Hierarchical Bayesian Models for Modeling Cognitive Processes

Çağrı Çöltekin

Center for Language and Cognition
University of Groningen
c.coltekin@rug.nl

March 11, 2009

Overview

Two views on statistics

- **Frequentist view**
  - Probabilities are long-run frequencies.
  - We can talk about probabilities only for well defined experiments with random outcomes.
  - Emphasis is on *objectivity*: data must speak for itself.
- **Bayesian view**
  - Probabilities are degrees of belief.
  - Probabilities can be assigned to any statement.
  - Inference is *subjective* (methods for somewhat objective inference exists)

Statistical Inference

- We collect a sample $\mathbf{X}$ from a population.
- We know (or assume) that $\mathbf{X}$ is according to a known distribution that can be parametrized by θ

$$p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)p(\theta)}{p(\mathbf{X})}$$

$p(\theta|\mathbf{X})$: *posterior*
$p(\mathbf{X}|\theta)$: *likelihood* $(\mathcal{L}(\theta))$
$p(\theta)$: *prior*
$p(\mathbf{X})$: Marginal probability of data $(\int p(\mathbf{X}|\theta)p(\theta)d\theta)$

Statistical Inference: the two approaches

$$p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)p(\theta)}{p(\mathbf{X})}$$
$$\propto p(\mathbf{X}|\theta)p(\theta)$$

- Frequentist approach:
  - θ is a fixed but unknown.
  - Inference is done via *Maximum Likelihood Estimate* (MLE). Prior information is never used.
  - We need to assess the reliability of the estimate by significance tests (p values, confidence intervals).
- Bayesian approach:
  - θ is is treated just like any other random variable.
  - Posterior contains all the necessary ingredients for the inference.

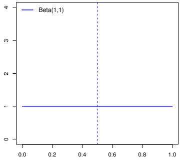
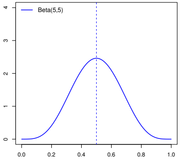
A simple example

We toss a coin 10 times, the outcome is $HHTHHHHTTH$. Let θ represent the chance that coin comes up 'H'.

Frequentist	Bayesian 1	Bayesian 2

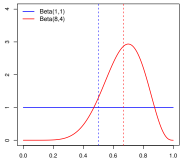
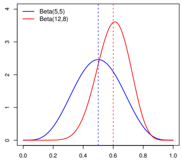
A simple example

We toss a coin 10 times, the outcome is $HHTHHHHTTH$. Let θ represent the chance that coin comes up 'H'.

Frequentist	Bayesian 1	Bayesian 2
No prior	Prior: 	Prior: 

A simple example

We toss a coin 10 times, the outcome is *HHTHHHHTTH*. Let θ represent the chance that coin comes up '*H*'.

Frequentist	Bayesian 1	Bayesian 2
$\widehat{\theta} = \underset{\theta}{\text{argmax}}\ \mathcal{L}(\theta)$ $\widehat{\theta} = 0.7$ $p = 0.2059$	Posterior: 	Posterior: 

Bayesian Inference: summary

$$p(\theta|\mathbf{X}) \propto p(\mathbf{X}|\theta)p(\theta)$$

Pros

- Allows complete inference: posterior distribution contains all the information needed.
- Interpretation of the results are straightforward.

Cons

- Computationally expensive: calculation of posteriors are not always easy.
- Choice of priors: it is sometimes difficult to justify the use of (subjective) priors.

Statistical Inference and Learning

- Statistical inference aims to draw conclusions about an underlying population using a sample drawn from this population.
- Learning can be viewed as making generalizations about the target concept by looking at a sample of data consistent with this concept.
- Statistical methods are most common learning methods in machine learning.
- More and more psychological phenomena is explained by sensitivity to statistical information in the environment.
- Language acquisition is no exception: children are known to exploit statistical regularities in the input in language learning.

(non-Bayesian) Statistical Learning

$$
\begin{aligned}
p(\theta|\mathbf{X}) &= \frac{p(\mathbf{X}|\theta)p(\theta)}{p(\mathbf{X})} \\
&\propto p(\mathbf{X}|\theta)p(\theta)
\end{aligned}
$$

We want to learn the parameter θ.

