

# Nonparametric Bayesian Inference for Topical Collocation Models

Mark Johnson

Dept of Computing  
Macquarie University  
Sydney  
Australia

Joint work with Lan Du, Massi Ciaramitra, Zhendong Zhao and Mark Steedman

# Outline

## Introduction

Probabilistic context-free grammars

Topic models as PCFGs

Adaptor grammars: a non-parametric extension of PCFGs

Segmentation with adaptor grammars

Finding topical collocations with adaptor grammars

Efficient implementation with boundary indicator sampling

Experimental evaluation

Conclusions and future work

# Beyond bags of words

- Traditional information retrieval and extraction models treat documents as *bags of words*
- But isolated words can be misleading, especially in *technical domains* such as biomedicine, finance, etc.
  - ▶ a *wash sale* isn't about cleaning anything
  - ▶ the *New York Times* isn't new, and doesn't have anything to do with arithmetic
  - ▶ a *neural net* is not a (e.g., fishing) net, and doesn't have much to do with brains
- Many collocations are *topic-specific*
  - ▶ the *white house* is non-compositional collocation in *politics*, but a compositional phrase in *real-estate*

## Prior work on collocations and topic models

- *Pipeline approaches* identify collocations in corpus in a preprocessing step, and uniformly replace each collocation in corpus with a single token (e.g., *neural net*  $\Rightarrow$  *neural\_net*) before topic modelling (e.g., Lau et al., 2013)
  - + scales well to large corpora
  - collocations are not topic-dependent
- *Extensions to LDA* jointly find topics and collocations
  - ▶ *LDACOL* generates each word either from a document-dependent topic, or from the preceding word (Griffiths et al., 2007)
  - ▶ The *Topical N-gram* model (TNG) generates each word either from a document-dependent topic, or from a combination of the preceding word and its topic (Wang et al., 2007)
    - the algorithms generally don't scale to large corpora
    - collocations aren't topic-dependent in LDACOL
- Our work *jointly infers topics and collocation*, and *the inference algorithm is parallelisable and scales to large corpora*

# Outline of our approach

- We extend *sequence segmentation* models to learn *topical collocations*
  - ▶ LDA topic models can be expressed as PCFGs (Johnson 2010)
  - ▶ *Adaptor grammars* (Johnson et al 2007) are a non-parametric Bayesian generalisation of PCFGs that can express both segmentation models and topic models
  - ▶ Goldwater et al (2006) introduced a non-parametric Bayesian approach to *word segmentation* that uses *point-wise sampling over boundary indicator variables*
- Here we take a *topical collocation model* initially defined as an adaptor grammar, and:
  - ▶ reparameterise it using a generalisation of Goldwater's *boundary indicator variables*, and
  - ▶ develop an efficient, *parallel sampler* that exploits *topic and word sparsity* (Yao et al, 2009; Newman et al., 2009)

# Outline

Introduction

Probabilistic context-free grammars

Topic models as PCFGs

Adaptor grammars: a non-parametric extension of PCFGs

Segmentation with adaptor grammars

Finding topical collocations with adaptor grammars

Efficient implementation with boundary indicator sampling

Experimental evaluation

Conclusions and future work

# Probabilistic Context-Free Grammars

- Probabilistic context-free grammars (PCFGs) define *probability distributions over trees*
- Each *nonterminal node* expands by
  - ▶ choosing a rule expanding that nonterminal, and
  - ▶ recursively expanding any nonterminal children it contains
- Probability of tree is *product of probabilities of rules* used to construct it

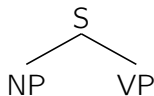
<i>Probability <math>\theta_r</math></i>	<i>Rule <math>r</math></i>	
1	$S \rightarrow NP VP$	S
0.7	$NP \rightarrow Sam$	
0.3	$NP \rightarrow Sandy$	
1	$VP \rightarrow V NP$	
0.8	$V \rightarrow likes$	
0.2	$V \rightarrow hates$	

$$\Pr(\text{Tree}) =$$

# Probabilistic Context-Free Grammars

- Probabilistic context-free grammars (PCFGs) define *probability distributions over trees*
- Each *nonterminal node* expands by
  - ▶ choosing a rule expanding that nonterminal, and
  - ▶ recursively expanding any nonterminal children it contains
- Probability of tree is *product of probabilities of rules* used to construct it

<i>Probability <math>\theta_r</math></i>	<i>Rule <math>r</math></i>
1	$S \rightarrow NP VP$
0.7	$NP \rightarrow Sam$
0.3	$NP \rightarrow Sandy$
1	$VP \rightarrow V NP$
0.8	$V \rightarrow likes$
0.2	$V \rightarrow hates$



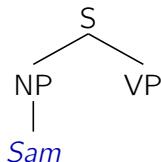
$$\Pr(\text{Tree}) = 1 \times$$



# Probabilistic Context-Free Grammars

- Probabilistic context-free grammars (PCFGs) define *probability distributions over trees*
- Each *nonterminal node* expands by
  - ▶ choosing a rule expanding that nonterminal, and
  - ▶ recursively expanding any nonterminal children it contains
- Probability of tree is *product of probabilities of rules* used to construct it

