Statistical learning of grammars

Mark Johnson

Brown University

BUCLD 2005

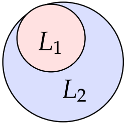
Outline

What is statistical learning?

- Statistical learners learn from statistical distributional properties of input
  - not just whether something occurs (logical learning), but how often
  - assumes input follows some (unknown) probability distribution
- Statistical learning (a.k.a. machine learning) is a separate field
  - mathematical theories relating learning goal with statistics
  - most informative statistic depends on:
    - what learner is trying to learn
    - current state of learner
  - *much more than transitional probabilities!*

Vapnik (1998) *Statistical Learning Theory*

Statistical learning and implicit negative evidence



- Logical approach to acquisition
  - No negative evidence

  $\Rightarrow$ *subset problem:* guess L_2 when true lg is L_1
- Statistical approach to learning
  - if $L_2 - L_1$ is *expected* to occur but doesn't
    $\Rightarrow$ L_2 is probably wrong
  - implicit negative evidence
  - *succeeds where logical learning fails* (e.g., PCFGs)
    - stronger input assumptions (follows distribution)
    - weaker success criteria (probabilistic)
- Both logic and statistics are kinds of inference
  - statistical inference uses more information from input

Units of generalization in learning

1. Colorless green ideas sleep furiously.
2. *Furiously sleep ideas green colorless.

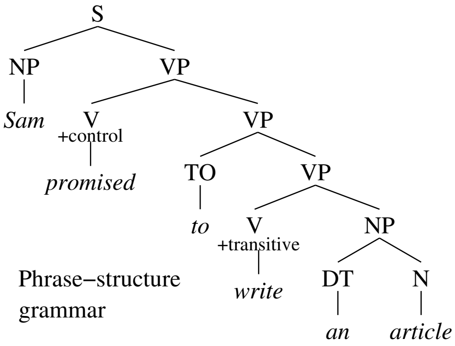
- Both *sentences* have zero frequency
  $\Rightarrow$ frequency $\neq$ well-formedness
- Hidden class *bigram model*

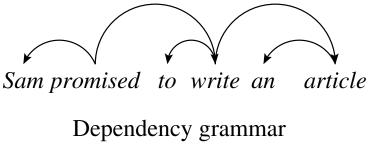
$$
\begin{aligned}
&\mathsf{P}(\text{colorless green ideas sleep furiously}) \\
&\quad = \mathsf{P}(\text{colorless})\mathsf{P}(\text{green}|\text{colorless})\ldots \\
&\quad = 2 \times 10^5 \times \mathsf{P}(\text{furiously sleep ideas green colorless})
\end{aligned}
$$

Chomsky (1957) *Syntactic Structures*
Pereira (2000) "Formal grammar and information theory: Together again?"

What are the right units of generalization?

- *grammars are tools for investigating different units of generalization*
- grammars can model wide variety of phenomena
  - various types of grammatical dependencies
  - word segmentation (Brent)
  - syllable structure (Goldwater and Johnson)
  - morphological dependencies (Goldsmith)





Why grammars?

1. Useful for both production and comprehension
2. Compositional representations seem necessary for semantic interpretation
3. *Curse of dimensionality:* the number of possibly related entities grows exponentially
   - 1,000 words = 1,000 unigrams, 1,000,000 bigrams, 1,000,000,000 trigrams, ... (*sparse data*)
   - grammars identify relationships to generalize over
   - sparse data problems are more severe with larger, more specialized representations
4. *"Glass-box" models:* (you can see inside) the learner's assumptions and conclusions are explicit

Outline

Probabilistic Context-Free Grammars

- The *probability* of a tree is the product of the probabilities of the rules used to construct it

1.0	S → NP VP	1.0	VP → V
0.75	NP → George	0.25	NP → Al
0.6	V → barks	0.4	V → snores

$$\mathrm{P}\left(\text{[S [NP George] [VP [V barks]]]}\right) = 0.45$$

$$\mathrm{P}\left(\text{[S [NP Al] [VP [V snores]]]}\right) = 0.1$$

There are stochastic variants of most grammars

- Grammar generates *candidate structures* (e.g., string of words, trees, OT candidates, construction grammar analyses, minimalist derivations, . . .)
- Associate *numerical weights* with *configurations* that occur in these structures
  - pairs of adjacent words
  - rules used to derive structure
  - constructions occuring in structure
  - P&P parameters (e.g., HEADFINAL)
- Combine (e.g., multiply) the weights of configurations occuring in a structure to get its *score*