- Maximum Likelihood Estimate(MLE):

$$\widehat{\theta} = \underset{\theta}{\operatorname{argmax}}\ p(\mathbf{X}|\theta)$$

- Maximum a posteriori (MAP) estimate:

$$\widehat{\theta} = \underset{\theta}{\operatorname{argmax}}\ p(\mathbf{X}|\theta)p(\theta)$$

Bayesian Learning

$$p(\theta|\mathbf{X}) \propto p(\mathbf{X}|\theta)p(\theta)$$

- No point estimates.
- No maximization.
- A Bayesian learner learns the posterior distribution, $p(\theta|\mathbf{X})$.
- If needed, point estimates can be made using,

$$E[\theta] = \int \theta p(\theta|\mathbf{X}) d\theta$$

Note the difference from MAP estimate.

Bayesian Learning: how?

$$p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)p(\theta)}{\int p(\mathbf{X}|\theta)p(\theta)d\theta}$$

- Given the probability density (or distribution) functions for prior and likelihood, we simply multiply, and normalize.
- Note that likelihood and prior does not have to be *proper*. Multiplication by a constant does not change the results.

Problem 1: computation

The computation involved is not always easy to carry out. Approximate methods, such as MCMC, are frequently used.

Problem 2: priors

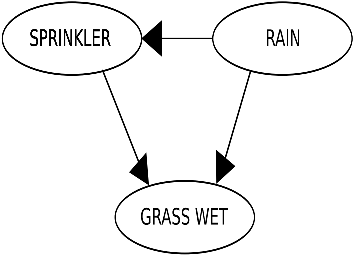
We need to choose a prior distribution.

Choice of priors

- Subjective priors: based on previous experience.
- Non-informative priors: try to be as objective as possible.
- Conjugate priors: for computational efficiency.
- Empirical Bayes: priors from data.
- Hierarchical priors: combining information from different variables.

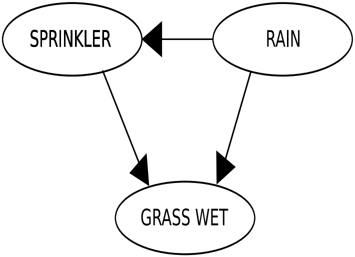
Digression: Graphical models (or 'Bayesian networks')

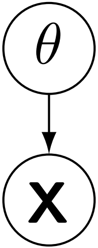
- Often multiple random variables interact.
- Bayesian networks are a convenient way to visualize the dependency relations between variables.



Digression: Graphical models (or 'Bayesian networks')

- Often multiple random variables interact.
- Bayesian networks are a convenient way to visualize the dependency relations between variables.





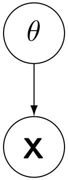
Hierarchical Bayesian Models

In our coin toss example we choose a *Beta*(α, β) prior with fixed α and β.

$$p(\theta|\mathbf{X}) \propto p(\mathbf{X}|\theta)p(\theta)$$

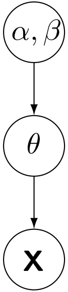
where,

$$\mathbf{X} \sim Binomial(\theta),\ \theta \sim Beta(\alpha, \beta)$$



Hierarchical Bayesian Models

In our coin toss example we choose a $Beta(\alpha, \beta)$ prior with fixed α and β.



$$p(\theta|\mathbf{X}) \propto p(\mathbf{X}|\theta)p(\theta)$$

where,

$$\mathbf{X} \sim Binomial(\theta),\ \theta \sim Beta(\alpha, \beta)$$

We can further extend this if choice of α and β can be guided by additional information.

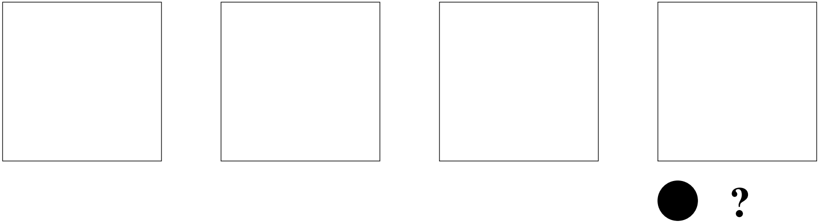
$$p(\theta, \alpha, \beta|\mathbf{X}) \propto p(\mathbf{X}|\theta)p(\theta|\alpha, \beta)p(\alpha, \beta)$$

where, e.g.,

$$\mathbf{X} \sim Binomial(\theta), \theta \sim Beta(\alpha, \beta), \alpha \sim N(\mu_\alpha, \sigma), \beta \sim N(\mu_\beta, \sigma)$$

