chomsky Normal Form.
For an input $w = w_1 \cdots w_n$ with $|w| = n$,

1. Item form $[A, i, j]$ with A a non-terminal, $1 \leq i \leq j \leq n$.
2. Deduction rules:

   Scan: $\dfrac{}{[A, i, i]} \; A \rightarrow w_i$

   Complete: $\dfrac{[B, i, j], [C, j+1, k]}{[A, i, k]} \; A \rightarrow B\,C$
3. Goal item: $[S, 1, n]$.

Idea (5)

Extension to a weighted deduction system:

- Each item has an additional weight. Intuition: weight = costs to build an item. (Usually, the higher the costs, the lower the probability.)
- The deduction rules specify how to compute the weight of the consequent item form the weights of the antecedent items.

Extending CYK with weights:

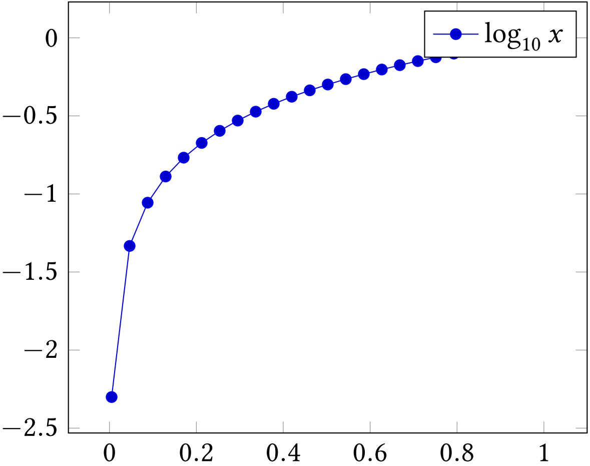
$$\text{Scan: } \frac{}{|log(p)| : [A, i, i]} \quad p : A \rightarrow w_i$$

$$\text{Complete: } \frac{x_1 : [B, i, j], x_2 : [C, j+1, k]}{x_1 + x_2 + |log(p)| : [A, i, k]} \quad p : A \rightarrow B\, C$$

(Note that $p_1 \cdot p_2 = 10^{log_{10}(p_1)} \cdot 10^{log_{10}(p_2)} = 10^{log_{10}(p_1) + log_{10}(p_2)}$.)

Idea (6)

Reminder: $\log 1 = 0, \log 0 = -\infty$



Algorithm (1)

- There is a linear order $<$ defined on the weights.
- The lower the weight, the better the item.
- For Knuth's algorithm, the weight functions f must be monotone nondecreasing in each variable and $f(x_1, \ldots, x_m) \geq max(x_1, \ldots, x_m)$.

In our example, this is the case:

$$\text{Complete: } \frac{x_1 : [B, i, j], x_2 : [C, j+1, k]}{x_1 + x_2 + |log(p)| : [A, i, k]} \; p : A \rightarrow B\, C$$

$f(x_1, x_2) = x_1 + x_2 + c$ where $c \geq 0$ is a constant.

Algorithm (2)

Algorithm for computing the goal item with the lowest weight, goes back to Knuth.
Goal: Find possible items with their lowest possible weight.
We need two sets:

- A set $\mathcal{C}$ (the chart) that contains items that have reached their final weight.
- A set $\mathcal{A}$ (the agenda) that contains items that are waiting to be processed as possible antecendents in further rule applications and that have not necessarily reached their final weight.

Initially, $\mathcal{C} = \emptyset$ and $\mathcal{A}$ contains all items that can be deduced from an empty antecedent set.

Algorithm (3)

```
while A ≠ ∅ do
  remove the best item x : I from A
        and add it to C
  if I goal item
  then stop and output true
  else
    for all y : I′ deduced from x : I and
          items in C:
      if there is no z with z : I′ ∈ C or z : I′ ∈ A
      then add y : I′ to A
      else if z : I′ ∈ A for some z
        then update weight of I′ in A to min(y, z)
```

Algorithm (4)

If the weight functions are as required, then the following is guaranteed:

- Whenever an item is the best in the agenda, you have found its lowest weight.
- Therefore, if this item is a goal item, then you have found the best parse tree for your input.
- If it is no goal item, you can store it in the chart.