Abney (1997) "Stochastic Attribute-Value Grammars"

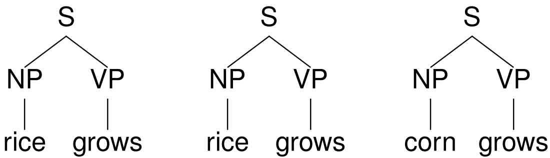
Outline

Learning as optimization

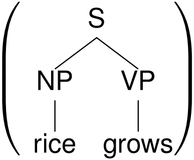
- Pick a task that the correct grammar should be able to do well
  - predicting sentences and their structures (supervised learning)
  - predicting the (next) words in sentences (unsupervised learning)
- Find weights that optimize performance on task
- Searching for optimal weights is usually easier than searching for optimal categorical grammars

Rummelhart and McClelland (1986) *Parallel Distributed Processing*
Tesar and Smolensky (2000) *Learnability in Optimality Theory*

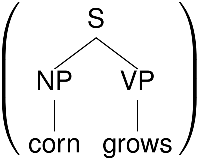
Learning PCFGs from trees (supervised)



Rule	**Count**	**Rel Freq**
S $\rightarrow$ NP VP	3	1
NP $\rightarrow$ rice	2	$2/3$
NP $\rightarrow$ corn	1	$1/3$
VP $\rightarrow$ grows	3	1

P 

$= 2/3$

P 

$= 1/3$

Rel freq is *maximum likelihood estimator* (selects rule probabilities that maximize probability of trees)

Grammars and generalizations

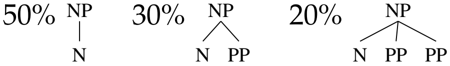
- Grammar determines units of generalization

Grammars and generalizations

- Grammar determines units of generalization
  - *Training data:* 50%: N, 30%: N PP, 20%: N PP PP

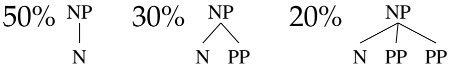
Grammars and generalizations

- Grammar determines units of generalization
  - *Training data:* 50%: N, 30%: N PP, 20%: N PP PP
  - with flat rules NP $\rightarrow$ N, NP $\rightarrow$ N PP, NP $\rightarrow$ N PP PP predicted probabilities replicate training data

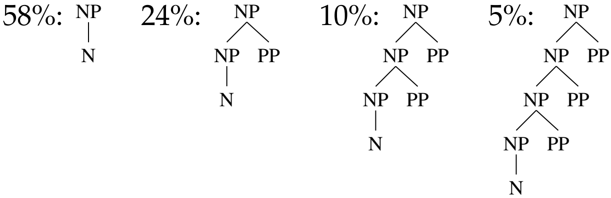


Grammars and generalizations

- Grammar determines units of generalization
  - *Training data:* 50%: N, 30%: N PP, 20%: N PP PP
  - with flat rules NP → N, NP → N PP, NP → N PP PP predicted probabilities replicate training data

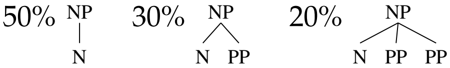


  - but with adjunction rules NP → N, NP → NP PP

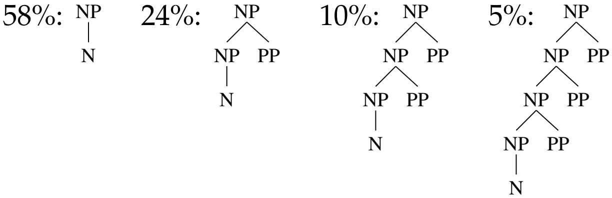


Grammars and generalizations

- Grammar determines units of generalization
  - *Training data:* 50%: N, 30%: N PP, 20%: N PP PP
  - with flat rules NP → N, NP → N PP, NP → N PP PP predicted probabilities replicate training data