A simple example: marbles in boxes

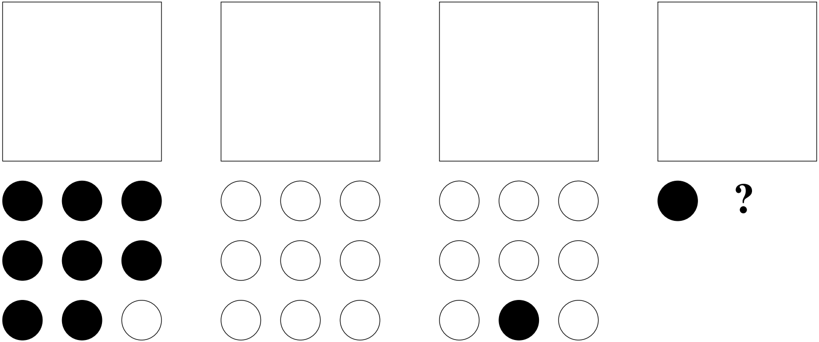
Boxes contain either white or black marbles:



* This example is adopted from a talk by J. Tenenbaum

A simple example: marbles in boxes

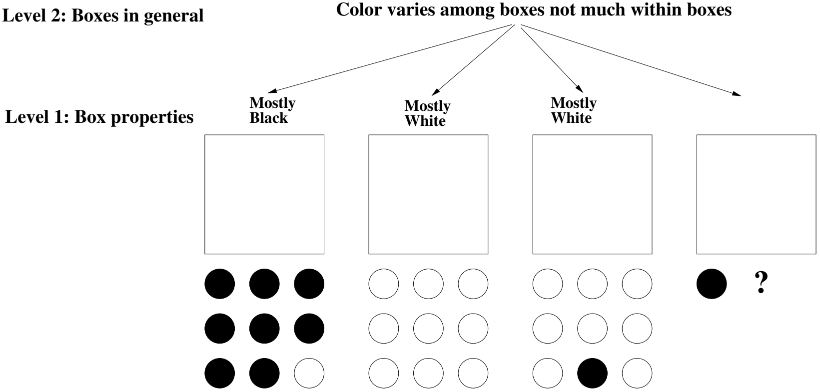
Boxes contain either white or black marbles:



* This example is adopted from a talk by J. Tenenbaum

A simple example: marbles in boxes

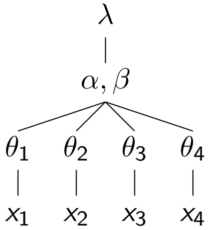
Boxes contain either white or black marbles:



* This example is adopted from a talk by J. Tenenbaum

HBM for marbles in boxes

- x_i are the data
- θ_i models box proportions, $x_i \sim \mathit{Binomial}(\theta)$
- α, β models the boxes in general, $\theta_i \sim \mathit{Beta}(\alpha, \beta)$
- λ models prior expectations in boxes in general



$$p(\lambda, \alpha, \beta, \theta \mid \mathbf{X}) \propto p(\mathbf{X} \mid \theta) p(\theta \mid \lambda, \alpha, \beta) p(\alpha, \beta \mid \lambda) p(\lambda)$$

Summary: HBMs for Modeling Human Learning

- Bayesian Statistics provides complete inference: posterior distribution contains all we need.
- Bayesian Learning is incremental: posterior can be used as prior for the next step.
- Hierarchical models allow a way to include information from different sources as prior knowledge.

Overview

Categorial Grammars

- Categorial Grammar (CG) encodes all the language specific syntactic information in the lexicon.
- CG Lexicon contains *lexical items* of the form:

$$\phi := \sigma : \mu$$

  where ϕ is the *phonological form*, σ is the syntactic category, and μ is the meaning of the lexical item.
- Syntactic categories in CG are,
  - either a basic category, such as **N**, **NP**, **S**
  - or, a complex category of the form $\mathbf{X}\backslash\mathbf{Y}$ or **X/Y**, where **X** and **Y** are any (basic or complex) CG categories. Informally:
    X/Y says: 'I need a **Y** to my *right* to become **X**'
    $\mathbf{X}\backslash\mathbf{Y}$ says: 'I need a **Y** to my *left* to become **X**'

Categorial Grammars (contd.)

