Language Acquisition as Statistical Inference

Mark Johnson

Joint work with many people, including
Ben Börschinger, Eugene Charniak, Katherine Demuth,
Michael Frank, Sharon Goldwater, Tom Griffiths,
Bevan Jones and Ed Stabler;
thanks to Bob Berwick, Stephen Crain and Mark Steedman
for comments and suggestions

Macquarie University
Sydney, Australia


September 2013
Main claims

• Setting grammatical parameters can be viewed as a *parametric statistical inference* problem
  ▶ e.g., learn *whether* language has verb raising
  ▶ if parameters are *local in the derivation tree* (e.g., lexical entries, including empty functional categories) then there is an efficient parametric statistical for identifying them
  ▶ only requires primary linguistic data contains *positive example sentences*

• In statistical inference usually *parameters have continuous values*, but *is this linguistically reasonable?*
Unsupervised estimation of globally normalised models

- The “standard” modelling dichotomy:
  
  *Generative models:* (e.g., HMMs, PCFGs)
  - locally normalised (rule probs expanding same nonterm sum to 1)
  - unsupervised estimation possible (e.g., EM, samplers, etc.)

  *Discriminative models:* (e.g., CRFs, “MaxEnt” CFGs)
  - globally normalised (feature weights don’t sum to 1)
  - unsupervised estimation generally viewed as impossible

- Claim: *unsupervised estimation of globally-normalised models is computationally feasible* if:
  1. the set of *derivation trees* is *regular* (i.e., generated by a CFG)
  2. all features are *local* (e.g., to a PCFG rule)
Outline

Statistics and probabilistic models

Parameter-setting as parametric statistical inference

An example of syntactic parameter learning

Estimating syntactic parameters using CFGs with Features

Experiments on a larger corpus

Conclusions, and where do we go from here?
Statistical inference and probabilistic models

- A **statistic** is *any function of the data*
  - usually chosen to *summarise* the data
- Statistical inference usually exploits not just the occurrence of phenomena, but also their *frequency*
- **Probabilistic models** predict the frequency of phenomena
  - very useful for statistical inference
    - inference usually involves *setting parameters* to *minimise difference* between model’s expected value of a statistic and its value in data
    - statisticians have shown certain procedures are *optimal* for wide classes of inference problems
- Probabilistic extensions for virtually all theories of grammar
  - *no inherent conflict between grammar and statistical inference*
  - technically, statistical inference can be used under virtually any theory of grammar
    - *but is anything gained by doing so?*
Do “linguistic frequencies” make sense?

- Frequencies of many surface linguistic phenomena vary dramatically with non-linguistic context
  - arguably, word frequencies aren’t part of “knowledge of English”
- Perhaps humans only use robust statistics
  - e.g., closed-class words are often orders of magnitude more frequent than open-class words
  - e.g., the conditional distribution of surface forms given meanings \( P(\text{SurfaceForm} | \text{Meaning}) \) may be almost categorical (Wexler’s “Uniqueness principle”, Clark’s “Principle of Contrast”)

Why exploit frequencies when learning?

- Human learning shows frequency effects
  - usually higher frequency $\Rightarrow$ faster learning
  - $\not\Rightarrow$ statistical learning (e.g., trigger models show frequency effects)
- Frequency statistics provide *potentially valuable information*
  - parameter settings may need updating if *expected frequency is significantly higher than empirical frequency*
  - $\Rightarrow$ avoid “no negative evidence” problems
- Statistical inference seems to work better for many aspects of language than other methods
  - scales up to larger, more realistic data
  - produces more accurate results
  - more robust to noise in the input
Some theoretical results about statistical grammar inference

- **statistical learning can succeed when categorical learning fails** (e.g., PCFGs can be learnt from positive examples alone, but CFGs can’t) (Horning 1969, Gold 1967)
  - statistical learning assumes more about the input (independent and identically-distributed)
  - and has a weaker notion of success (convergence in distribution)
- **learning PCFG parameters from positive examples alone is computationally intractable** (Cohen et al 2012)
  - this is a “worst-case” result, typical problems (or “real” problems) may be easy
  - result probably generalises to Minimalist Grammars (MGs) as well
  ⇒ MG inference algorithm sketched here will run slowly, or will converge to wrong parameter estimates, for some MGs on some data
Parametric and non-parametric inference

