

Learning grammar(s) statistically

Mark Johnson

Brown University

Hawaii 2005

Outline

Introduction

Stochastic grammars

Supervised learning

Unsupervised learning

Applying this to real data

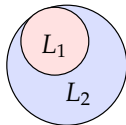
Factoring learning into simpler components

Conclusion

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!*

Statistical learning and implicit negative evidence



- ▶ Logical approach to acquisition
 - ▶ No negative evidence
 - ⇒ *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
 - ⇒ 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
⇒ frequency \neq well-formedness
- ▶ Hidden class *bigram model*

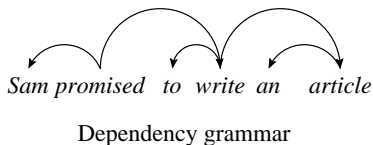
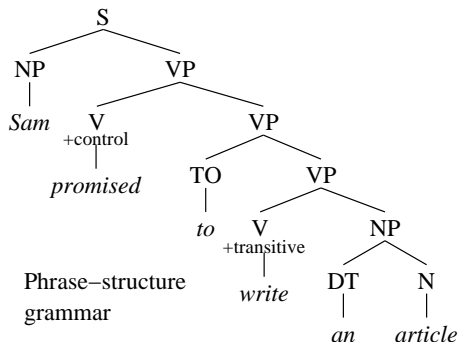
$$\begin{aligned} & P(\text{colorless green ideas sleep furiously}) \\ &= P(\text{colorless})P(\text{green}|\text{colorless}) \dots \\ &= 2 \times 10^5 \times P(\text{furiously sleep ideas green colorless}) \end{aligned}$$

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

Introduction

Stochastic grammars

Supervised learning

Unsupervised learning

Applying this to real data

Factoring learning into simpler components

Conclusion

Probabilistic Context-Free Grammars

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

1.0 $S \rightarrow NP VP$

0.75 $NP \rightarrow \text{George}$

0.6 $V \rightarrow \text{barks}$

1.0 $VP \rightarrow V$

0.25 $NP \rightarrow \text{AI}$

0.4 $V \rightarrow \text{snores}$

$$P \left(\begin{array}{c} S \\ \swarrow \quad \searrow \\ NP \quad VP \\ | \quad | \\ \text{George} \quad V \\ | \\ \text{barks} \end{array} \right) = 0.45$$

$$P \left(\begin{array}{c} S \\ \swarrow \quad \searrow \\ NP \quad VP \\ | \quad | \\ \text{AI} \quad V \\ | \\ \text{snores} \end{array} \right) = 0.1$$

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 occurring in structure
 - ▶ P&P parameters (e.g., HEADFINAL)
- ▶ Combine (e.g., multiply) the weights of configurations occurring in a structure to get its *score*

Outline

Introduction

Stochastic grammars

Supervised learning

Unsupervised learning

Applying this to real data

Factoring learning into simpler components

Conclusion

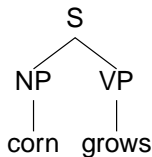
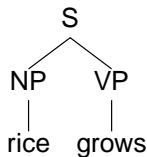
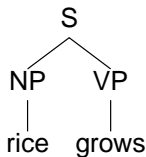
Learning as optimization

- ▶ 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
$NP \rightarrow \text{rice}$	2	$2/3$
$NP \rightarrow \text{corn}$	1	$1/3$
$VP \rightarrow \text{grows}$	3	1

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

$$P \left(\begin{array}{c} S \\ / \quad \backslash \\ NP \quad VP \\ | \quad | \\ \text{rice} \quad \text{grows} \end{array} \right) = 2/3$$

$$P \left(\begin{array}{c} S \\ / \quad \backslash \\ NP \quad VP \\ | \quad | \\ \text{corn} \quad \text{grows} \end{array} \right) = 1/3$$

Grammars and generalizations

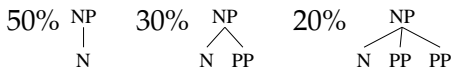
- ▶ Grammar determines units of generalization

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 $NP \rightarrow N$, $NP \rightarrow NPP$, $NP \rightarrow NPPP$
predicted probabilities replicate training data