- More formally, CG has two rules:
  Forward Application: $X/Y \quad Y \Rightarrow \mathbf{X} \quad (>)$
  Backward Application: $Y \quad X\backslash Y \Rightarrow \mathbf{X} \quad (<)$
- An example derivation:

$$\begin{array}{ccc}
\textit{Mary} & \textit{likes} & \textit{Peter} \\
\hline
\mathbf{NP} & (\mathbf{S}\backslash\mathbf{NP})/\mathbf{NP} & \mathbf{NP} \\
\hline
 & \mathbf{S}\backslash\mathbf{NP} \; > & \\
\hline
 & \mathbf{S} \; < &
\end{array}$$

Categorial Grammars (contd.)

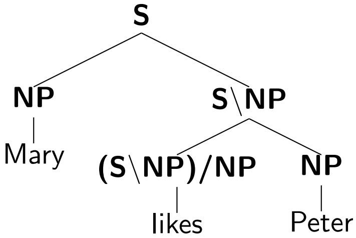
- More formally, CG has two rules:

  Forward Application: $X/Y \quad Y \quad \Rightarrow \quad \mathbf{X} \quad (>)$

  Backward Application: $Y \quad X\backslash Y \quad \Rightarrow \quad \mathbf{X} \quad (<)$

- An example derivation:

$$
\begin{array}{ccc}
\textit{Mary} & \textit{likes} & \textit{Peter} \\
\overline{\mathbf{NP}} & \overline{\mathbf{(S\backslash NP)/NP}} & \overline{\mathbf{NP}} \\
 & \multicolumn{2}{c}{\overline{\qquad\qquad\mathbf{S\backslash NP}\qquad\qquad}^{>}} \\
\multicolumn{3}{c}{\overline{\qquad\qquad\qquad\mathbf{S}\qquad\qquad\qquad}^{<}}
\end{array}
$$



A simple CG learner for learning word-grammars

- Input is a set of words.
- We want to assign a probability, θ, to possible lexical items ($\langle \phi, \sigma \rangle$ pairs).
- Note: probability is the 'system's belief' that the $\langle \phi, \sigma \rangle$ pair at hand is a lexical item.
- Only 3 categories (known in advance):
  - **W** : free morpheme (word, or stem)
  - **W/W** :

A simple CG learner for learning word-grammars

- Input is a set of words.
- We want to assign a probability, θ, to possible lexical items ($\langle\phi, \sigma\rangle$ pairs).
- Note: probability is the 'system's belief' that the $\langle\phi, \sigma\rangle$ pair at hand is a lexical item.
- Only 3 categories (known in advance):
  - **W** : free morpheme (word, or stem)
  - **W/W** : prefix
  - **W\W** :

A simple CG learner for learning word-grammars

- Input is a set of words.
- We want to assign a probability, θ, to possible lexical items ($\langle \phi, \sigma \rangle$ pairs).
- Note: probability is the 'system's belief' that the $\langle \phi, \sigma \rangle$ pair at hand is a lexical item.
- Only 3 categories (known in advance):
  - **W** : free morpheme (word, or stem)
  - **W/W** : prefix
  - **W\W** : suffix
- We adopt a *Beta/Binomial* model.
- We assume each input word provides evidence for the lexical hypothesis in question, If hypothesis used in the interpretation of the input.

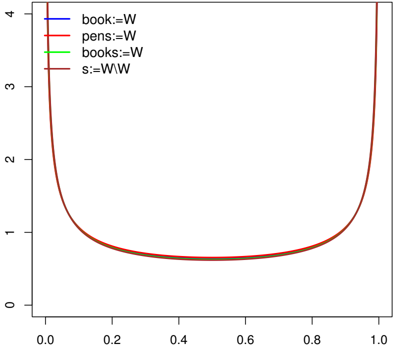
The CG learner: A simple algorithm

- Input is unsegmented words (and the lexicon).
- Output is the lexicalized grammar with probability assignments.