• A *parametric model* is one with a finite number of prespecified parameters
  ▶ Principle-and-parameters grammars are parametric models

• *Parametric inference* is inference for the parameter values of a parametric model

• A *non-parametric model* is one which can’t be defined using a bounded number of parameters
  ▶ a lexicon is a non-parametric model if there’s no universal bound on possible lexical entries (e.g., phonological forms)

• *Non-parametric inference* is inference for (some properties of) nonparametric models
Outline

Statistics and probabilistic models

Parameter-setting as parametric statistical inference

An example of syntactic parameter learning

Estimating syntactic parameters using CFGs with Features

Experiments on a larger corpus

Conclusions, and where do we go from here?
Statistical inference for MG parameters

• Claim: there is a *statistical algorithm for inferring parameter values of Minimalist Grammars* (MGs) from positive example sentences alone, assuming:
  ▶ MGs are efficiently parsable
  ▶ MG *derivations* (not parses!) have a *context-free structure*
  ▶ parameters are associated with *subtree-local configurations* in derivations (e.g., lexical entries)
  ▶ a probabilistic version of MG with *real-valued parameters*

• Example: learning verb-raising parameters from toy data
  ▶ e.g., learn language has V>T movement from examples like *Sam sees often Sasha*
  ▶ truth in advertising: this example uses an equivalent CFG instead of an MG to generate derivations

• *Not tabula rasa learning*: we estimate parameter values (e.g., that a language has V>T movement); the possible parameters and their linguistic implications are prespecified (e.g., innate)
Outline of the algorithm

- Use a “MaxEnt” probabilistic version of MGs
- Although MG derived structures are not context-free (because of movement) they have context-free derivation trees (Stabler and Keenan 2003)
- Parametric variation is subtree-local in derivation tree (Chiang 2004)
  - e.g., availability of specific empty functional categories triggers different movements

⇒ The partition function can be efficiently calculated (Hunter and Dyer 2013)
⇒ Standard “hill-climbing” methods for context-free grammar parameter estimation generalise to MGs
Maximum likelihood statistical inference procedures

- If we have:
  - a probabilistic model $P$ that depends on parameter values $w$, and
  - data $D$ we want to use to infer $w$

the **Principle of Maximum Likelihood** is: *select the $w$ that makes the probability of the data $P(D)$ as large as possible*

- Maximum likelihood inference is *asymptotically optimal* in several ways

- Maximising likelihood is an *optimisation problem*

- *Calculating $P(D)$* (or something related to it) is necessary
  - need the *derivative of the partition function* for hill-climbing search
Maximum Likelihood and the Subset Principle

- The Maximum Likelihood Principle entails a probabilistic version of the Subset Principle (Berwick 1985)
- Maximum Likelihood Principle: select parameter weights $w$ to make the probability of data $P(D)$ as large as possible
- $P(D)$ is the *product* of the probabilities of the sentences in $D$
  - $\Rightarrow w$ assigns each sentence in $D$ relatively large probability
  - $\Rightarrow w$ generates at least the sentences in $D$
- Probabilities of all sentences must *sum to 1*
  - $\Rightarrow$ can assign higher probability to sentences in $D$ if $w$ generates fewer sentences outside of $D$
    - e.g., if $w$ generates 100 sentences, then each can have prob. 0.01
      if $w$ generates 1,000 sentences, then each can have prob. 0.001
  - $\Rightarrow$ Maximum likelihood estimation selects $w$ so sentences in $D$ have high prob., and few sentences not in $D$ have high prob.
The utility of continuous-valued parameters

- Standardly, linguistic parameters are discrete (e.g., Boolean)
- Most statistical inference procedures use continuous parameters
- In the models presented here, parameters and lexical entries are associated with real-valued weights
  - E.g., if $w_{V \rightarrow T} \ll 0$ then a derivation containing V-to-T movement will be much less likely than one that does not
  - E.g., if $w_{\text{will}:V} \ll 0$ then a derivation containing the word will with syntactic category V will be much less likely
- Continuous parameter values and probability models:
  - are a continuous relaxation of discrete parameter space
  - define a gradient that enables incremental “hill climbing” search
  - can represent partial or incomplete knowledge with intermediate values (e.g., when learner isn’t sure)
  - but also might allow “zombie” parameter settings that don’t correspond to possible human languages
Derivations in Minimalist Grammars