<i>Probability <math>\theta_r</math></i>	<i>Rule <math>r</math></i>
1	$S \rightarrow NP VP$
0.7	$NP \rightarrow Sam$
0.3	$NP \rightarrow Sandy$
1	$VP \rightarrow V NP$
0.8	$V \rightarrow likes$
0.2	$V \rightarrow hates$

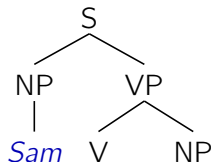


$$\text{Pr}(\text{Tree}) = 1 \times 0.7 \times$$

# Probabilistic Context-Free Grammars

- Probabilistic context-free grammars (PCFGs) define *probability distributions over trees*
- Each *nonterminal node* expands by
  - ▶ choosing a rule expanding that nonterminal, and
  - ▶ recursively expanding any nonterminal children it contains
- Probability of tree is *product of probabilities of rules* used to construct it

<i>Probability <math>\theta_r</math></i>	<i>Rule <math>r</math></i>
1	$S \rightarrow NP VP$
0.7	$NP \rightarrow Sam$
0.3	$NP \rightarrow Sandy$
1	$VP \rightarrow V NP$
0.8	$V \rightarrow likes$
0.2	$V \rightarrow hates$

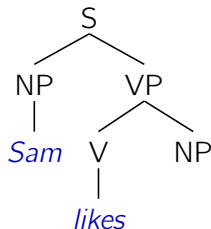


$$\Pr(\text{Tree}) = 1 \times 0.7 \times 1 \times$$

# Probabilistic Context-Free Grammars

- Probabilistic context-free grammars (PCFGs) define *probability distributions over trees*
- Each *nonterminal node* expands by
  - ▶ choosing a rule expanding that nonterminal, and
  - ▶ recursively expanding any nonterminal children it contains
- Probability of tree is *product of probabilities of rules* used to construct it

<i>Probability <math>\theta_r</math></i>	<i>Rule <math>r</math></i>
1	$S \rightarrow NP VP$
0.7	$NP \rightarrow Sam$
0.3	$NP \rightarrow Sandy$
1	$VP \rightarrow V NP$
0.8	$V \rightarrow likes$
0.2	$V \rightarrow hates$

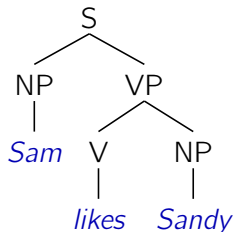


$$\Pr(\text{Tree}) = 1 \times 0.7 \times 1 \times 0.8 \times$$

# Probabilistic Context-Free Grammars

- Probabilistic context-free grammars (PCFGs) define *probability distributions over trees*
- Each *nonterminal node* expands by
  - ▶ choosing a rule expanding that nonterminal, and
  - ▶ recursively expanding any nonterminal children it contains
- Probability of tree is *product of probabilities of rules* used to construct it

<i>Probability <math>\theta_r</math></i>	<i>Rule <math>r</math></i>
1	$S \rightarrow NP VP$
0.7	$NP \rightarrow Sam$
0.3	$NP \rightarrow Sandy$
1	$VP \rightarrow V NP$
0.8	$V \rightarrow likes$
0.2	$V \rightarrow hates$



$$\Pr(\text{Tree}) = 1 \times 0.7 \times 1 \times 0.8 \times 0.3$$

# PCFGs as models of natural language syntax

- Simple PCFGs like this are *not very good models of natural language syntax*
  - ▶ PCFGs aren't good parameterisations of natural language
  - ▶ accurate PCFGs need thousands of nonterminal symbols and hundreds of thousands of rules
- ⇒ smoothing is an essential “black art”
  - ▶ unsupervised estimators of PCFGs perform very poorly *even when initialised with correct parses*
- But PCFGs can model many other interesting things!

# Outline

Introduction

Probabilistic context-free grammars

**Topic models as PCFGs**

Adaptor grammars: a non-parametric extension of PCFGs

Segmentation with adaptor grammars

Finding topical collocations with adaptor grammars

Efficient implementation with boundary indicator sampling

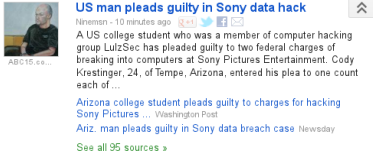
Experimental evaluation


Conclusions and future work

# Topic models for document processing

- Topic models *cluster words and documents into topics*
  - ▶ usually *unsupervised* (i.e., topics aren't given in training data)
- Important for document analysis and information extraction
  - ▶ Example: clustering news stories for information retrieval
  - ▶ Example: tracking evolution of a research topic over time

Computers

  
Ninemsn - 10 minutes ago  
A US college student who was a member of computer hacking group LulzSec has pleaded guilty to two federal charges of breaking into computers at Sony Pictures Entertainment. Cody Krestinger, 24, of Tempe, Arizona, entered his plea to one count each of ...  
[Arizona college student pleads guilty to charges for hacking Sony Pictures](#) ... Washington Post  
[Ariz. man pleads guilty in Sony data breach case](#) Newsday  
[See all 95 sources >](#)