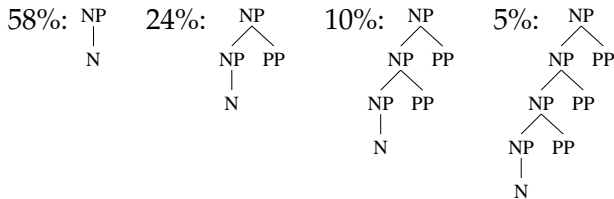


Grammars and generalizations

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

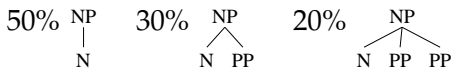


- ▶ but with adjunction rules $NP \rightarrow N$, $NP \rightarrow NP PP$

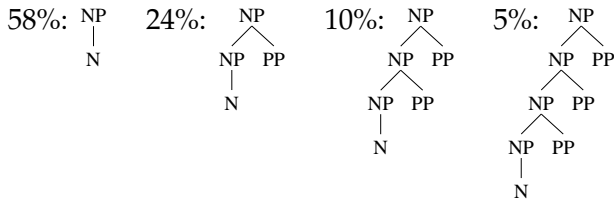


Grammars and generalizations

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



- ▶ but with adjunction rules $NP \rightarrow N$, $NP \rightarrow NP PP$



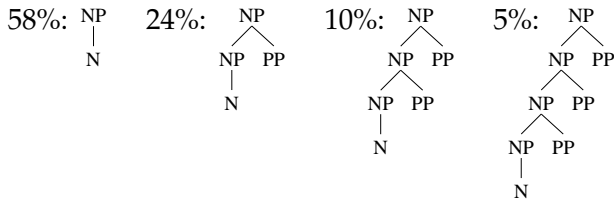
- ▶ Finding *best units of generalization*

Grammars and generalizations

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



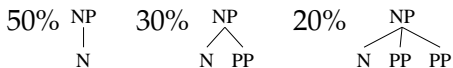
- ▶ but with adjunction rules $NP \rightarrow N$, $NP \rightarrow NP PP$



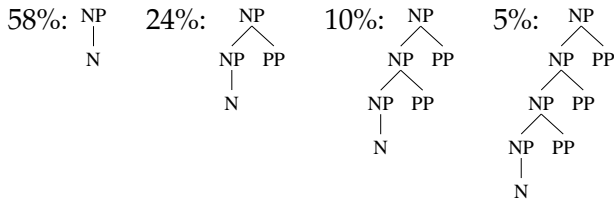
- ▶ Finding *best units of generalization*
 - ▶ Predicate and argument structure in Lexicalized Tree-Adjoining Grammar

Grammars and generalizations

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

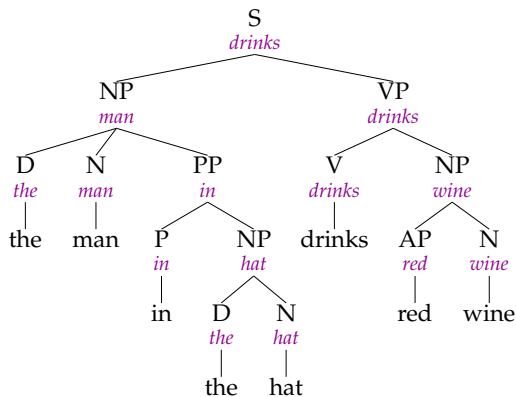


- ▶ but with adjunction rules $NP \rightarrow N$, $NP \rightarrow NP PP$



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

Lexical “head-to-head” dependencies



Rules:

$S \rightarrow NP \ VP$
drinks \rightarrow *man* *drinks*

$VP \rightarrow V \ NP$
drinks \rightarrow *drinks* *wine*

$NP \rightarrow AP \ N$
wine \rightarrow *red* *wine*

...