For each input word w,

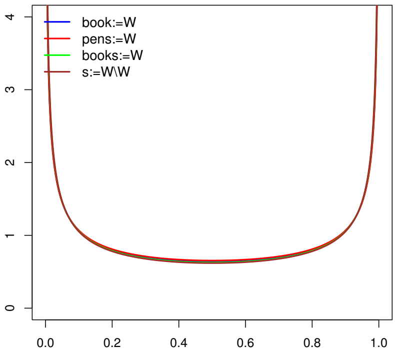
1. Try to segment the input using the current lexicon.
2. If there is no possible segmentation, assume that we have found evidence for a lexical item $w := W$.
3. If we can segment the input as $w = \phi_1 \dots \phi_N$, assume that we have observed evidence for each tuple $\langle \phi_i, \sigma_j \rangle$ which yields a correct parse of w.
4. We update the parameters of the Beta distribution associated with the lexical hypotheses.

An example



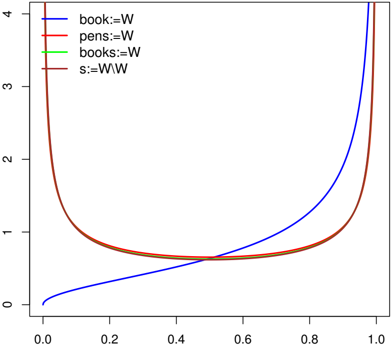
Lexicon	{}
Input	book
Hypotheses	
Parses	

An example



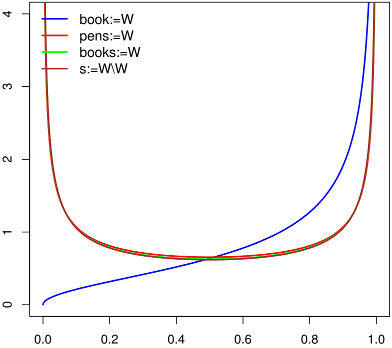
Lexicon	{}
Input	book
Hypotheses	book:=W
Parses	

An example



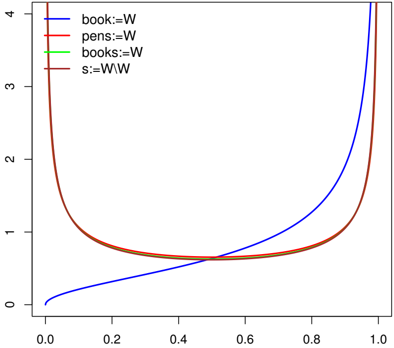
Lexicon	{book:=W}
Input	book
Hypotheses	book:=W
Parses	$\dfrac{\textbf{book}}{\textbf{W}}$

An example



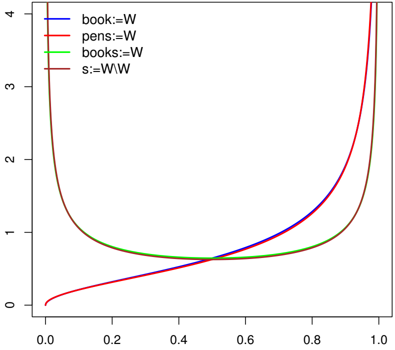
Lexicon	{book:=W}
Input	pens
Hypotheses	
Parses	

An example



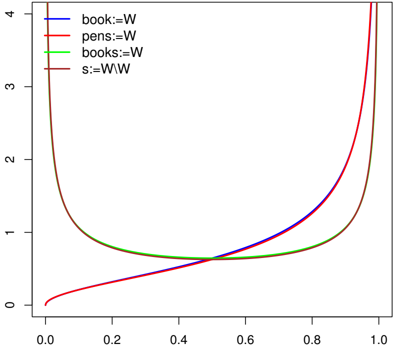
Lexicon	{book:=W}
Input	pens
Hypotheses	pens:=W
Parses	

An example



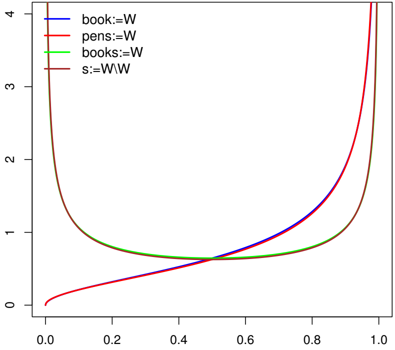
Lexicon	{book:=W, pens:=W}
Input	pens
Hypotheses	pens:=W
Parses	$\dfrac{\textbf{pens}}{\textbf{W}}$