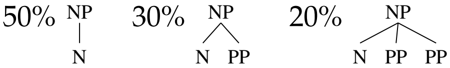
  - but with adjunction rules NP → N, NP → NP PP



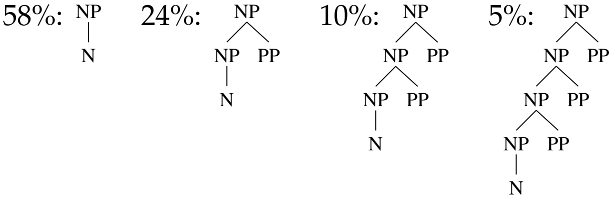
- Finding *best units of generalization*

Grammars and generalizations

- Grammar determines units of generalization
  - *Training data:* 50%: N, 30%: N PP, 20%: N PP PP
  - with flat rules NP → N, NP → N PP, NP → N PP PP predicted probabilities replicate training data



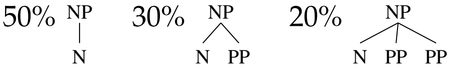
  - but with adjunction rules NP → N, NP → NP PP



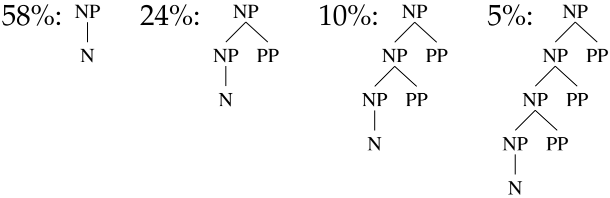
- Finding *best units of generalization*
  - Predicate and argument structure in Lexicalized Tree-Adjoining Grammar

Grammars and generalizations

- Grammar determines units of generalization
  - *Training data:* 50%: N, 30%: N PP, 20%: N PP PP
  - with flat rules NP → N, NP → N PP, NP → N PP PP
    predicted probabilities replicate training data



  - but with adjunction rules NP → N, NP → NP PP



- Finding *best units of generalization*
  - Predicate and argument structure in Lexicalized Tree-Adjoining Grammar
  - Head-argument dependencies in Dependency Grammar

Outline

Learning from words alone (unsupervised)

- Training data consists of strings of words w
- Optimize grammar's ability to predict w: find grammar that makes w as likely as possible
- *Expectation maximization* is an iterative procedure for building unsupervised learners out of supervised learners
  - parse a bunch of sentences with current guess at grammar
  - weight each parse tree by its probability under current grammar
  - estimate grammar from these weighted parse trees as before
- Each iteration is *guaranteed* not to decrease $\mathsf{P}(w)$ (but can get trapped in local minima)

Dempster, Laird and Rubin (1977) "Maximum likelihood from incomplete data via the EM algorithm"

Expectation Maximization with a toy grammar

Initial rule probs

rule	**prob**
…	…
VP $\rightarrow$ V	0.2
VP $\rightarrow$ V NP	0.2
VP $\rightarrow$ NP V	0.2
VP $\rightarrow$ V NP NP	0.2
VP $\rightarrow$ NP NP V	0.2
…	…
Det $\rightarrow$ the	0.1
N $\rightarrow$ the	0.1
V $\rightarrow$ the	0.1