  
BBC News - 10 hours ago  
More than half a million Apple computers have been infected with the Flashback Trojan, according to a Russian anti-virus firm.  
[Mac Computers Affected by Hacker Attack: Researcher](#) BusinessWeek  
[Apple Mac Computers Hit in Hacker Attack, Researcher Says](#) Bloomberg  
[In Depth: Mac Botnet Infects More Than 600000 Apple Computers](#) alWeek  
[See all 230 sources >](#)

# Mixture versus admixture topic models

- In a *mixture model*, each document has a *single topic*
  - ▶ all words in the document come from this topic
- In *admixture models*, each document has a *distribution over topics*
  - ▶ a single document can have multiple topics (number of topics in a document controlled by prior)
  - ⇒ can capture more complex relationships between documents than a mixture model
- Both mixture and admixture topic models typically use a “*bag of words*” representation of a document



## Example: documents from NIPS corpus

Annotating an unlabeled dataset is one of the bottlenecks in using supervised learning to build good predictive models. Getting a dataset labeled by experts can be expensive and time consuming. With the advent of crowdsourcing services ...

---

The task of recovering intrinsic images is to separate a given input image into its material-dependent properties, known as reflectance or albedo, and its light-dependent properties, such as shading, shadows, specular highlights, ...

---

In each trial of a standard visual short-term memory experiment, subjects are first presented with a display containing multiple items with simple features (e.g. colored squares) for a brief duration and then, after a delay interval, their memory for ...

---

Many studies have uncovered evidence that visual cortex contains specialized regions involved in processing faces but not other object classes. Recent electrophysiology studies of cells in several of these specialized regions revealed that at least some ...

## Example (cont): ignore function words

Annotating an unlabeled dataset is one of the bottlenecks in using supervised learning to build good predictive models. Getting a dataset labeled by experts can be expensive and time consuming. With the advent of crowdsourcing services ...

---

The task of recovering intrinsic images is to separate a given input image into its material-dependent properties, known as reflectance or albedo, and its light-dependent properties, such as shading, shadows, specular highlights, ...

---

In each trial of a standard visual short-term memory experiment, subjects are first presented with a display containing multiple items with simple features (e.g. colored squares) for a brief duration and then, after a delay interval, their memory for ...

---

Many studies have uncovered evidence that visual cortex contains specialized regions involved in processing faces but not other object classes. Recent electrophysiology studies of cells in several of these specialized regions revealed that at least some ...

## Example (cont): mixture topic model

Annotating an unlabeled dataset is one of the bottlenecks in using supervised learning to build good predictive models. Getting a dataset labeled by experts can be expensive and time consuming. With the advent of crowdsourcing services ...

---

The task of recovering intrinsic images is to separate a given input image into its material-dependent properties, known as reflectance or albedo, and its light-dependent properties, such as shading, shadows, specular highlights, ...

---

In each trial of a standard visual short-term memory experiment, subjects are first presented with a display containing multiple items with simple features (e.g. colored squares) for a brief duration and then, after a delay interval, their memory for ...

---

Many studies have uncovered evidence that visual cortex contains specialized regions involved in processing faces but not other object classes. Recent electrophysiology studies of cells in several of these specialized regions revealed that at least some ...

## Example (cont): admixture topic model

Annotating an unlabeled dataset is one of the bottlenecks in using supervised learning to build good predictive models. Getting a dataset labeled by experts can be expensive and time consuming. With the advent of crowdsourcing services ...

---

The task of recovering intrinsic images is to separate a given input image into its material-dependent properties, known as reflectance or albedo, and its light-dependent properties, such as shading, shadows, specular highlights, ...

---

In each trial of a standard visual short-term memory experiment, subjects are first presented with a display containing multiple items with simple features (e.g. colored squares) for a brief duration and then, after a delay interval, their memory for ...

---

Many studies have uncovered evidence that visual cortex contains specialized regions involved in processing faces but not other object classes. Recent electrophysiology studies of cells in several of these specialized regions revealed that at least some ...

# This paper's goal: Collocation topic models

Annotating an unlabeled dataset is one of the bottlenecks in using supervised learning to build good predictive models. Getting a dataset labeled by experts can be expensive and time consuming. With the advent of crowdsourcing services ...

---

The task of recovering intrinsic images is to separate a given input image into its material-dependent properties, known as reflectance or albedo, and its light-dependent properties, such as shading, shadows, specular highlights, ...

---

In each trial of a standard visual short-term memory experiment, subjects are first presented with a display containing multiple items with simple features (e.g. colored squares) for a brief duration and then, after a delay interval, their memory for ...

---

Many studies have uncovered evidence that visual cortex contains specialized regions involved in processing faces but not other object classes. Recent electrophysiology studies of cells in several of these specialized regions revealed that at least some ...