- ▶ *Lexicalization* captures a wide variety of syntactic (and semantic!) head-argument and head-adjunct dependencies
- ▶ *Smoothing* (i.e., generalizing beyond structures seen in data) is essential (but not well understood)

Outline

Introduction

Stochastic grammars

Supervised learning

Unsupervised learning

Applying this to real data

Factoring learning into simpler components

Conclusion

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 $P(w)$ (but can get trapped in local minima)

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

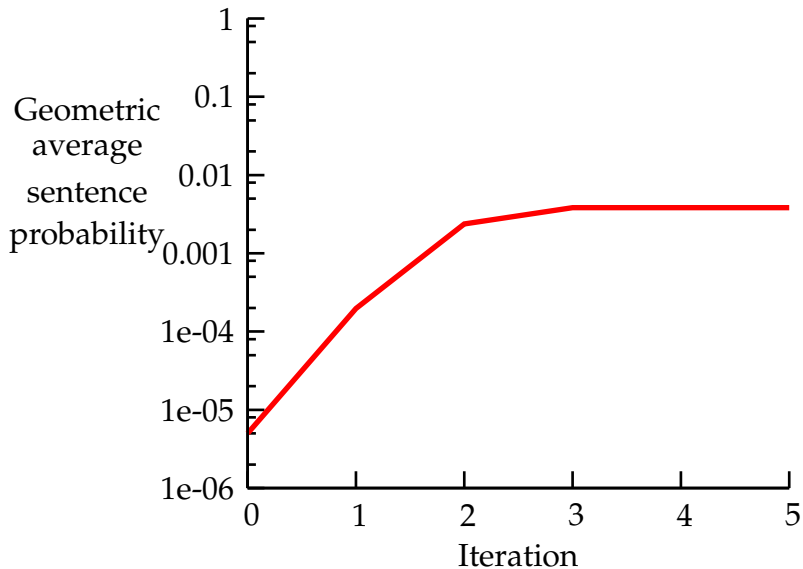
Expectation Maximization with a toy grammar

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

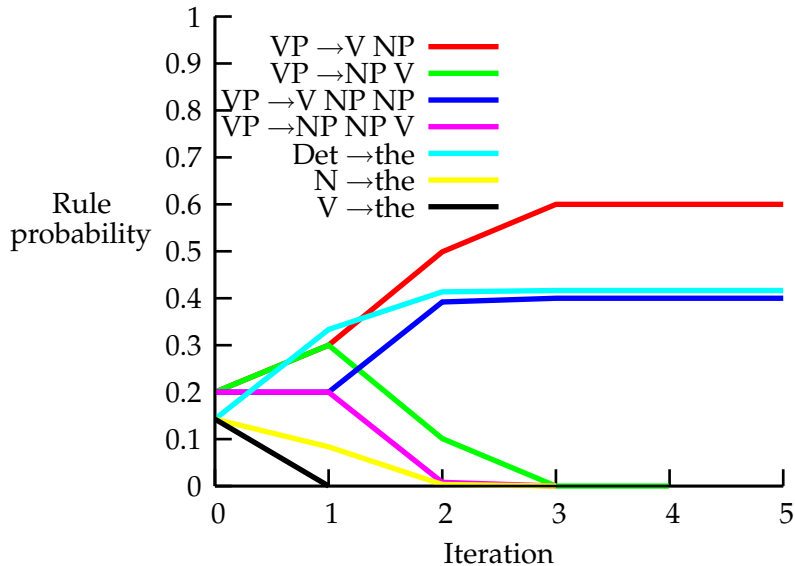
“English” input
the dog bites
the dog bites a man
a man gives the dog a bone
...

“pseudo-Japanese” input
the dog bites
the dog a man bites
a man the dog a bone gives
...

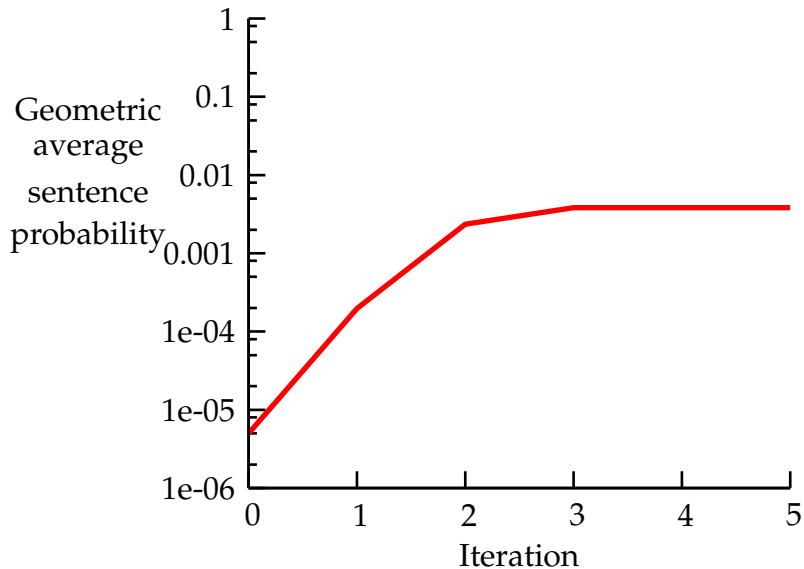
Probability of “English”