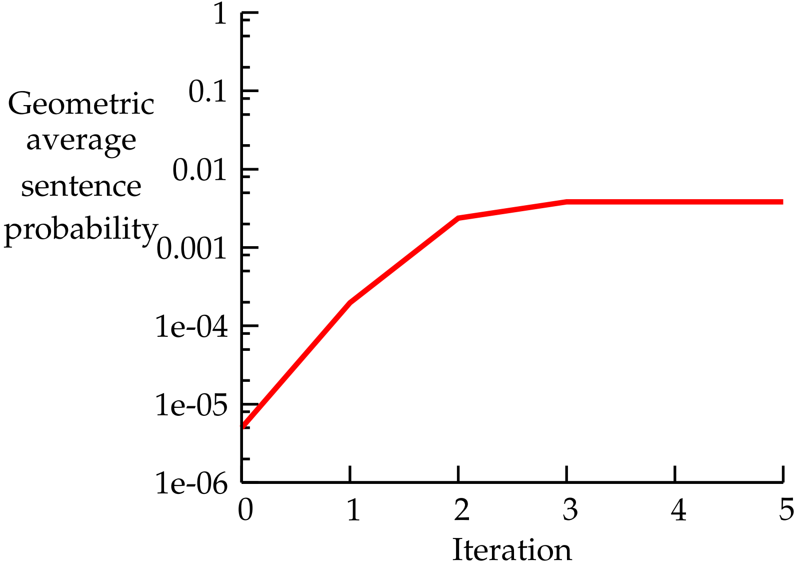
"English" input

the dog bites
the dog bites a man
a man gives the dog a bone
…

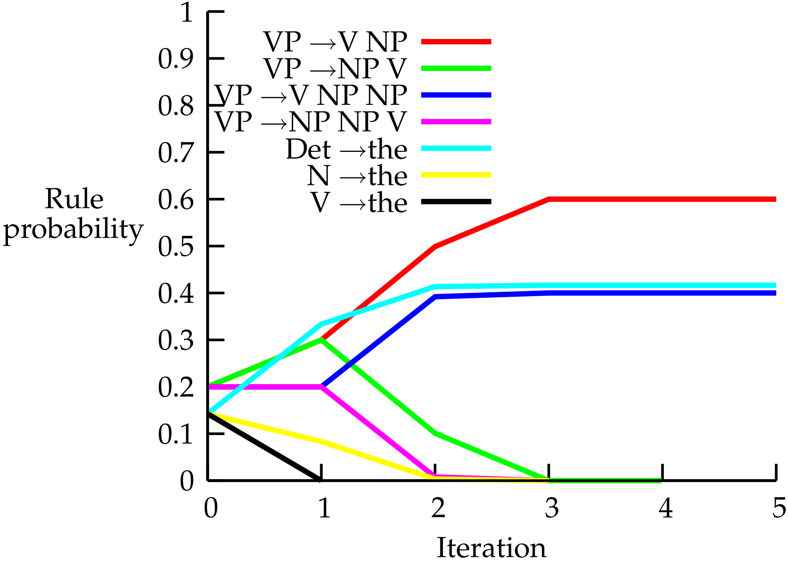
"pseudo-Japanese" input

the dog bites
the dog a man bites
a man the dog a bone gives
…

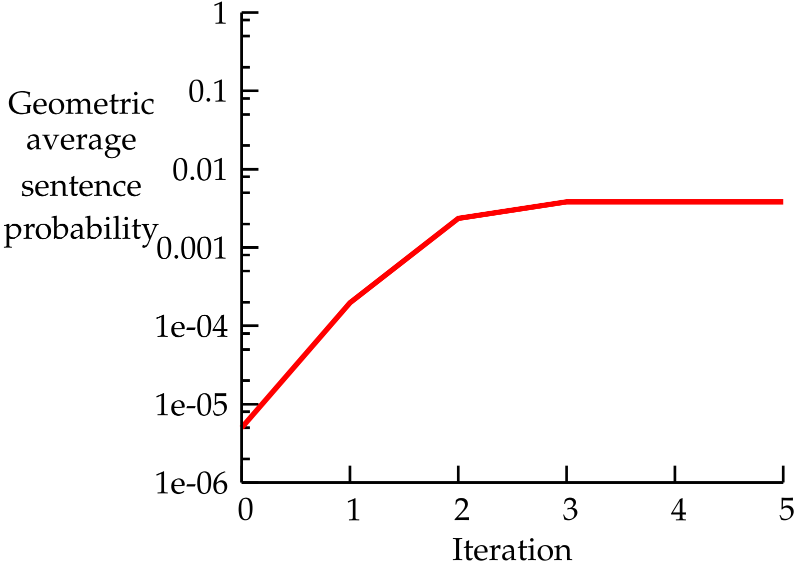
Probability of "English"