- Grammar has two fundamental operations: *external merge* (head-complement combination) and *internal merge* (movement).
- Both operations are driven by feature checking:
  - derivation terminates when all formal features have been checked or cancelled.
- MG as formalised by Stabler and Keenan (2003):
  - the *string and derived tree languages* MGs generate are *not context-free*, but
  - MG derivations are specified by a *derivation tree*, which abstracts over surface order to reflect the structure of internal and external merges, and
  - the *possible derivation trees* have a *context-free structure* (c.f. TAG).
which wine the queen prefers
Example MG derivation tree

\[ \varepsilon ::= V + \text{wh} C \]

\[ \cdot = D \ V \]

\[ \cdot = D \ D \ - \text{wh} \]

\[ \text{prefers} ::= D = D \ V \]

\[ \cdot = N \ D \]

\[ \text{the} ::= N \ D \]

\[ \cdot = N \ D \ - \text{wh} \]

\[ \text{which} ::= N \ D \ - \text{wh} \]

\[ \text{wine} ::= N \]

which wine the queen prefers
Calculating the probability $P(D)$ of data $D$

- If data $D$ is a sequence of independently generated sentences $D = (s_1, \ldots, s_n)$, then:

  $$P(D) = P(s_1) \times \ldots \times P(s_n)$$

- If a sentence $s$ is ambiguous with derivations $\tau_1, \ldots, \tau_m$ then:

  $$P(s) = P(\tau_1) + \ldots + P(\tau_m)$$

- These are standard formal language theory assumptions
  - which does not mean they are correct!
  - Luong et al (2013) shows learning can improve by modeling dependencies between $s_i$ and $s_{i+1}$

- Key issue: how do we define the probability $P(\tau)$ of derivation $\tau$?

- If $s$ is very ambiguous (as is typical during learning), need to calculate $P(s)$ without enumerating all its derivations
For Maximum Likelihood inference we need to calculate the MG derivations of the sentences in the training data $D$.

Stabler (2012) describes several algorithms for parsing with MGs:

- MGs can be translated to equivalent Multiple CFGs (MCFGs).
- While MCFGs are strictly more expressive than CFGs, for any given sentence there is a CFG that generates an equivalent set of parses (Ljunglöf 2012).

$\Rightarrow$ CFG methods for “efficient” parsing (Lari and Young 1990) should generalise to MGs.
MaxEnt probability distributions on MG derivations

- Associate each parameter $\pi$ with a function from derivations $\tau$ to the number of times some configuration appears in $\tau$
  - e.g., $+\text{wh}(\tau)$ is the number of WH-movements in $\tau$
  - same as *constraints* in Optimality Theory
- Each parameter $\pi$ has a *real-valued weight* $w_\pi$
- The probability $P(\tau)$ of derivation $\tau$ is:
  \[ P(\tau) = \frac{1}{Z} \exp \left( \sum_\pi w_\pi \pi(\tau) \right) \]
  where $\pi(\tau)$ is the number of times the configuration $\pi$ occurs in $\tau$
- $w_\pi$ generalises a conventional binary parameter value:
  - if $w_\pi > 0$ then each occurrence of $\pi$ *increases* $P(\tau)$
  - if $w_\pi < 0$ then each occurrence of $\pi$ *decreases* $P(\tau)$
- Essentially the same as Abney (1996) and Harmonic Grammar (Smolensky et al 1993)
The importance of the partition function $Z$

- Probability $P(\tau)$ of a derivation $\tau$:

$$P(\tau) = \frac{1}{Z} \exp\left(\sum_{\pi} w_{\pi} \pi(\tau)\right)$$

- The *partition function* $Z$ is crucial for statistical inference
  - inference algorithms for learning $w_{\pi}$ without $Z$ are more heuristic