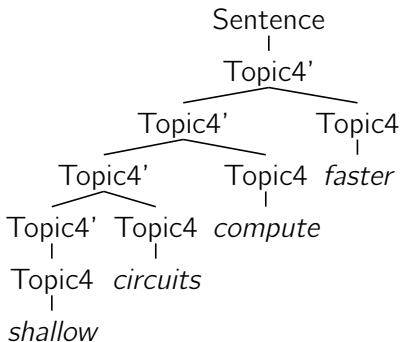
## Mixture versus admixture models

- Admixture models are more complex than mixture models
  - ⇒ Admixture models often *require more data to learn*
- Mixture models can describe shorter documents (phrases, clauses or single sentences) fairly well, where one topic per document assumption is not too bad
  - ▶ e.g., Twitter posts
- Admixture models are better for longer documents, which are likely to have more than one topic
  - ▶ e.g., long news articles

# Mixture topic models as PCFGs (1)

- Idea: Design PCFG so that:
  - ▶ non-deterministic rules implement generative steps in topic model
  - ▶ deterministic rules propagate information to appropriate place

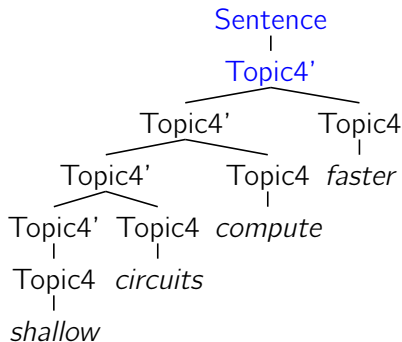
Sentence  $\rightarrow$  Topic $_i'$        $i \in 1, \dots, \ell$   
Topic $_i'$   $\rightarrow$  Topic $_i'$  Topic $_i$        $i \in 1, \dots, \ell$   
Topic $_i'$   $\rightarrow$  Topic $_i$        $i \in 1, \dots, \ell$   
Topic $_i$   $\rightarrow$   $w$        $i \in 1, \dots, \ell$   
                                  $w \in \mathcal{W}$



## Mixture topic models as PCFGs (2)

- Choose a topic for sentence (non-deterministically)

Sentence  $\rightarrow$  Topic $_i'$        $i \in 1, \dots, \ell$   
Topic $_i'$   $\rightarrow$  Topic $_i'$  Topic $_i$        $i \in 1, \dots, \ell$   
Topic $_i'$   $\rightarrow$  Topic $_i$        $i \in 1, \dots, \ell$   
Topic $_i$   $\rightarrow$   $w$        $i \in 1, \dots, \ell$   
    $w \in \mathcal{W}$

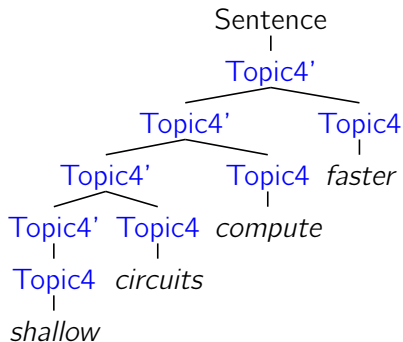




# Mixture topic models as PCFGs (3)

- Copy sentence topic to each word (deterministically)

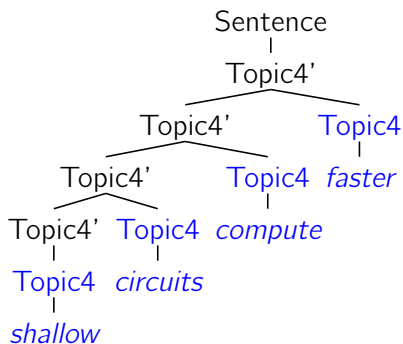
Sentence  $\rightarrow$  Topic $_i'$        $i \in 1, \dots, \ell$   
Topic $_i'$   $\rightarrow$  Topic $_i'$  Topic $_i$      $i \in 1, \dots, \ell$   
Topic $_i'$   $\rightarrow$  Topic $_i$              $i \in 1, \dots, \ell$   
Topic $_i$   $\rightarrow$   $w$                      $w \in \mathcal{W}$



## Mixture topic models as PCFGs (4)

- Generate each word from sentence topic (non-deterministically)

Sentence  $\rightarrow$  Topic $_i'$        $i \in 1, \dots, \ell$   
Topic $_i'$   $\rightarrow$  Topic $_i'$  Topic $_i$        $i \in 1, \dots, \ell$   
Topic $_i'$   $\rightarrow$  Topic $_i$        $i \in 1, \dots, \ell$   
Topic $_i$   $\rightarrow$   $w$        $w \in \mathcal{W}$



# Admixture topic models as PCFGs

- Admixture topic models are usually applied to entire documents
- Standard PCFG parsing algorithms require time proportional to *cube* of sentence length
  - ▶ while PCFGs can generate full documents, with standard parsing algorithms they would be unacceptably slow
  - ▶ see Luong et al. (2013) for a predictive parsing algorithm for very long strings
- *Document ids* let us break a document into several smaller chunks
  - ▶ a document id is a special nonterminal identifying the document this input came from