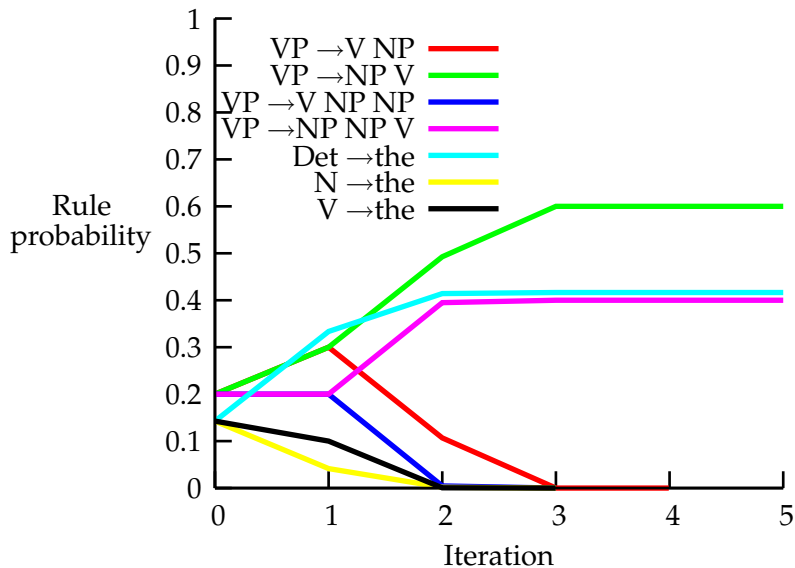
Rule probabilities from “English”



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 \Rightarrow 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
 - ⇒ lower average parse probability
 - ⇒ less clear information about which rules are most useful