- Calculating $Z$ naively involves *summing over all possible derivations of all possible strings*, but this is usually *infeasable*

- But if the possible derivations $\tau$ have a context-free structure and the $\pi$ configurations are “local”, it is *possible to calculate $Z$ without exhaustive enumeration*
Calculating the partition function $Z$ for MGs

- Hunter and Dyer (2013) and Chiang (2004) observe that the partition function $Z$ for MGs can be efficiently calculated generalising the techniques of Nederhof and Satta (2008) if:
  - the parameters $\pi$ are functions of local subtrees of the derivation tree $\tau$, and
  - the possible MG derivations have a context-free structure

- Stabler (2012) suggests that empty functional categories control parametric variation in MGs
  - e.g., if lexicon contains “$\varepsilon::=V +wh C$” then language has WH-movement
  - the number of occurrences of each empty functional category is a function of local subtrees

⇒ If we define a parameter $\pi_\lambda$ for each lexical entry $\lambda$ where:
  - $\pi_\lambda(\tau) =$ number of times $\lambda$ occurs in derivation $\tau$
  - then the partition function $Z$ can be efficiently calculated.
Outline

Statistics and probabilistic models

Parameter-setting as parametric statistical inference

An example of syntactic parameter learning

Estimating syntactic parameters using CFGs with Features

Experiments on a larger corpus

Conclusions, and where do we go from here?
A “toy” example

- Involves verb movement and inversion (Pollock 1989)
- 3 different sets of 25–40 input sentences
  - (“English”) Sam often sees Sasha, Q will Sam see Sasha, . . .
  - (“French”) Sam sees often Sasha, Sam will often see Sasha, . . .
  - (“German”) Sees Sam often Sasha, Will Sam Sasha see, . . .
- Syntactic parameters: $V \succ T$, $T \succ C$, $T \succ Q$, $XP \succ SpecCP$, $V_{\text{init}}$, $V_{\text{fin}}$
- Lexical parameters associating all words with all categories (e.g., $\text{will}:I$, $\text{will}:Vi$, $\text{will}:Vt$, $\text{will}:D$)
- Hand-written CFG instead of MG; parameters associated with CF rules rather than empty categories (Chiang 2004)
  - grammar inspired by MG analyses
  - *calculates same parameter functions* $\pi$ as MG would
  - could use a MG parser if one were available
“English”: no V-to-T movement

Jean has often seen Paul

Jean e often sees Paul
“French” : V-to-T movement

Jean

\[
\text{TP} \quad \begin{array}{c}
\text{DP} \\
\text{a} \\
\text{AP} \\
\text{souvent} \\
\text{VP} \\
\text{V} \\
\text{DP} \\
\text{Paul}
\end{array}
\]

\[
\text{voit} \\
\text{AP} \\
\text{souvent} \\
\text{VP} \\
\text{V} \\
\text{DP} \\
\text{t} \\
\text{Paul}
\]

Paul

Jean

\[
\text{TP} \quad \begin{array}{c}
\text{DP} \\
\text{T'} \\
\text{T} \\
\text{VP} \\
\text{a} \\
\text{AP} \\
\text{souvent} \\
\text{VP} \\
\text{V} \\
\text{DP} \\
\text{Paul}
\end{array}
\]

\[
\text{voit} \\
\text{AP} \\
\text{souvent} \\
\text{VP} \\
\text{V} \\
\text{DP} \\
\text{t} \\
\text{Paul}
\]
“English”: T-to-C movement in questions
“French”: T-to-C movement in questions

```
CP       CP
  C'       C'
  C       C'
     TP     TP
     DP     DP  T'
   vous    vous
   vous    vous
      T    T
     VP    VP
    t     t
   AP    AP
  souvent souvent
     V     V
   DP    DP
   Paul  Paul
```

29/58
“German”: V-to-T and T-to-C movement

```
CP  
C'  
C   T'  
    TP  
      DP  
        Jean  
          VP  
            T
              DP  
                V  
                  hat
                    DP  
                      V  
                        Paul
gesehen
```

```
CP  
C'  
C   T'  
    TP  
      DP  
        Jean  
          VP  
            T
              DP  
                V  
                  hat
                    DP  
                      V  
                        Paul
gesehen
```

```
CP  
C'  
C   T'  
    TP  
      DP  
        Jean  
          VP  
            T
              DP  
                V  
                  t
                    DP  
                      V  
                        Paul
t
```
“German”: V-to-T, T-to-C and XP-to-SpecCP movement

