

Unsupervised grammar induction with Minimum Description Length

Roni Katzir
trifilij@mit.edu

MIT Linguistics

Infolab Meeting
September 6, 2006

Introduction

- The problems of learning
- Implications
- Descriptions

Learner

- Architecture
- Results

Discussion

Introduction

The problems of learning

Implications

Descriptions

Learner

Architecture

Results

Discussion

Problems

- ▶ Analytical truths not learnable (Plato)
- ▶ Induction not justifiable (Hume)
- ▶ Interesting families of languages not identifiable in the limit (Gold)
- ▶ Search space too big (Peirce)
- ▶ Input limited, noisy, non-representative (Chomsky)

Plato

- ▶ Two kinds of search:
 - ▶ Search for what you know – futile
 - ▶ Search for what you do not know – impossible
- ▶ The geometry lesson: We can acquire mathematical knowledge without being taught

Hume

- ▶ No necessity that the future will conform to the past
- ▶ Causality is never observed
- ▶ Using previous cases of induction to support the next one is circular
- ▶ Hume: induction not logically justifiable, but useful for survival
- ▶ But when is it useful?

Grue and Bleen

- ▶ Goodman: not all observations give rise to generalizations
 - ▶ grue = green and was examined before t or blue and not examined before t
 - ▶ bleen = blue and was examined before t or blue and not examined before t
- ▶ Set a future t (say, t = September 7, 2006). All emeralds examined so far are grue.
- ▶ Conclusion: future ones are likely to be grue as well
- ▶ Problem: doesn't seem quite right
- ▶ But why? And what makes green and blue better?

Gold

- ▶ If real learning is impossible, redefine success:
 - ▶ Set C of candidate hypotheses (= languages) given in advance
 - ▶ Clean input guaranteed: every element in the correct hypothesis appears somewhere; nothing else will be observed
 - ▶ Segmented input: each observation unambiguously delimited
 - ▶ Indefinitely late commitment: wrong guesses allowed, as long as after some point all guesses are identical and correct
- ▶ However (Gold, 1967): for any interesting family of hypotheses, this weak notion of learnability is still not enough

Gold (contd.)

- ▶ In particular, C is not identifiable in the limit if it contains a family $\{L_i\}_{i \geq 0}$ where $L_i \subset L_{i+1}, i \geq 0$ as well as $L_\infty = \cup_{i \geq 0} L_i$
- ▶ Intuitive reason:
 - ▶ If L_i was used, a non-conservative guess $L_j, j > i$ will never be corrected by future observations.
 - ▶ On the other hand, a conservative guesser will never guess L_∞
- ▶ More formally, if a guesser g can identify correctly all of the L_i , we can construct a sequence of observations in L_∞ that g will necessarily fail to identify

Empirical problems

- ▶ Even finite search spaces can be too big
- ▶ Noise
- ▶ Insufficient data
- ▶ Unsegmented input

Introduction

The problems of learning

Implications

Descriptions

Learner

Architecture

Results

Discussion

Linguistic implications

- ▶ Explanatory adequacy
- ▶ Radical UG - finitely many languages
- ▶ Principles & Parameters
- ▶ Rules vs. Statistics

Computational implications

- ▶ Supervised learning
- ▶ Restricted domains
- ▶ Task-specific heuristics

Introduction

The problems of learning
Implications
Descriptions

Learner

Architecture
Results

Discussion

Ingredients

- ▶ Parsimony: do not multiply entities beyond necessity
- ▶ Descriptions:

"I maintain that a universal is not something real that exists in a subject [of inherence], either inside or outside the mind, but that it has being only as a thought-object in the mind. It is a kind of mental picture which as a thought-object has a being similar to that which the thing outside the mind has in its real existence."

– William of Ockham, ca. 1285–1349 in: P. Boehner, 1957/1990, p. 41, Hackett Publishing.