Outline

Introduction

Stochastic grammars

Supervised learning

Unsupervised learning

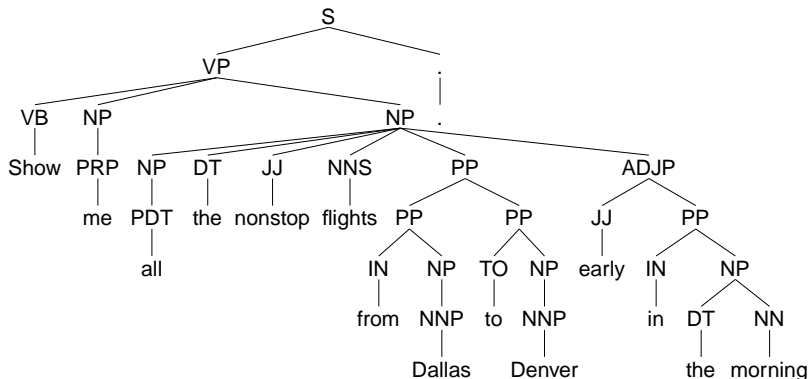
Applying this to real data

Factoring learning into simpler components

Conclusion

Applying EM learning to real language

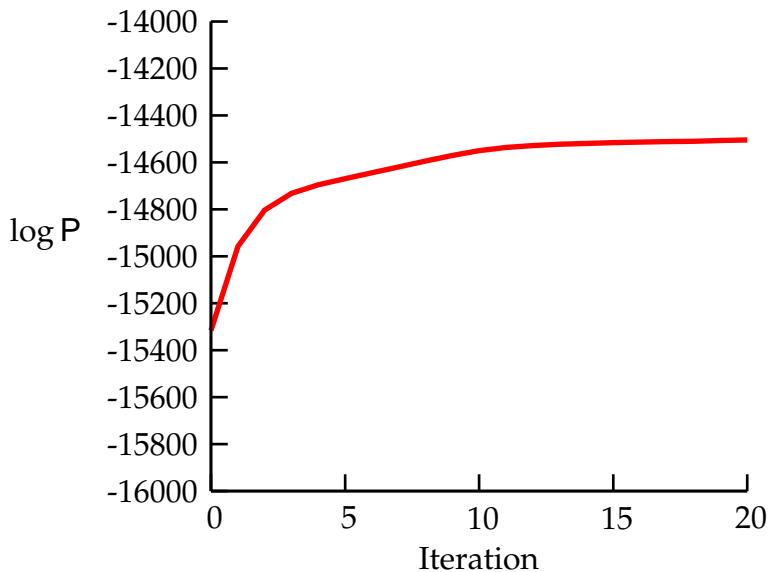
- ▶ ATIS treebank consists of 1,300 hand-constructed parse trees
- ▶ ignore the words (in this experiment)
- ▶ about 1,000 PCFG rules are needed to build these trees



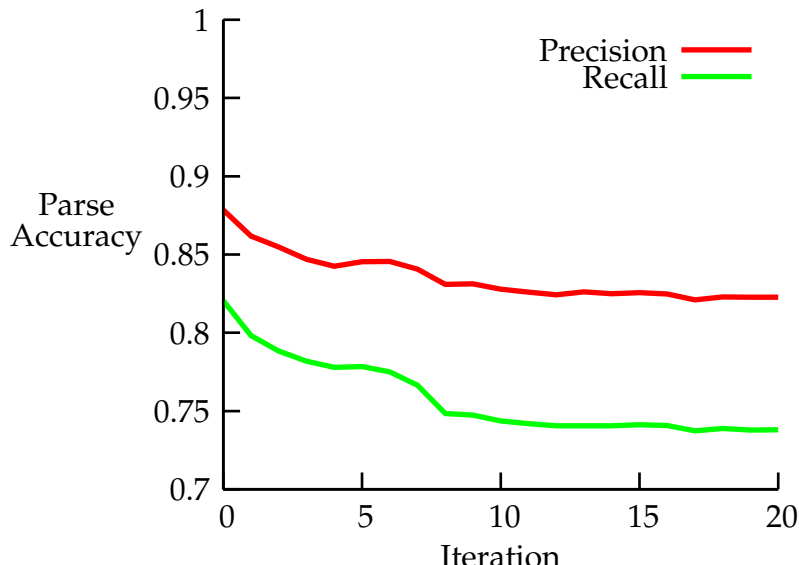
Training from real language

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

Applying this to real data

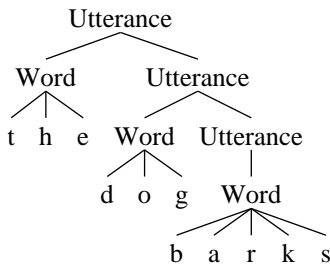
Factoring learning into simpler components

Conclusion

Factoring the language learning problem

- ▶ Factor the language learning problem into linguistically simpler components
- ▶ Focus on components that might be less dependent on context and semantics (e.g., word segmentation, phonology)
- ▶ Identify relevant information sources (including prior knowledge, e.g., UG) by comparing models
- ▶ Combine components to produce more ambitious learners
- ▶ PCFG-like grammars are a natural way to formulate many of these components

Word Segmentation



Data = t h e d o g b a r k s

Utterance \rightarrow Word Utterance

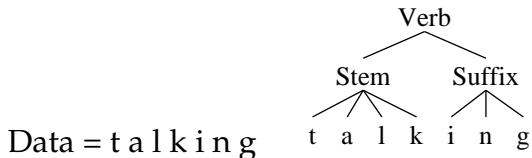
Utterance \rightarrow Word

Word $\rightarrow w$

$w \in \Sigma^*$

- ▶ Algorithms for word segmentation from this information already exists (e.g., Elman, Brent)
- ▶ Likely that children perform some word segmentation before they know the meanings of words