# Admixture topic models as PCFGs (1)

- Prefix strings from document  $j$  with a *document identifier* “ $\_j$ ”

Sentence  $\rightarrow$  Doc' $_j$      $j \in 1, \dots, m$

Doc' $_j \rightarrow$   $\_j$      $j \in 1, \dots, m$

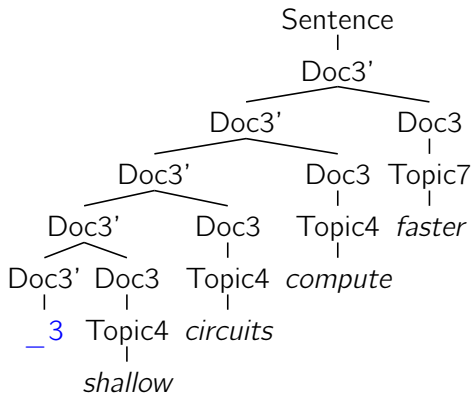
Doc' $_j \rightarrow$  Doc' $_j$  Doc $_j$      $j \in 1, \dots, m$

Doc $_j \rightarrow$  Topic $_i$      $i \in 1, \dots, \ell$

Topic $_i \rightarrow$   $w$      $j \in 1, \dots, m$

Topic $_i \rightarrow$   $w$      $i \in 1, \dots, \ell$

Topic $_i \rightarrow$   $w$      $w \in \mathcal{W}$



# Admixture topic models as PCFGs (2)

- Spine deterministically *propagates document id up through tree*

Sentence  $\rightarrow \text{Doc}'_j \quad j \in 1, \dots, m$

$\text{Doc}'_j \rightarrow \_j \quad j \in 1, \dots, m$

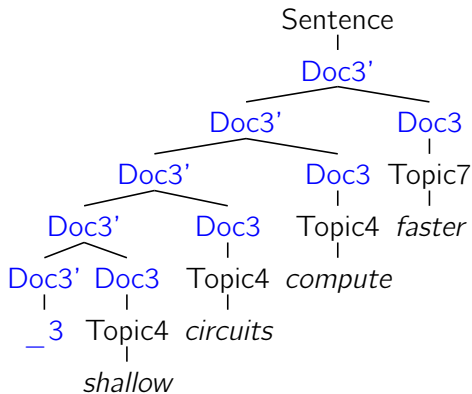
$\text{Doc}'_j \rightarrow \text{Doc}'_j \text{ Doc}_j \quad j \in 1, \dots, m$

$\text{Doc}_j \rightarrow \text{Topic}_i \quad i \in 1, \dots, \ell$

$j \in 1, \dots, m$

$\text{Topic}_i \rightarrow w \quad i \in 1, \dots, \ell$

$w \in \mathcal{W}$



# Admixture topic models as PCFGs (3)

- $\text{Doc}_j \rightarrow \text{Topic}_i$  rules nondeterministically map *documents to topics*

$\text{Sentence} \rightarrow \text{Doc}'_j \quad j \in 1, \dots, m$

$\text{Doc}'_j \rightarrow \_j \quad j \in 1, \dots, m$

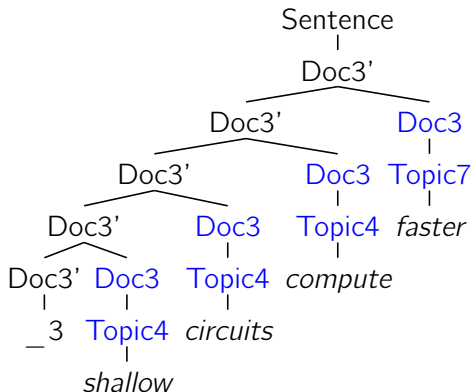
$\text{Doc}'_j \rightarrow \text{Doc}'_j \text{ Doc}_j \quad j \in 1, \dots, m$

$\text{Doc}_j \rightarrow \text{Topic}_i \quad i \in 1, \dots, \ell$

$\text{Doc}_j \rightarrow \text{Topic}_i \quad j \in 1, \dots, m$

$\text{Topic}_i \rightarrow w \quad i \in 1, \dots, \ell$

$\text{Topic}_i \rightarrow w \quad w \in \mathcal{W}$



# Admixture topic models as PCFGs (4)

- $\text{Topic}_i \rightarrow w$  rules nondeterministically map *topics to words*

$\text{Sentence} \rightarrow \text{Doc}'_j \quad j \in 1, \dots, m$

$\text{Doc}'_j \rightarrow \_j \quad j \in 1, \dots, m$

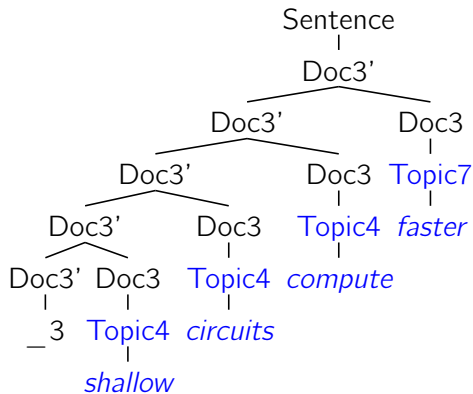
$\text{Doc}'_j \rightarrow \text{Doc}'_j \text{ Doc}_j \quad j \in 1, \dots, m$