Jean hat Paul gesehen

Paul schlägt häufig t

häufig sah Jean Paul t
Input to parameter inference procedure

- A CFG designed to mimic MG derivations, with parameters associated with rules
- 25–40 sentences, such as:
  ▶ (“English”) Sam often sees Sasha, Q will Sam see Sasha
  ▶ (“French”) Sam sees often Sasha, Q see Sam Sasha
  ▶ (“German”) Sam sees Sasha, sees Sam Sasha, will Sam Sasha see

- Identifying parameter values is easy if we know lexical categories
- Identifying lexical entries is easy if we know parameter values
- Learning both jointly faces a “chicken-and-egg” problem
Algorithm for statistical parameter estimation

- Parameter estimation algorithm:
  - Initialise parameter weights somehow
  - Repeat until converged:
    - calculate likelihood and its derivatives
    - update parameter weights to increase likelihood
- Very simple parameter weights updates suffice
- Computationally most complex part of procedure is parsing the data to calculate likelihood and its derivatives
  ⇒ learning is a by-product of parsing
- Straight-forward to develop incremental on-line versions of this algorithm (e.g., stochastic gradient ascent)
  ▶ an advantage of explicit probabilistic models is that there are standard techniques for developing algorithms with various properties
Outline

Statistics and probabilistic models

Parameter-setting as parametric statistical inference

An example of syntactic parameter learning

**Estimating syntactic parameters using CFGs with Features**

Experiments on a larger corpus

Conclusions, and where do we go from here?
Context-free grammars with Features

- A **Context-Free Grammar with Features** (CFGF) is a “MaxEnt CFG” in which **features are local to local trees** (Chiang 2004), i.e.:
  - each rule $r$ is assigned feature values $f(r) = (f_1(r), \ldots, f_m(r))$
    - $f_i(r)$ is count of $i$th feature on $r$ (normally 0 or 1)
  - features are associated with weights $w = (w_1, \ldots, w_m)$
- The feature values of a tree $t$ are the sum of the feature values of the rules $R(t) = (r_1, \ldots, r_\ell)$ that generate it:
  \[
  f(t) = \sum_{r \in R(t)} f(r)
  \]
- A CFGF assigns probability $P(t)$ to a tree $t$:
  \[
  P(t) = \frac{1}{Z} \exp(w \cdot f(t)), \text{ where: } Z = \sum_{t' \in T} \exp(w \cdot f(t'))
  \]
  and $T$ is the set of *all parses for all strings* generated by grammar
Log likelihood and its derivatives

- Minimise *negative log likelihood* plus a Gaussian regulariser
  - Gaussian mean $\mu = -1$, variance $\sigma^2 = 10$
- Derivative of log likelihood requires *derivative of log partition function* $\log Z$

$$\frac{\partial \log Z}{\partial w_j} = \mathbb{E}[f_j]$$

where expectation is calculated over $\mathcal{T}$ (set of *all parses for all strings* generated by grammar)

CFGF used here

CP → C' \; \sim Q \sim XP > SpecCP
CP → DP C'/DP; \sim Q XP > SpecCP
C' → TP; \sim T > C
C'/DP → TP/DP; \sim T > C
C'/DP → T TP/T, DP; T > C
C' → Vi TP/Vi; V > T T > C

- Parser does not handle epsilon rules ⇒ manually “compiled out”
- 24-40 sentences, 44 features, 116 rules, 40 nonterminals, 12 terminals
  > while every CFGF distribution can be generated by a PCFG with
  the same rules (Chi 1999), it is differently parameterised (Hunter
  and Dyer 2013)
Sample trees generated by CFGF

- **Sam**
  - **TP**
    - **DP**
      - **VP**
        - **AP**
          - **VP**
            - **often**
              - **V’**
                - **V**
                  - **eats**
                    - **fish**
            - **DP**