$\Rightarrow$ no exhaustive parsing needed.
However: $\mathcal{A}$ needs to be treated as a priority queue which can be expensive.

CYK Example

.6 S→SS (.2) .1 S→SA (1) .3 S→a (.5) .4 A→a (.4) .6 A→b (.2)
(the number in brackets is the $|log|$)

Input: aa

Chart	Agenda
	$.4 : [A, 1, 1]$, $.4 : [A, 2, 2]$, $.5 : [S, 1, 1]$, $.5 : [S, 2, 2]$
$.4 : [A, 1, 1]$	$.4 : [A, 2, 2]$, $.5 : [S, 1, 1]$, $.5 : [S, 2, 2]$
$.4 : [A, 1, 1]$, $.4 : [A, 2, 2]$	$.5 : [S, 1, 1]$, $.5 : [S, 2, 2]$
$.4 : [A, 1, 1]$, $.4 : [A, 2, 2]$,$.5 : [S, 1, 1]$	$.5 : [S, 2, 2]$, $1.9 : [S, 1, 2]$
$.4 : [A, 1, 1]$, $.4 : [A, 2, 2]$,$.5 : [S, 1, 1]$ $.5 : [S, 2, 2]$	$1.2 : [S, 1, 2]$

Parsing

Extension to parsing:

- Whenever we generate a new item, we store it not only with its weight but also with backpointers to its antecedent items.
- Whenever we update the weight of an item, we also have to update the backpointers.

In order to read off the best parse tree, we have to start from the best goal item and follow the backpointers.

Left Corner Example

Deduction rules with weights (goal item $[S, 0, n]$):

Scan: $$\frac{}{0 : [a, i, i+1]} \; w_{i+1} = a$$

Left Corner Predict: $$\frac{x : [A, i, j]}{x : [B \to A \bullet \alpha]} \; B \to A\alpha \in P$$

Complete: $$\frac{x_1 : [A \to \alpha \bullet B\beta, i, j], x_2 : [B, j, k]}{x_1 + x_2 : [A \to \alpha B \bullet \beta, i, k]}$$

Convert: $$\frac{x : [B \to \gamma \bullet, j, k]}{x + |log(p)| : [B, j, k]} \; p : B \to \gamma$$

Left Corner Example

.6 S→SS (.2) .1 S→SA (1) .3 S→a (.5) .4 A→a (.4) .6 A→b (.4)

Input: aa

1. $\mathcal{A} = \{0 : [a, 0, 1], 0 : [a, 1, 2]\}$

Chart:

j			
2			
1			
0			
	0	1	2 i

Left Corner Example

2. $\mathcal{A} = \{0 : [a, 1, 2], 0 : [S \to a\bullet, 0, 1], 0 : [A \to a\bullet, 0, 1]\}$

Chart:

2			
1	0 : *a*		
0			
	0	1	2

3. $\mathcal{A} = \{0 : [S \to a\bullet, 0, 1], 0 : [S \to a\bullet, 1, 2], 0 : [A \to a\bullet, 0, 1], 0 : [A \to a\bullet, 1, 2]\}$

Chart:

2		0 : *a*	
1	0 : *a*		
0			
	0	1	2

Left Corner Example

4. $\mathcal{A} = \{0 : [A \to a\bullet, 0, 1], 0 : [A \to a\bullet, 1, 2], .5 : [S, 0, 1], .5 : [S, 1, 2]\}$

Chart:

2		$0 : a, 0 : S \to a\bullet$	
1	$0 : a, 0 : S \to a\bullet$		
0			
	0	1	2

(two convert operations)

5. $\mathcal{A} = \{.4 : [A, 0, 1], .4 : [A, 1, 2], .5 : [S, 0, 1], .5 : [S, 1, 2]\}$

Chart:

2		$0 : a, 0 : S \to a\bullet$ $0 : A \to a\bullet$	
1	$0 : a, 0 : S \to a\bullet$ $0 : A \to a\bullet$		
0			
	0	1	2

(two convert operations)

Left Corner Example

6. $\mathcal{A} = \{.5 : [S, 0, 1], .5 : [S, 1, 2]\}$

Chart:

2		$0 : a, 0 : S \to a\bullet$ $0 : A \to a\bullet, .4 : A$	
1	$0 : a, 0 : S \to a\bullet$ $0 : A \to a\bullet, .4 : A$		
0			
	0	1	2

(two steps)

Left Corner Example

7. $\mathcal{A} = \{.5 : [S \to S \bullet A, 0, 1], .5 : [S \to S \bullet S, 0, 1],$
$.5 : [S \to S \bullet A, 1, 2], .5 : [S \to S \bullet S, 1, 2]\}$

Chart:

2		$0 : a, 0 : S \to a\bullet$ $0 : A \to a\bullet, .4 : A, .5 : S$	
1	$0 : a, 0 : S \to a\bullet$ $0 : A \to a\bullet, .4 : A, .5 : S$		
0			
	0	1	2

Left Corner Example

8. $\mathcal{A} = \{.5 : [S \rightarrow S \bullet S, 0, 1], .5 : [S \rightarrow S \bullet A, 1, 2],$
$.5 : [S \rightarrow S \bullet S, 1, 2], .9 : [S \rightarrow SA\bullet, 0, 2]\}$

Chart:

2		$0 : a, 0 : S \rightarrow a\bullet$ $0 : A \rightarrow a\bullet, .4 : A, .5 : S$	
1	$0 : a, 0 : S \rightarrow a\bullet$ $0 : A \rightarrow a\bullet, .4 : A, .5 : S$ $.5 : S \rightarrow S \bullet A$		
0			
	0	1	2

Left Corner Example

9. $\mathcal{A} = \{.9 : [S \to SA\bullet, 0, 2], 1 : [S \to SS\bullet, 0, 2]\}$

Chart:

2		$0 : a$, $0 : S \to a\bullet$ $0 : A \to a\bullet$, $.4 : A$, $.5 : S$ $.5 : S \to S \bullet A$, $.5 : S \to S \bullet S$	
1	$0 : a$, $0 : S \to a\bullet$ $0 : A \to a\bullet$, $.4 : A$, $.5 : S$ $.5 : S \to S \bullet A$, $.5 : S \to S \bullet S$		
0			
	0	1	2

Left Corner Example

10. $\mathcal{A} = \{1 : [S \to SS\bullet, 0, 2], 1.9 : [S, 0, 2]\}$

Chart:

2	$.9 : S \to SA\bullet$	$0 : a, 0 : S \to a\bullet$ $0 : A \to a\bullet, .4 : A, .5 : S$ $.5 : S \to S \bullet A, .5 : S \to S \bullet S$	
1	$0 : a, 0 : S \to a\bullet$ $0 : A \to a\bullet, .4 : A, .5 : S$ $.5 : S \to S \bullet A, .5 : S \to S \bullet S$		
0			
	0	1	2

Left Corner Example

11. $\mathcal{A} = \{1.2 : [S, 0, 2]\}$
Chart:

2	$.9 : S \to SA\bullet$ $1 : S \to SS\bullet$	$0 : a, 0 : S \to a\bullet$ $0 : A \to a\bullet, .4 : A, .5 : S$ $.5 : S \to S \bullet A, .5 : S \to S \bullet S$	
1	$0 : a, 0 : S \to a\bullet$ $0 : A \to a\bullet, .4 : A, .5 : S$ $.5 : S \to S \bullet A, .5 : S \to S \bullet S$		
0			
	0	1	2

Left Corner Example

12. $\mathcal{A} = \emptyset$

Chart:

2	$.9 : S \to SA\bullet$ $1 : S \to SS\bullet$ $1.2 : S$	$0 : a$, $0 : S \to a\bullet$ $0 : A \to a\bullet$, $.4 : A$, $.5 : S$ $.5 : S \to S \bullet A$, $.5 : S \to S \bullet S$	
1	$0 : a$, $0 : S \to a\bullet$ $0 : A \to a\bullet$, $.4 : A$, $.5 : S$ $.5 : S \to S \bullet A$, $.5 : S \to S \bullet S$		
0			
	0	1	2

Nederhof, Mark-Jan. 2003. Weighted Deductive Parsing and Knuth's Algorithm. *Computational Linguistics* 29(1). 135–143.