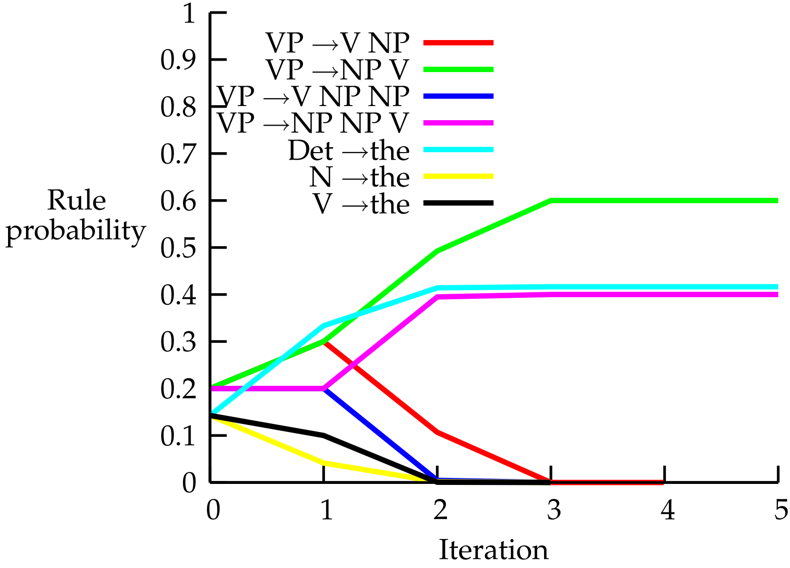
Rule probabilities from “English”



Probability of “Japanese”



Rule probabilities from “Japanese”



Statistical grammar learning

- Simple algorithm: learn from your best guesses
  - requires learner to parse the input
- "Glass box" models: learner's prior knowledge and learnt generalizations are *explicitly represented*
- Optimization of smooth function of rule weights ⇒ learning can involve small, incremental updates
- Learning structure (rules) is hard, but …
- Parameter estimation can approximate rule learning
  - start with "superset" grammar
  - estimate rule probabilities
  - discard low probability rules

The importance of starting small

- EM works by learning from its own parses
  - Each parse is weighted by its probability
  - Rules used in high-probability parses receive strong reinforcement
- In grammar-based models, ambiguity grows with sentence length
  - longer sentences are typically highly ambiguous

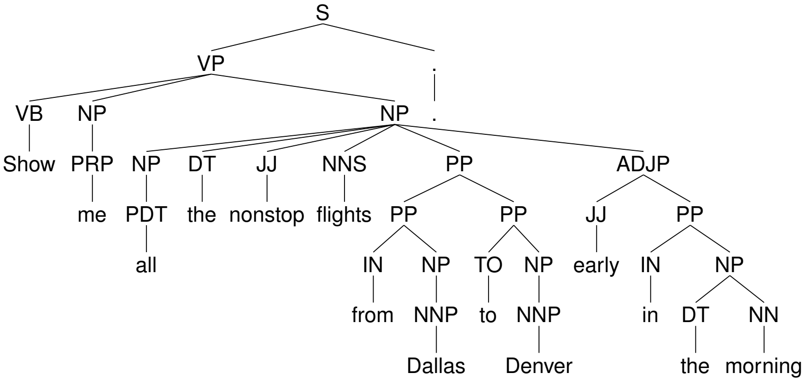
  $\Rightarrow$ lower average parse probability

  $\Rightarrow$ less clear information about which rules are most useful

Outline

Applying EM learning to real language

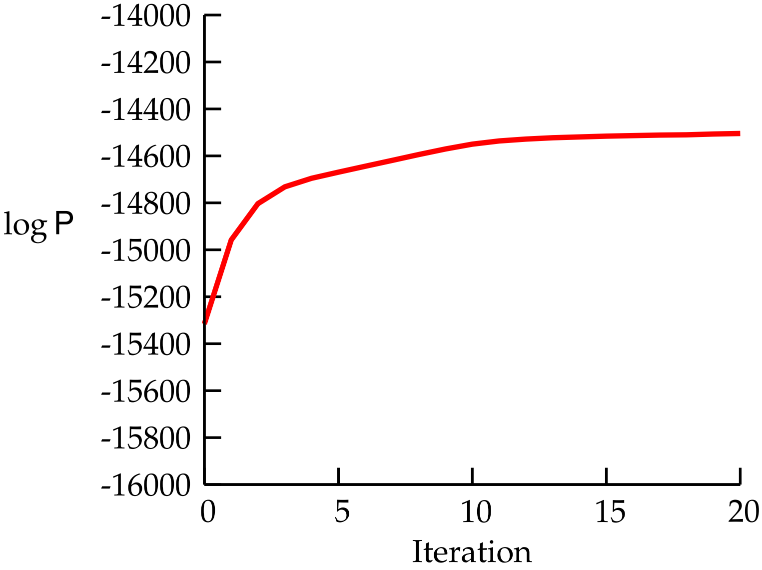
- ATIS treebank consists of 1,300 hand-constructed parse trees
- ignore the words (in this experiment)
- about 1,000 PCFG rules are needed to build these trees