Concatenative morphology



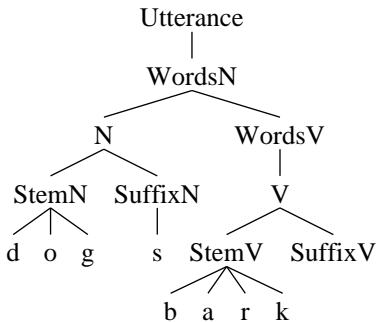
Verb \rightarrow Stem Suffix

Stem $\rightarrow w$ $w \in \Sigma^*$

Suffix $\rightarrow w$ $w \in \Sigma^*$

- ▶ Morphological alternation provides primary evidence for phonological generalizations (“trucks” /s/ vs. “cars” /z/)
- ▶ Morphemes may also provide clues for word segmentation
- ▶ Algorithms for doing this already exist (e.g., Goldsmith)

PCFG components can be integrated



$\text{Utterance} \rightarrow \text{Words}_S \quad S \in \mathcal{S}$

$\text{Words}_S \rightarrow S \text{ Words}_T \quad T \in \mathcal{S}$

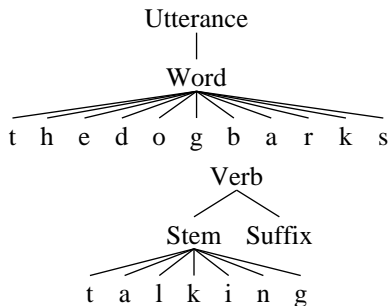
$S \rightarrow \text{Stem}_S \text{ Suffix}_S$

$\text{Stem}_S \rightarrow t \quad t \in \Sigma^*$

$\text{Suffix}_S \rightarrow f \quad f \in \Sigma^*$

Problems with maximum likelihood estimation

- ▶ Maximum likelihood picks model that best fits the data
- ▶ *Saturated models* exactly mimic the training data
⇒ highest likelihood
- ▶ Need a different estimation framework



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)
 - ▶ prefer “simple” or *sparse* grammars
 - ▶ can be over-ridden by sufficient data
- ▶ evaluate *different kinds of universals*

Bayesian estimation

$$\underbrace{P(\text{Hypothesis}|\text{Data})}_{\text{Posterior}} \propto \underbrace{P(\text{Data}|\text{Hypothesis})}_{\text{Likelihood}} \underbrace{P(\text{Hypothesis})}_{\text{Prior}}$$

- ▶ Priors can be sensitive to linguistic structure (e.g., a word should contain a vowel)
- ▶ Priors can encode linguistic universals and markedness preferences (e.g., complex clusters appear at word onsets)
- ▶ Priors can prefer *sparse solutions*
- ▶ The choice of the prior is as much a linguistic issue as the design of the grammar!

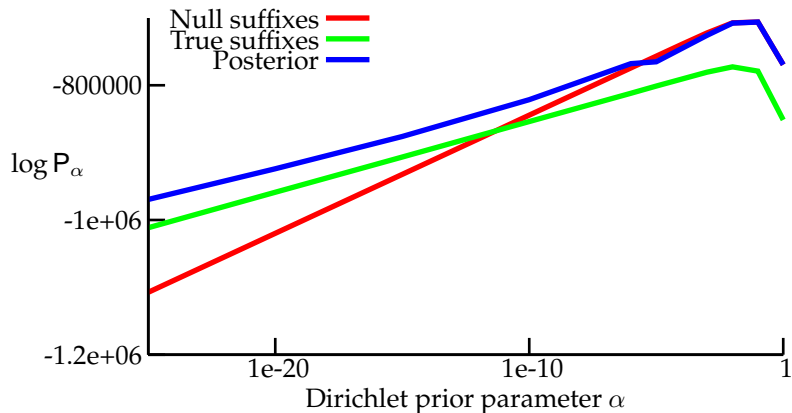
Morphological segmentation experiment

- ▶ Bayesian estimator with Dirichlet prior
 - ▶ prefers sparser solutions as $\alpha \rightarrow 0$
- ▶ Gibbs Sampler used to sample from posterior distribution of parses
 - ▶ reanalyses each word based on a grammar estimated from the parses of the other words
- ▶ Trained on orthographic verbs from U Penn. Wall Street Journal treebank
 - ▶ behaves similarly with broad phonemic child-directed input