Algorithmic probability, Kolmogorov complexity

- ▶ Solomonoff (1960,1964)
- ▶ Kolmogorov (1965)
- ▶ Chaitin (1966)

Main idea

- ▶ The Kolmogorov complexity of a string D is the length of the shortest computer program that prints out D and then halts
- ▶ In what programming language?
 - ▶ Answer 1: it doesn't matter. Only an additive constant involved.
 - ▶ Answer 2: it matters a lot. We know nothing about the constant.
- ▶ Another concern: not computable

Computable approximations

- ▶ MML (Wallace and Boulton, 1968)
- ▶ MDL (Rissanen 1978)

Introduction

The problems of learning
Implications
Descriptions

Learner

Architecture
Results

Discussion

Main issues

- ▶ Space: any subset of Context-Free Grammars
- ▶ Objective: minimize total description length
- ▶ Search:
 - ▶ Problem: many local optima
 - ▶ Approach: Simulated Annealing (Kirkpatrick et al., 1983)

Search

► Initial state

 $D := \text{pabikugolatuda...}$ $T := T_0$ $G := \begin{cases} \gamma \rightarrow p \gamma \\ \gamma \rightarrow a \gamma \\ \gamma \rightarrow b \gamma \\ \vdots \end{cases}$

Search

► Repeat:

$$G' := \text{random_neighbor}(G)$$

$$\Delta := \text{Energy}(G', D) - \text{Energy}(G, D)$$

$$p := \begin{cases} 1 & \Delta \leq 0 \\ e^{-\frac{\Delta}{T}} & \Delta > 0 \end{cases}$$

$$G := G' \text{ with probability } p$$

$$T := \alpha T$$

Search

► Repeat:

$$G' := \text{random_neighbor}(G)$$

$$\Delta := \text{Energy}(G', D) - \text{Energy}(G, D)$$

$$p := \begin{cases} 1 & \Delta \leq 0 \\ e^{-\frac{\Delta}{T}} & \Delta > 0 \end{cases}$$

$$G := G' \text{ with probability } p$$

$$T := \alpha T$$

Random Neighbor

- ▶ Insert
- ▶ Delete
- ▶ New rule
- ▶ Split
- ▶ Substitute

Random Neighbor

Insert:

$$G := \begin{cases} A \rightarrow B C \\ B \rightarrow D E \end{cases} \Rightarrow G := \begin{cases} A \rightarrow B X C \\ B \rightarrow D E \end{cases}$$

Random Neighbor

Delete:

$$G := \begin{cases} A \rightarrow B C \\ B \rightarrow D E \end{cases} \Rightarrow G := \begin{cases} A \rightarrow B \\ B \rightarrow D E \end{cases}$$

Random Neighbor

New rule:

$$G := \begin{cases} A \rightarrow B C \\ B \rightarrow D E \end{cases} \Rightarrow G := \begin{cases} A \rightarrow B C \\ B \rightarrow D E \\ Y \rightarrow \end{cases}$$

Random Neighbor

Substitute:

$$G := \begin{cases} A \rightarrow B C \\ B \rightarrow D E \end{cases} \Rightarrow G := \begin{cases} A \rightarrow D E C \\ B \rightarrow D E \end{cases}$$

Random Neighbor

Split:

$$G := \begin{cases} A \rightarrow B C \\ B \rightarrow D E \end{cases} \Rightarrow G := \begin{cases} A \rightarrow Z C \\ B \rightarrow D E \\ Z \rightarrow B \end{cases}$$