$\text{Doc}_j \rightarrow \text{Topic}_i \quad i \in 1, \dots, \ell$

$\text{Doc}_j \rightarrow \text{Topic}_i \quad j \in 1, \dots, m$

$\text{Topic}_i \rightarrow w \quad i \in 1, \dots, \ell$

$\text{Topic}_i \rightarrow w \quad w \in \mathcal{W}$



## Why are these reductions interesting?

- *Not* claiming that topic modelling should be done using PCFGs
  - ▶ PCFG parsing takes time proportional to *cube* of document length
  - ▶ standard topic model algorithms take time *linear* in document length
- The PCFG reductions suggest *new kinds of models that merge grammars and topic models*
  - ▶ easily implemented and evaluated (on small corpora at least)
- Grammars are good at:
  - ▶ grouping words into hierarchically-structured larger units
  - ▶ tracking relative ordering of these units



# Outline

Introduction

Probabilistic context-free grammars

Topic models as PCFGs

**Adaptor grammars: a non-parametric extension of PCFGs**

Segmentation with adaptor grammars

Finding topical collocations with adaptor grammars

Efficient implementation with boundary indicator sampling

Experimental evaluation

Conclusions and future work

# Motivation for adaptor grammars

- PCFGs are *parametric models*
  - ▶ a PCFG can be viewed as a set of multinomials (one for each nonterminal)
  - ▶ learning a PCFG  $\Rightarrow$  setting the rule probabilities
- But in some cases *the rules* themselves have to be learnt
- One way to formulate this:
  - ▶ there is an *infinite set* of *possible rules*
  - ▶ but *only finitely many have non-zero probability*
- In an adaptor grammar, the *possible rules are the yields of the trees generated by a PCFG*
  - ▶ adaptor grammars formalise this by using a PCFG to define the base distribution of a *Dirichlet Process* or a *Pitman-Yor Process*
  - ▶ recursion in the PCFG  $\Rightarrow$  *hierarchical Dirichlet Processes*

# Outline

Introduction

Probabilistic context-free grammars

Topic models as PCFGs

Adaptor grammars: a non-parametric extension of PCFGs

**Segmentation with adaptor grammars**

Finding topical collocations with adaptor grammars

Efficient implementation with boundary indicator sampling

Experimental evaluation

Conclusions and future work

# Unsupervised word segmentation

- *Word segmentation* task: *segment utterances into words* (Elman 1993, Brent 1996)
- Input: phoneme sequences with *sentence boundaries*
- Task: identify *word boundaries*, and hence words

y Δ u ▲ w Δ a Δ n Δ t ▲ t Δ u ▲ s Δ i ▲ D Δ 6 ▲ b Δ U Δ k  
"you want to see the book"

- Ignoring phonology and morphology, this involves *learning the pronunciations of the lexical items* in the language

# CFG models of word segmentation

Words  $\rightarrow$  Word

Words  $\rightarrow$  Word Words

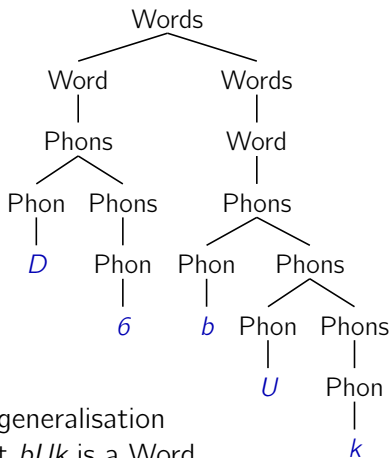
Word  $\rightarrow$  Phons

Phons  $\rightarrow$  Phon

Phons  $\rightarrow$  Phon Phons

Phon  $\rightarrow a | b | \dots$

- CFG trees can *describe* segmentation, but
- PCFGs *can't distinguish* good segmentations from bad ones
  - ▶ PCFG rules are *too small* a unit of generalisation
  - ▶ need to learn e.g., probability that *bUk* is a Word



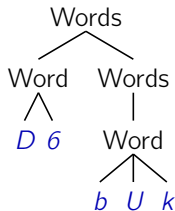
# Towards non-parametric grammars

Words  $\rightarrow$  Word

Words  $\rightarrow$  Word Words

Word  $\rightarrow$  *all possible phoneme sequences*

- Learn probability Word  $\rightarrow$  *b U k*
- But *infinitely many possible Word expansions*  
 $\Rightarrow$  this grammar is *not a PCFG*
- Given *fixed training data*, only finitely many useful rules  
 $\Rightarrow$  use data to choose Word rules as well as their probabilities
- An adaptor grammar can do precisely this!