Posterior samples from WSJ verb tokens

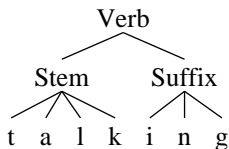
$\alpha = 0.1$	$\alpha = 10^{-5}$	$\alpha = 10^{-10}$	$\alpha = 10^{-15}$
expect	expect	expect	expect
expects	expects	expects	expects
expected	expected	expected	expected
expecting	expect ing	expect ing	expect ing
include	include	include	include
includes	includes	includ es	includ es
included	included	includ ed	includ ed
including	including	including	including
add	add	add	add
adds	adds	adds	add s
added	added	add ed	added
adding	adding	add ing	add ing
continue	continue	continue	continue
continues	continues	continue s	continue s
continued	continued	continu ed	continu ed
continuing	continuing	continu ing	continu ing
report	report	report	report

Log posterior of models on token data



- ▶ Correct solution is nowhere near as likely as posterior
⇒ model is wrong!

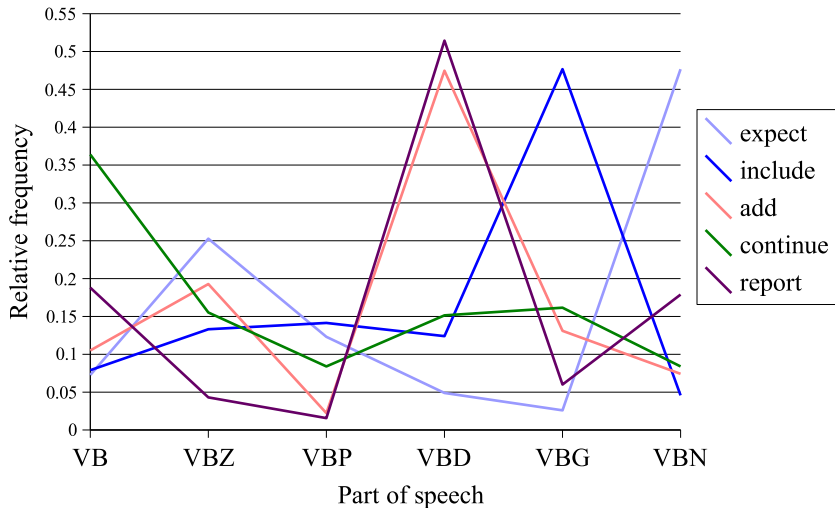
Independence assumption in PCFG model



$$P(\text{Word}) = P(\text{Stem})P(\text{Suffix})$$

- ▶ Model expects relative frequency of each suffix *to be the same for all stems*

Relative frequencies of inflected verb forms



Types and tokens

- ▶ A word *type* is a distinct word shape
- ▶ A word *token* is an occurrence of a word

Data = "the cat chased the other cat"

Tokens = "the" 2, "cat" 2, "chased" 1, "other" 1

Types = "the" 1, "cat" 1, "chased" 1, "other" 1

- ▶ Using word types instead of word tokens effectively normalizes for frequency variations

Posterior samples from WSJ verb *types*

$\alpha = 0.1$	$\alpha = 10^{-5}$	$\alpha = 10^{-10}$	$\alpha = 10^{-15}$
expect	expect	expect	exp ect
expects	expect s	expect s	exp ects
expected	expect ed	expect ed	exp ected
expect ing	expect ing	expect ing	exp ecting
include	includ e	includ e	includ e
include s	includ es	includ es	includ es
included	includ ed	includ ed	includ ed
including	includ ing	includ ing	includ ing
add	add	add	add
adds	add s	add s	add s
add ed	add ed	add ed	add ed
adding	add ing	add ing	add ing
continue	continu e	continu e	continu e
continue s	continu es	continu es	continu es
continu ed	continu ed	continu ed	continu ed
continuing	continu ing	continu ing	continu ing
report	report	repo rt	rep ort

Summary so far

- ▶ Unsupervised learning is difficult on real data!
- ▶ There's a lot to learn from simple problems
 - ▶ need models that require all stems in same class to have same suffixes but permit suffix frequencies to vary with the stem
- ▶ Related problems arise in other linguistic domains as well
 - ▶ Many verbs share the same subcategorization frames, but subcategorization frame frequencies depend on head verb.
- ▶ Hopefully we can combine these simple learners to study their interaction in more complex domains

Outline

Introduction

Stochastic grammars

Supervised learning

Unsupervised learning

Applying this to real data

Factoring learning into simpler components

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

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 probabilistic grammars

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