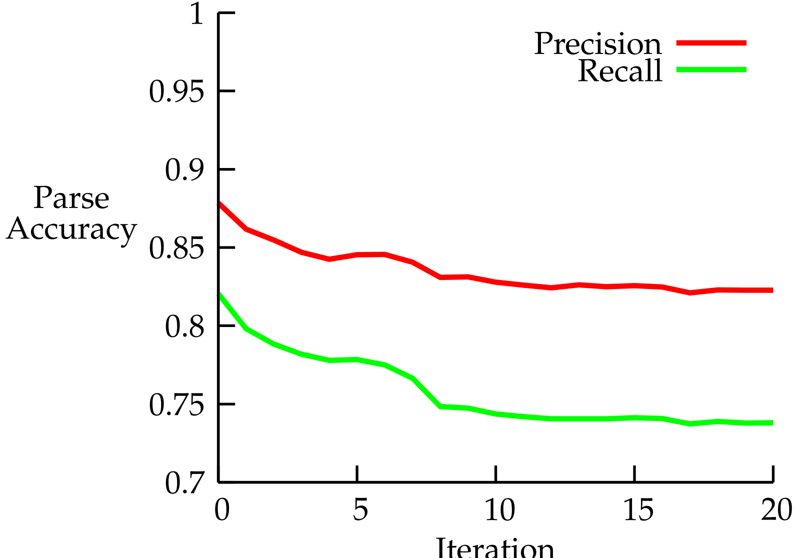
Training from real language

1. Extract productions from trees and estimate probabilities probabilities from trees to produce PCFG.
2. Initialize EM with the treebank grammar and MLE probabilities
3. Apply EM (to strings alone) to re-estimate production probabilities.
4. At each iteration:
   - Measure the likelihood of the training data and the quality of the parses produced by each grammar.
   - Test on training data (so poor performance is not due to overlearning).

Probability of training strings



Accuracy of parses produced using the learnt grammar



Discussion

- Predicting words $\neq$ finding correct structure
- Why didn't the learner find the right structures?
  - Grammar *ignores semantics* (Zettlemoyer and Collins)
  - Predicting words is wrong objective
  - Wrong kind of grammar (Klein and Manning)
  - Wrong training data (Yang)
  - Wrong learning algorithm (much work in CL and ML)

de Marken (1995) "Lexical heads, phrase structure and the induction of grammar"

Outline

Introduction

Stochastic grammars

Supervised learning

Unsupervised learning

Learning a real language

Conclusion

Summary

- Statistical learning *extracts more information from input*
- *Curse of dimensionality:* something must guide learner to focus on correct generalizations
- Stochastic versions of most kinds of grammar
- Statistical grammar learning combines:
  - compositional representations
  - optimization-based learning
- *Glass box:* grammars use explicit representations
  - generalizations learnt
  - prior knowledge assumed
  - predicting the input $\neq$ correctly analysing the input
- Applied to psycholinguistics (Jurafsky, Crocker)
- Should be useful for child language

Bayesian learning

- A statistical learning framework that integrates:
  - *likelihood of the data* (prediction)
  - bias or *prior knowledge* (e.g., innate constraints)
- "hard" priors ignore some analyses, focus on others
- "soft" priors bias learner toward certain hypotheses
  - markedness constraints (e.g., syllables have onsets)
  - can be over-ridden by sufficient data
- evaluate *different kinds of universals*

Grammars in computational linguistics

1980s: hand-written linguistic grammars on linguistically interesting examples

early 1990s: simple statistical models dominate speech recognition and computational linguistics

- they can *learn*
- corpus-based evaluation methodology

late 1990s: techniques for statistical learning of probabilitstic grammars

today: loosely linguistic grammar-based approaches are competitive, but so are non-grammar-based approaches