Search

- ▶ Repeat:

$$G' := \text{random_neighbor}(G)$$

$$\Delta := \text{Energy}(G', D) - \text{Energy}(G, D)$$

$$p := \begin{cases} 1 & \Delta \leq 0 \\ e^{-\frac{\Delta}{T}} & \Delta > 0 \end{cases}$$

$$G := G' \text{ with probability } p$$

$$T := \alpha T$$

Search

► Repeat:

$$G' := \text{random_neighbor}(G)$$

$$\Delta := \text{Energy}(G', D) - \text{Energy}(G, D)$$

$$p := \begin{cases} 1 & \Delta \leq 0 \\ e^{-\frac{\Delta}{T}} & \Delta > 0 \end{cases}$$

$$G := G' \text{ with probability } p$$

$$T := \alpha T$$

Energy

- ▶ Total description length:

$$\text{Energy}(G, D) := |G| + |\text{code}(D|G)|$$

Energy

- ▶ Total description length:

$$\text{Energy}(G, D) := |G| + |\text{code}(D|G)|$$

- ▶ Measuring the grammar:

$$G := \left\{ \begin{array}{l} A \rightarrow B A \\ A \rightarrow B \\ B \rightarrow C D \\ \vdots \\ E \rightarrow F G \end{array} \right.$$

Energy

- ▶ Total description length:

$$\text{Energy}(G, D) := |G| + |\text{code}(D|G)|$$

- ▶ Measuring the grammar:

$$G := \left\{ \begin{array}{l} A \rightarrow B A \\ A \rightarrow B \\ B \rightarrow C D \\ \vdots \\ E \rightarrow F G \end{array} \right.$$

- ▶ G as parameter:

$$G := \text{ABA\#AB\#BCD\#\dots\#EFG\#\#}$$

Energy

- ▶ Codes for categories:

#	000
A	001
⋮	⋮
G	111

- ▶ Preamble: length of category code $k = \lceil \lg(|Categories| + 1) \rceil$

ABA#AB#BCD#...#EFG##

$$= \underbrace{000}_k 1 \underbrace{001}_k \underbrace{010}_k \underbrace{001}_k \underbrace{000}_k \dots \underbrace{000}_k$$

$$|G| \approx k \cdot \left[\sum_{r \in G} |r| + 1 \right]$$

Energy

$$\text{Energy}(G, D) := |G| + |\text{code}(D|G)|$$

Energy

$$\text{Energy}(G, D) := |G| + |\text{code}(D|G)|$$

Energy

$$\text{Energy}(G, D) := |G| + |\text{code}(D|G)|$$

- ▶ Group rules by left-hand side
- ▶ Enumerate expansions:

Rule	Code
$A \rightarrow BC$	00
$A \rightarrow BB$	01
$A \rightarrow C$	10
$B \rightarrow a$	0
$B \rightarrow b$	1
$C \rightarrow c$	ϵ
\vdots	\vdots

Energy

- ▶ $T = [A[B \dots] [C \dots]]$
- ▶ Pre-order:
 $C(T) = C(A)C(A \rightarrow BC | A)C(\dots | B)\dots C(\dots | C)\dots$

Some practical points

- ▶ Feasibility: parsing of arbitrary CFG is $O(n^3)$
- ▶ For long inputs: remove $X \rightarrow \epsilon$, $X \rightarrow X \Delta$
- ▶ For longer inputs: parse only with linear rules
- ▶ Grammar size: arbitrary cutoff at some N

Incrementality

- ▶ Symbolic representation: easy to see what we have
- ▶ If we know something about the language, we can use it at any point
- ▶ Easy to add, modify

Introduction

The problems of learning
Implications
Descriptions

Learner

Architecture
Results

Discussion

Case 1:pabiku

- ▶ Background:
- ▶ Saffran et al. (1996): word segmentation by 8-month old infants
- ▶ Vocabulary: pabiku, golatu, daropi, tibudo
- ▶ Text: concatenation of words from the vocabulary
- ▶ Speech synthesizer, flat intonation, no word-breaks
- ▶ 2 minutes = 180 words = 1080 segments

pabiku results: babies

- ▶ After exposure to text, infants distinguish between words (e.g. pabiku) and non-words that appear in the text (e.g. bikuda)
- ▶ Conclusion: infants use statistical information to segment words
- ▶ In particular (Aslin et al., 1998): use of transitional probabilities
- ▶ Unit of representation: the syllable

pabiku results: MDL learner

- ▶ Input: as in the experiment of Saffran et al.
- ▶ Difference: 60 words instead of 180
- ▶ Result:

pabiku results: MDL learner

- ▶ Input: as in the experiment of Saffran et al.
- ▶ Difference: 60 words instead of 180
- ▶ Result:

Step 37: current temp. = 14.994450998883487 Grammar:

$\gamma \rightarrow g \gamma; \gamma \rightarrow l \gamma; \gamma \rightarrow a \gamma; \gamma \rightarrow p \gamma; \gamma \rightarrow k \gamma; \gamma \rightarrow . \gamma; \gamma \rightarrow$
 $t \gamma; \gamma \rightarrow u \gamma; \gamma \rightarrow o \gamma; \gamma \rightarrow d \gamma; \gamma \rightarrow b \gamma; \gamma \rightarrow i \gamma; \gamma \rightarrow$
 $r \gamma; l \rightarrow$

Grammar length: 146 Encoding length: 1442

pabiku results: MDL learner

- ▶ Input: as in the experiment of Saffran et al.
- ▶ Difference: 60 words instead of 180
- ▶ Result:

Step 191: current temp. = 14.971377200361164 Grammar:

$k \rightarrow b \gamma; \gamma \rightarrow g \gamma; \gamma \rightarrow k u \gamma; \gamma \rightarrow l \gamma; \gamma \rightarrow d \gamma; \gamma \rightarrow$
 $a \gamma; \gamma \rightarrow p \gamma; \gamma \rightarrow i \gamma; \gamma \rightarrow b \gamma; \gamma \rightarrow t \gamma; \gamma \rightarrow u \gamma; \gamma \rightarrow$
 $o \gamma; \gamma \rightarrow r \gamma$

Grammar length: 142 Encoding length: 1382

pabiku results: MDL learner

- ▶ Input: as in the experiment of Saffran et al.
- ▶ Difference: 60 words instead of 180
- ▶ Result:

Step 618: current temp. = 14.907585393190937

Grammar:

$k \rightarrow k$ g 23; $i \rightarrow p$; 23 $\rightarrow o$; $\gamma \rightarrow$; $\gamma \rightarrow g$ γ ; $\gamma \rightarrow p$ a γ ; $\gamma \rightarrow$
 a γ ; $\gamma \rightarrow t$ γ ; $\gamma \rightarrow u$ γ ; $\gamma \rightarrow o$ γ ; $\gamma \rightarrow k$ u γ ; $\gamma \rightarrow o$ l γ ; $\gamma \rightarrow$
 o p γ ; $\gamma \rightarrow d$ γ ; $\gamma \rightarrow b$ γ ; $\gamma \rightarrow i$ γ ; $\gamma \rightarrow r$ γ ; $o \rightarrow$; $t \rightarrow g$

Grammar length: 200 Encoding length: 1199

pabiku results: MDL learner

- ▶ Input: as in the experiment of Saffran et al.
- ▶ Difference: 60 words instead of 180
- ▶ Result:

Step 1406: current temp. = 14.790574662993796 Grammar:

$i \rightarrow l; r \rightarrow p; \gamma \rightarrow k u \gamma; \gamma \rightarrow p a \gamma; \gamma \rightarrow d \gamma; \gamma \rightarrow a \gamma; \gamma \rightarrow l \gamma; \gamma \rightarrow b \gamma; \gamma \rightarrow r o p \gamma; \gamma \rightarrow t \gamma; \gamma \rightarrow g o l a t \gamma; \gamma \rightarrow u \gamma; \gamma \rightarrow o \gamma; \gamma \rightarrow r \gamma; o \rightarrow i; t \rightarrow a b b \gamma r \gamma a k p$

Grammar length: 200 Encoding length: 971

pabiku results: MDL learner

- ▶ Input: as in the experiment of Saffran et al.
- ▶ Difference: 60 words instead of 180
- ▶ Result:

Step 5635: current temp. = 14.178119831796465

Grammar: $k \rightarrow k; \gamma \rightarrow k u \gamma \gamma; \gamma \rightarrow b u d o \gamma; \gamma \rightarrow$
 $p a b i k u \gamma; \gamma \rightarrow t i \gamma; \gamma \rightarrow d a r o p i \gamma; \gamma \rightarrow i \gamma g; \gamma \rightarrow$
 $g o l a t u \gamma; d \rightarrow \gamma; b \rightarrow r u k; b \rightarrow t; t \rightarrow \gamma$

Grammar length: 176 Encoding length: 225

pabiku results: MDL learner

- ▶ Input: as in the experiment of Saffran et al.
- ▶ Difference: 60 words instead of 180
- ▶ Result:

Step 5837: current temp. = 14.149508793558308

Grammar: $k \rightarrow p; \gamma \rightarrow t i b u d o \gamma; \gamma \rightarrow \gamma; \gamma \rightarrow$
 $p a b i k u \gamma; \gamma \rightarrow i a \gamma \gamma g; \gamma \rightarrow d a r o p i \gamma; \gamma \rightarrow$
 $g o l a t u \gamma; b \rightarrow u d o o \gamma; b \rightarrow k u l; d \rightarrow; l \rightarrow; t \rightarrow \gamma$

Grammar length: 179 Encoding length: 183

More structure?

- ▶ So far, just a concatenation of words from a lexicon
- ▶ No real use of grammar
- ▶ Slightly more challenging: $a*b^*$

Case 2: a^*b^*
$$\begin{aligned}\gamma &\rightarrow A B \\ A &\rightarrow aaaa A \mid A1 \\ A1 &\rightarrow a A \mid \epsilon \\ B &\rightarrow bbbb B \mid B1 \\ B1 &\rightarrow b B \mid \epsilon\end{aligned}$$

Case 2: a^*b^*

- ▶ Results using actual grammar:
 - ▶ Grammar length: 74 Encoding length: 1831 Energy: 1905
- ▶ Concatenation grammar:
 - ▶ $\gamma \rightarrow a\gamma; \gamma \rightarrow a\gamma$
 - ▶ Grammar length: 11 Encoding length: 3153 Energy: 3164
- ▶ After 2000 steps:
 - ▶ $\gamma \rightarrow a\gamma\gamma a^5\gamma; \gamma \rightarrow b^{11}\gamma b b\gamma a^6 b$
 - ▶ Grammar length: 48 Encoding length: 1967 Energy: 2015

More structure

- ▶ More complex grammar
- ▶ Larger vocabulary
- ▶ More rules

Case 3: aturtlekillsmax

 $S \rightarrow NP VP$ $NP \rightarrow Nm \mid D N$ $Nm \rightarrow max \mid sam \mid kim \mid bill \mid mary$ $D \rightarrow the \mid a$ $N \rightarrow man \mid dog \mid cat \mid turtle$ $VP \rightarrow Vin \mid Vtr NP \mid Vtl NP CP \mid Vsy CP$ $Vin \rightarrow walks \mid runs$ $Vtr \rightarrow kills \mid hits$

...

aturtlekillsmax

▶ Sample sentences:

```
aturtleknowsthatsteltellskimthattheturtlewalksands  
amkillsmax.kimknowsthatsteltellsmaxthatkimtellsk  
imthatkimhitstheman.kimtellsaturtlethatmaxruns
```

aturtlekillsmax

► Results:

Step : 423000 Temperature : 26.2

Grammar: $1 \rightarrow aman1$; $1 \rightarrow hits1o$; $m \rightarrow h$; $1 \rightarrow bill1m$; $m \rightarrow au$; $m \rightarrow \epsilon$; $1 \rightarrow or1$; $1 \rightarrow knowsthat1u$; $1 \rightarrow urtle1$; $a \rightarrow n$; $1 \rightarrow eman$; $m \rightarrow a$; $1 \rightarrow heui$; $1 \rightarrow edog$; $1 \rightarrow saysthat1$; $t \rightarrow e$; $a \rightarrow x$; $1 \rightarrow wak1e$; $a \rightarrow o$; $1 \rightarrow and1$; $1 \rightarrow \epsilon$; $1 \rightarrow tells1$; $1 \rightarrow walks1c$; $1 \rightarrow raac$; $1 \rightarrow runs1$; $a \rightarrow uu$; $1 \rightarrow acat1cx$; $1 \rightarrow x$; $1 \rightarrow kim1ky$; $m \rightarrow o$; $a \rightarrow \epsilon$; $1 \rightarrow th11$; $1 \rightarrow at1w$; $m \rightarrow r$; $1 \rightarrow mall$; $1 \rightarrow et$; $1 \rightarrow r$; $1 \rightarrow ecat$; $1 \rightarrow adog1ty$; $1 \rightarrow inksthat$; $1 \rightarrow ry$; $1 \rightarrow sam1os$; $1 \rightarrow kills1$; $1 \rightarrow a$
Grammar length: 778 Encoding length: 10620 Energy: 11398

aturtlekillsmax

► Results:

Step : 423000 Temperature : 26.2

1 → *aman*1; 1 → *hits*1o; m → h; 1 → *bill*1m; m → au; m → ε; 1 → or1; 1 → *knowsthat*1u; 1 → *urtle*1; a → n; 1 → *eman*; m → a; 1 → *heui*; 1 → *edog*; 1 → *saysthat*1; t → e; a → x; 1 → wak1e; a → o; 1 → *and*1; 1 → ε; 1 → *tells*1; 1 → *walks*1c; 1 → raac; 1 → *runs*1; a → uu; 1 → *acat*1cx; 1 → x; 1 → *kim*1ky; m → o; a → ε; 1 → th11; 1 → at1w; m → r; 1 → mall1; 1 → et; 1 → r; 1 → ecat; 1 → *adog*1ty; 1 → *inksthat*; 1 → ry; 1 → *sam*1os; 1 → *kills*1; 1 → a

Grammar length: 778 Encoding length: 10620 Energy: 11398

aturtlekillsmax

- ▶ With concatenation grammar:
 - ▶ Grammar length: 247 Encoding length: 34480 Energy: 34727
- ▶ With actual grammar:
 - ▶ Grammar length: 590 Encoding length: 5508 Energy: 6098

Introduction

The problems of learning
Implications
Descriptions

Learner

Architecture
Results

Discussion

Interim conclusions

- ▶ Accurate segmentation
- ▶ Inaccurate structural learning
- ▶ Objective function seems correct
- ▶ Search not robust enough
- ▶ Possible solutions:
 - ▶ Cooling agenda
 - ▶ Neighbors
 - ▶ Humans

Cooling agenda

- ▶ Start with higher temperature
- ▶ Slow down cooling agenda
- ▶ In theory, convergence guaranteed
- ▶ In practice?

Neighbors

- ▶ Current operations:
 - ▶ Insert
 - ▶ Delete
 - ▶ Merge
 - ▶ Split
- ▶ Possible additions:
 - ▶ Context-dependent abstraction
 - ▶ Collapse categories
 - ▶ ...
- ▶ Concern: choice of operations arbitrary
- ▶ Better option: make neighbors related to the representation

Human help

- ▶ The representations are readable. Why not use that?
- ▶ Sometimes we have some prior knowledge about the input (a few words, a grammatical construction, etc.)
- ▶ Other times we can say something after segmentation has taken place
- ▶ Any piece of knowledge can help

Summing up

- ▶ Learning requires structural representations
- ▶ Representations encode their own parameters and the data
- ▶ In theory, nothing more is needed
- ▶ In practice, this is true for some tasks but not for others
 - ▶ True for segmentation
 - ▶ Not true for non-concatenative structure
- ▶ Main problem: search is too difficult
- ▶ Things that can help:
 - ▶ Restricted search space
 - ▶ Better starting point
 - ▶ Time