# Unigram adaptor grammar (Brent)

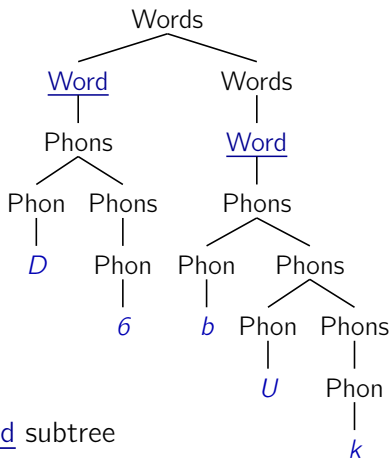
Words  $\rightarrow$  Word

Words  $\rightarrow$  Word Words

Word  $\rightarrow$  Phons

Phons  $\rightarrow$  Phon

Phons  $\rightarrow$  Phon Phons



- Word nonterminal is adapted
- $\Rightarrow$  To generate a Word:
- ▶ select a previously generated Word subtree with probability  $\propto$  number of times it has been generated
  - ▶ expand using Word  $\rightarrow$  Phons rule with probability  $\propto \alpha_{\text{Word}}$  and recursively expand Phons

# Adaptor grammars as a non-parametric extension of PCFGs

- An adaptor grammar *reuses previously-generated subtrees*  $T_A$  of adapted nonterminals  $A$
- This is equivalent to *adding a rule*  $A \rightarrow w$  to the grammar, where  $w$  is the yield of  $T_A$ 
  - ▶ for implementation efficiency, adaptor grammars constrain  $w$  to *only consist of terminals*
  - ▶ *Fragment Grammars* (O'Donnell 2009) lift this restriction
- If the base CFG generates an *infinite number of trees*  $T_A$  for  $A$ , then the adaptor grammar is *non-parametric*
- But any set of sample parses for a *finite training corpus* only contains a *finite number of number of adapted subtrees*
  - ⇒ *sampling methods* (e.g., MCMC) are a natural approach to learning and parsing adaptor grammars
  - ▶ in implementation terms, an adaptor grammar is like a PCFG with a *constantly changing set of rules*



# Computation with adaptor grammars

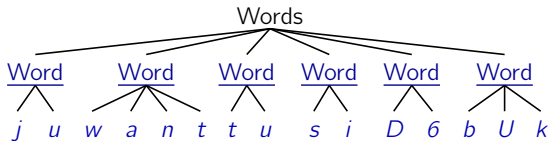
- Adaptor grammars are *strictly more expressive than PCFGs*
    - ▶ non-parametric  $\Rightarrow$  can't be represented by a finite parameter vector
  - But the *posterior predictive distribution* can be *approximated by a PCFG* where *the rules vary depending on the data*
- $\Rightarrow$  Metropolis-within-Gibbs MCMC sampler (Johnson et al., 2007)  
repeat forever:
- ▶ randomly pick a string from training data
  - ▶ compute approximating PCFG for posterior predictive distribution given parses for other sentences
  - ▶ *sample a parse from approximating PCFG*
  - ▶ use a Metropolis-Hastings accept-reject step to correct for approximation
- The parsing step is usually the slowest (cubic in length of string)
  - Cohen et al. (2010) have developed a mean-field variational Bayes inference algorithm for adaptor grammars

# Unigram model of word segmentation

- Unigram “bag of words” model (Brent):
  - ▶ generate a *dictionary*, i.e., a set of words, where each word is a random sequence of phonemes
    - Bayesian prior prefers smaller dictionaries
  - ▶ generate each utterance by choosing each word at random from dictionary
- Brent’s unigram model as an adaptor grammar:

Words  $\rightarrow$  Word<sup>+</sup>

Word  $\rightarrow$  Phoneme<sup>+</sup>



- Accuracy of word segmentation learnt: *56% token f-score* (same as Brent model)
- But we can construct many more word segmentation models using AGs

# Adaptor grammar learnt from Brent corpus

- **Initial grammar**

1	Words $\rightarrow$ <u>Word</u> Words	1	Words $\rightarrow$ <u>Word</u>
1	<u>Word</u> $\rightarrow$ Phon		
1	Phons $\rightarrow$ Phon Phons	1	Phons $\rightarrow$ Phon
1	Phon $\rightarrow$ <i>D</i>	1	Phon $\rightarrow$ <i>G</i>
1	Phon $\rightarrow$ <i>A</i>	1	Phon $\rightarrow$ <i>E</i>

- **A grammar learnt from Brent corpus**

16625	Words $\rightarrow$ <u>Word</u> Words	9791	Words $\rightarrow$ <u>Word</u>
1575	<u>Word</u> $\rightarrow$ Phons		
4962	Phons $\rightarrow$ Phon Phons	1575	Phons $\rightarrow$ Phon
134	Phon $\rightarrow$ <i>D</i>	41	Phon $\rightarrow$ <i>G</i>
180	Phon $\rightarrow$ <i>A</i>	152	Phon $\rightarrow$ <i>E</i>
460	<u>Word</u> $\rightarrow$ (Phons (Phon <i>y</i> ) (Phons (Phon <i>u</i> )))		
446	<u>Word</u> $\rightarrow$ (Phons (Phon <i>w</i> ) (Phons (Phon <i>A</i> ) (Phons (Phon <i>t</i> ))))		
374	<u>Word</u> $\rightarrow$ (Phons (Phon <i>D</i> ) (Phons (Phon <i>6</i> )))		
372	<u>Word</u> $\rightarrow$ (Phons (Phon <i>&amp;</i> ) (Phons (Phon <i>n</i> ) (Phons (Phon <i>d</i> ))))		