- **you**
  - **CP**
    - **C’**
      - **Vt**
        - **ad**
          - **TP/Vt**
            - **T’/Vt**
              - **vous**
                - **VP/Vt**
                  - **AP**
                    - **VP/Vt**
                      - **souvent**
                        - **DP**
                          - **Paul**

- **Jean**
  - **CP**
    - **AP**
      - **C’/AP**
        - **Vi**
          - **schläft**
            - **DP**

Estimated parameter value for English, French, and German languages.
Estimated parameter value

- $T > C$
- $T > C_Q$
- $\neg XP > SpecCP$
- $\neg T > C$
- $\neg T > C_Q$
- $XP > SpecCP$

Languages:
- English
- French
- German
Lexical parameters for English

Estimated parameter value

Sam  will  often  see  sleep

D  T  A  Vt  Vi
Learning English parameters

![Graph of parameter values versus gradient-ascent iterations for different words and actions like will:Vt, will:Vi, will:T, will:DP, will:AP, Sam:Vt, Sam:Vi, Sam:T, Sam:DP, Sam:AP, see:Vt, see:Vi, see:T, see:DP, see:AP, and sleep:Vt.](image)
Learning English lexical and syntactic parameters

Parameter value vs. Gradient-ascent iterations

- Sam:DP
- will:T
- often:AP
- ~XP>SpecCP
- ~V>T
- ~T>C
- T>Q
- Vinitial
Learning “often” in English

![Graph showing the learning process of “often” in English with gradient-ascent iterations and parameter values. The graph plots parameter value against gradient-ascent iterations. The parameters include often:Vt, often:Vi, often:T, often:DP, and often:AP. The graph shows how these parameters change over iterations.]
Relation to other work

- Many other “toy” parameter-learning systems:
  - E.g., Yang (2002) describes an error-driven learner with templates triggering parameter value updates
  - we jointly learn lexical categories and syntactic parameters

- Error-driven learners like Yang’s can be viewed as an approximation to the algorithm proposed here:
  - on-line error-driven parameter updates are a stochastic approximation to gradient-based hill-climbing
  - MG parsing is approximated with template matching
Relation to Harmonic Grammar and Optimality Theory

- Harmonic Grammars are MaxEnt models that associate weights with configurations much as we do here (Smolensky et al 1993)
  - because no constraints are placed on possible parameters or derivations, little detail about computation for parameter estimation
- Optimality Theory can be viewed as a discretised version of Harmonic Grammar in which *all parameter weights must be negative*
- MaxEnt models like these are widely used in phonology (Goldwater and Johnson 2003, Hayes and Wilson 2008)
Outline

Statistics and probabilistic models

Parameter-setting as parametric statistical inference

An example of syntactic parameter learning

Estimating syntactic parameters using CFGs with Features

Experiments on a larger corpus

Conclusions, and where do we go from here?
Unsupervised parsing on WSJ10

- Input: POS tag sequences of all sentences of length 10 or less in WSJ PTB.
- $X'$-style grammar coded as a CFG

\[
\begin{align*}
XP & \rightarrow YP XP & XP & \rightarrow XP YP \\
XP & \rightarrow YP X' & XP & \rightarrow X' YP \\
XP & \rightarrow X' & X' & \rightarrow YP X' \\
X' & \rightarrow YP X' & X' & \rightarrow X' YP \\
X' & \rightarrow YP X & X' & \rightarrow X YP \\
X' & \rightarrow X & X' & \rightarrow X YP
\end{align*}
\]

where $X$ and $Y$ range over all 45 Parts of Speech (POS) in corpus

- 9,975 CFG rules in grammar
- PCFG estimation procedures (e.g., EM) do badly on this task (Klein and Manning 2004)
Example parse tree generated by XP grammar

- Evaluate by *unlabelled* precision and recall wrt standard treebank parses
2 grammars, 4 different parameterisations

1. **XP grammar**: a PCFG with 9,975 rules  
   ▶ estimated using Variational Bayes with Dirichlet prior ($\alpha = 0.1$)

2. **DS grammar**: a CFG designed by Noah Smith to capture approximately the same generalisations as DMV model  
   ▶ 5,250 CFG rules  
   ▶ also estimated using Variational Bayes with Dirichlet prior