An example



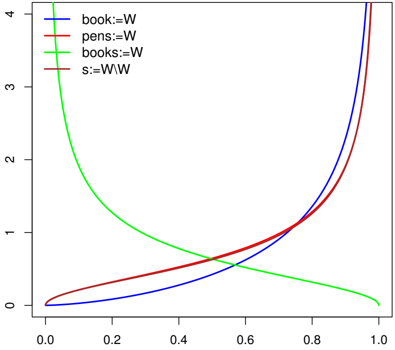
Lexicon	{book:=W, pens:=W}
Input	books
Hypotheses	
Parses	

An example



Lexicon	{book:=W, pens:=W}
Input	books
Hypotheses	book:=W, books:=W, s:=W, book:=W/W, s:=W\W
Parses	

An example



Lexicon $\{$book:=W, pens:=W, s:=W\W$\}$

Input books

Hypotheses book:=W, books:=W, s:=W, book:=W/W, s:=W\W

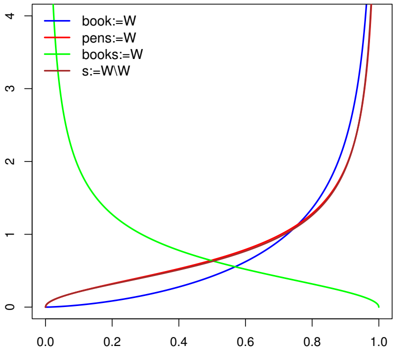
Parses

$$\dfrac{\text{books}}{W}$$

$$\dfrac{\dfrac{\textbf{book}}{\mathbf{W}} \quad \dfrac{\textbf{s}}{\mathbf{W \backslash W}}}{\mathbf{W}}<$$

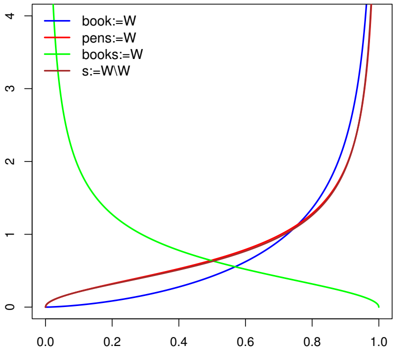
$$\dfrac{\dfrac{\text{book}}{W/W} \quad \dfrac{\text{s}}{W}}{W}>$$

An example



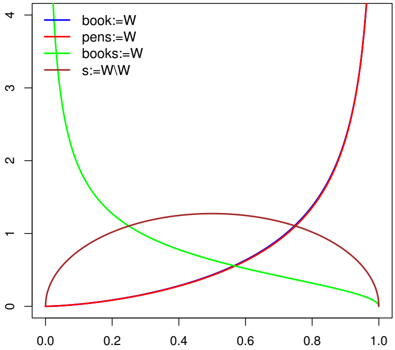
Lexicon	{book:=W, pens:=W, s:=W\W}
Input	pens
Hypotheses	
Parses	

An example



Lexicon	{book:=W, pens:=W, s:=W\W}
Input	pens
Hypotheses	pen:=W, pens:=W, s:=W, pen:=W/W, s:=W\W
Parses	

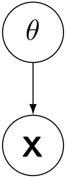
An example



Lexicon	{book:=W, pens:=W, s:=W\W}
Input	pens
Hypotheses	pen:=W, pens:=W, s:=W, pen:=W/W, s:=W\W
Parses	$\dfrac{\textbf{pens}}{\mathbf{W}}$ $\dfrac{\dfrac{\text{pen}}{W} \quad \dfrac{\text{s}}{W\backslash W}}{W}<$
	$\dfrac{\dfrac{\text{pen}}{W/W} \quad \dfrac{\text{s}}{W}}{W}>$

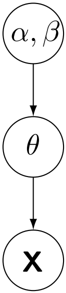
A hierarchical extension

- Our current model assumes rather non-informative values for α and β.
- We can extend this model to get more informative priors

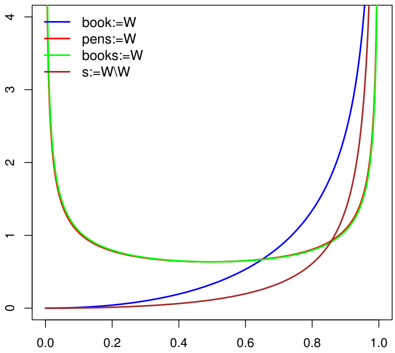


A hierarchical extension

- Our current model assumes rather non-informative values for α and β.
- We can extend this model to get more informative priors
- We treat α and β as random variables.
- We make use of *context predictability* as another source providing a hierarchical informative prior for possible segments.

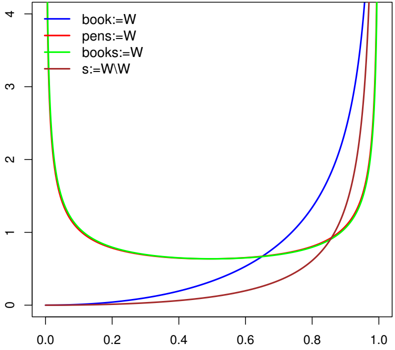


Hierarchical extension: example



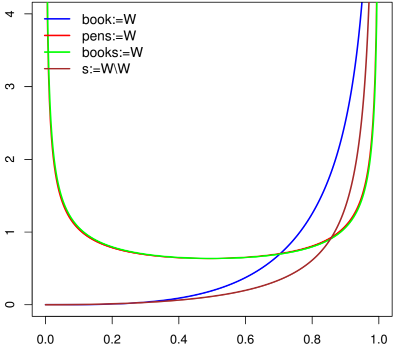
Lexicon	$\{\}$
Input	book
Hypotheses	
Parses	

Hierarchical extension: example



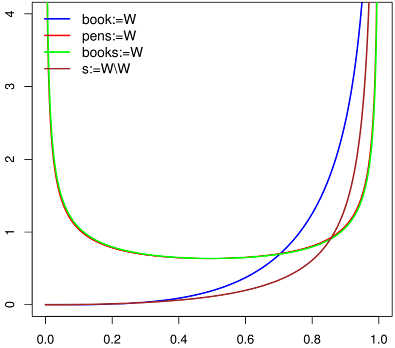
Lexicon	{}
Input	book
Hypotheses	book:=W
Parses	

Hierarchical extension: example



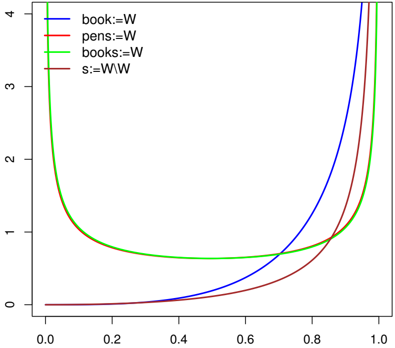
Lexicon	{book:=W}
Input	book
Hypotheses	book:=W
Parses	**book** $\overline{\textbf{W}}$

Hierarchical extension: example



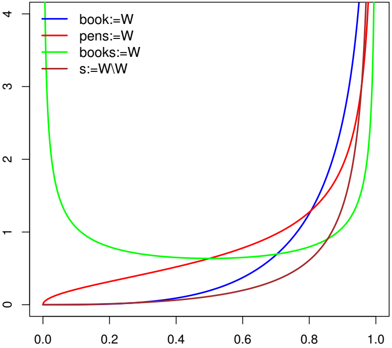
Lexicon	{book:=W}
Input	pens
Hypotheses	
Parses	

Hierarchical extension: example



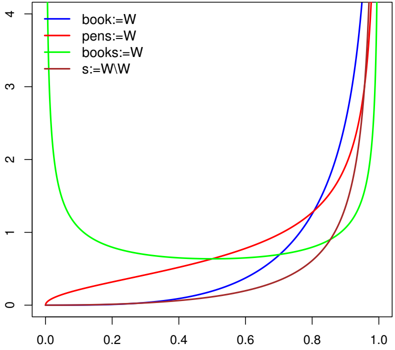
Lexicon	{book:=W}
Input	pens
Hypotheses	pens:=W
Parses	

Hierarchical extension: example



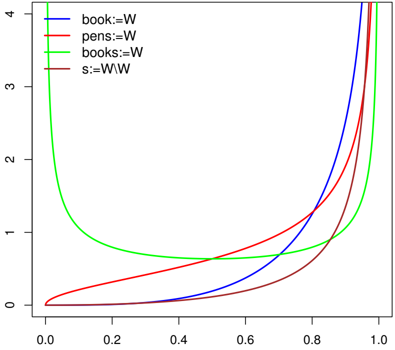
Lexicon	{book:=W, pens:=W}
Input	pens
Hypotheses	pens:=W
Parses	**pens** / **W**