3. **XPF0 grammar**: same rules as XP grammar, but one feature per rule  
   ▶ estimated by maximum likelihood with L2 regulariser ($\sigma = 1$)  
   ▶ same expressive power as XP grammar

4. **XPF1 grammar**: same rules as XP grammar, but multiple features per rule  
   ▶ 12,095 features in grammar  
   ▶ extra parameters shared across rules for e.g., head direction, etc., which *couple probabilities of rules*  
   ▶ estimated by maximum likelihood with L2 regulariser ($\sigma = 1$)  
   ▶ same expressive power as XP grammar
Experimental results

- Each estimator initialised from 100 different random starting points
- XP PCFG does badly (as Klein and Manning describe)
- XPF0 grammar does as well or better than Smith’s specialised DS grammar
- Adding additional coupling factors in XP1 grammar reduce variance in estimated grammar
Outline

Statistics and probabilistic models

Parameter-setting as parametric statistical inference

An example of syntactic parameter learning

Estimating syntactic parameters using CFGs with Features

Experiments on a larger corpus

Conclusions, and where do we go from here?
Statistical inference for syntactic parameters

- No inherent contradiction between probabilistic models, statistical inference and grammars
- Statistical inference can be used to set real-valued parameters (learn empty functional categories) in Minimalist Grammars (MGs)
  - parameters are local in context-free derivation structures
    ⇒ efficient computation
  - can solve “chicken-and-egg” learning problems
  - does not need negative evidence
- Not a *tabula rasa* learner
  - depends on a rich inventory of prespecified parameters
Technical challenges in syntactic parameter estimation

- The partition function $Z$ can become unbounded during estimation
  - modify search procedure (for our cases, optimal grammar always has finite $Z$)
  - use an alternative EM-based training procedure?
- Difficult to write linguistically-interesting CFGFs
  - epsilon-removal grammar transform would permit grammars with empty categories
  - MG-to-CFG compiler?
Future directions in syntactic parameter acquisition

- **Are real-valued parameters linguistically reasonable?**
- Does approach “scale up” to realistic grammars and corpora?
  - parsing and inference components use efficient dynamic programming algorithms
  - many informal proposals, but no “universal” MGs (perhaps start with well-understood families like Romance?)
  - generally disappointing results scaling up PCFGs (de Marken 1995)
  - but our grammars lack so much (e.g., LF movement, binding)
- Exploit semantic information in the non-linguistic context
  - e.g., learn from surface forms paired with their logical form semantics (Kwiatkowski et al 2012)
  - but what information does child extract from non-linguistic context?
- Use a nonparametric Bayesian model to learn the empty functional categories of a language (c.f., Bisk and Hockenmaier 2013)
Why probabilistic models?

- Probabilistic models are a *computational level* description
  - they define the relevant variables and dependencies between them
- Models are stated at a *higher level of abstraction* than algorithms:
  - easier to see how to incorporate additional dependencies (e.g., non-linguistic context)
- There are standard ways of constructing inference algorithms for probabilistic models:
  - usually multiple algorithms for same model with different properties (e.g., incremental, on-line)
- My opinion: *it’s premature to focus on algorithms*
  - identify relevant variables and their dependencies first!
  - *optimal inference procedures* let us explore consequences of a model *without committing to any particular algorithm*
How might statistics change linguistics?

• Few examples where probabilistic models/statistical inference provides crucial insights
  ▶ role of negative evidence in learning
  ▶ statistical inference compatible with conventional parameter setting

• Non-parametric inference can learn which parameters are relevant
  ▶ needs a generative model or “grammar” of possible parameters
  ▶ but probability theory is generally agnostic as to parameters

• Probabilistic models have more relevance to psycholinguistics and language acquisition
  ▶ these are computational processes
  ▶ explicit computational models can make predictions about the time course of these processes
This research was supported by Australian Research Council Discovery Projects DP110102506 and DP110102593.


Interested in computational linguistics and its relationship to linguistics, language acquisition or neurolinguistics? **We’re recruiting PhD students!** Contact me or anyone from Macquarie University for more information.