Hierarchical extension: example



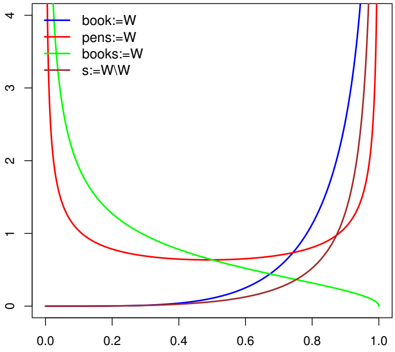
Lexicon	{book:=W, pens:=W}
Input	books
Hypotheses	
Parses	

Hierarchical extension: example



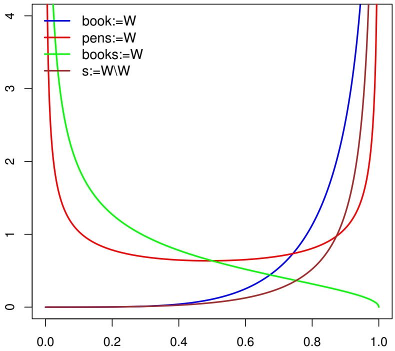
Lexicon	{book:=W, pens:=W}
Input	books
Hypotheses	book:=W, books:=W, s:=W, book:=W/W, s:=W\W
Parses	

Hierarchical extension: example



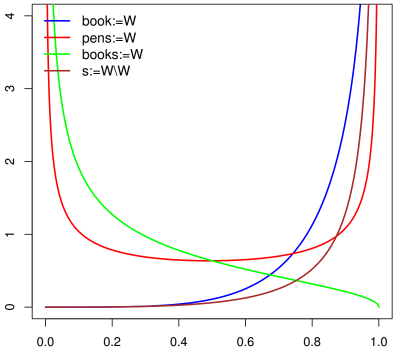
Lexicon	{book:=W, pens:=W, s:=W\W}
Input	books
Hypotheses	book:=W, books:=W, s:=W, book:=W/W, s:=W\W
Parses	$\dfrac{\text{books}}{W}$
	$\dfrac{\dfrac{\textbf{book}}{\mathbf{W}} \quad \dfrac{\textbf{s}}{\mathbf{W \backslash W}}}{\mathbf{W}}<$
	$\dfrac{\dfrac{\text{book}}{W/W} \quad \dfrac{\text{s}}{W}}{W}>$

Hierarchical extension: example



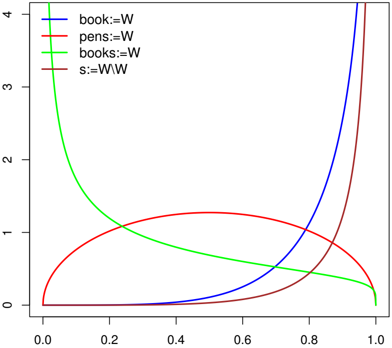
Lexicon	{book:=W, pens:=W, s:=W\W}
Input	pens
Hypotheses	
Parses	

Hierarchical extension: example



Lexicon	{book:=W, pens:=W, s:=W\W}
Input	pens
Hypotheses	pen:=W, pens:=W, s:=W, pen:=W/W, s:=W\W
Parses	

Hierarchical extension: example



Lexicon	{book:=W, pens:=W, s:=W\W, pen:=W}
Input	pens
Hypotheses	pen:=W, pens:=W, s:=W, pen:=W/W, s:=W\W
Parses	$\dfrac{\text{pens}}{W}$ $\dfrac{\dfrac{\textbf{pen}}{\mathbf{W}} \quad \dfrac{\textbf{s}}{\mathbf{W \backslash W}}}{\mathbf{W}}<$ $\dfrac{\dfrac{\text{pen}}{W/W} \quad \dfrac{\text{s}}{W}}{W}>$

Summary

- Bayesian statistics provides a different approach to statistical inference and learning.
- Use of (subjective) priors is not always bad: Modeling cognitive processes is a good example.
- Hierarchical priors is a good way to combine information from different sources.