# More complex adaptor grammar models of word segmentation

- Because adaptor grammar models generalise PCFGs, we can combine the topic model grammars and word segmentation grammars
    - ▶ non-linguistic context does improve word segmentation
    - ▶ social cues do not improve word segmentation (as far as we can tell)
  - We can learn the internal structure of words too
    - ▶ words are a sequence of syllables
    - ▶ learn syllable structure jointly with word segmentation
    - ▶ we can learn different structures for word-peripheral and word-internal syllables
- ⇒ the best reported accuracy for unsupervised word segmentation (89% f-score)

# Outline

Introduction

Probabilistic context-free grammars

Topic models as PCFGs

Adaptor grammars: a non-parametric extension of PCFGs

Segmentation with adaptor grammars

**Finding topical collocations with adaptor grammars**

Efficient implementation with boundary indicator sampling

Experimental evaluation

Conclusions and future work

# Topical collocation models

Annotating an unlabeled dataset is one of the bottlenecks in using supervised learning to build good predictive models. Getting a dataset labeled by experts can be expensive and time consuming. With the advent of crowdsourcing services ...

---

The task of recovering intrinsic images is to separate a given input image into its material-dependent properties, known as reflectance or albedo, and its light-dependent properties, such as shading, shadows, specular highlights, ...

---

In each trial of a standard visual short-term memory experiment, subjects are first presented with a display containing multiple items with simple features (e.g. colored squares) for a brief duration and then, after a delay interval, their memory for ...

---

Many studies have uncovered evidence that visual cortex contains specialized regions involved in processing faces but not other object classes. Recent electrophysiology studies of cells in several of these specialized regions revealed that at least some ...

# Topic model with collocations

- Combines *PCFG for admixture topic model* and *segmentation adaptor grammar*

Sentence  $\rightarrow$  Doc<sub>*j*</sub>       $j \in 1, \dots, m$

Doc<sub>*j*</sub>  $\rightarrow$  *j*       $j \in 1, \dots, m$

Doc<sub>*j*</sub>  $\rightarrow$  Doc<sub>*j*</sub> Topic<sub>*i*</sub>       $i \in 1, \dots, \ell;$

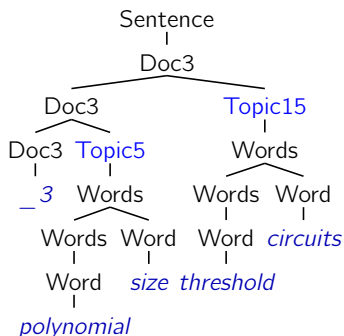
$j \in 1, \dots, m$

Topic<sub>*i*</sub>  $\rightarrow$  Words       $i \in 1, \dots, \ell$

Words  $\rightarrow$  Word

Words  $\rightarrow$  Words Word

Word  $\rightarrow$  *w*       $w \in \mathcal{W}$



# Data preparation in Griffiths et al (2007)

- Documents are papers from NIPS proceedings ( $\sim 3$  million words)
- Case normalised
- Segmented at *punctuation* and *function words*

annotating an unlabeled dataset is one of the bottlenecks in using supervised learning to build good predictive models. getting a dataset labeled by experts can be expensive and time consuming. with the advent of crowdsourcing services ...

---

the task of recovering intrinsic images is to separate a given input image into its material-dependent properties, known as reflectance or albedo, and its light-dependent properties, such as shading, shadows, specular highlights, ...



## Finding topical collocations in NIPS abstracts

- Run topical collocation adaptor grammar on NIPS corpus
- Run with  $\ell = 20$  topics (i.e., 20 distinct  $\text{Topic}_i$  nonterminals)
- Corpus is segmented by punctuation
  - ▶ terminal strings are fairly short
  - ⇒ inference is fairly efficient
- Used Pitman-Yor adaptors
  - ▶ sampled Pitman-Yor  $a$  and  $b$  parameters
  - ▶ flat and “vague Gamma” priors on Pitman-Yor  $a$  and  $b$  parameters

## Sample output on NIPS corpus, 20 topics

- Multiword subtrees learned by adaptor grammar:

T_0 → gradient descent	T_1 → associative memory
T_0 → cost function	T_1 → standard deviation
T_0 → fixed point	T_1 → randomly chosen
T_0 → learning rates	T_1 → hamming distance
T_3 → membrane potential	T_10 → ocular dominance
T_3 → action potentials	T_10 → visual field
T_3 → visual system	T_10 → nervous system
T_3 → primary visual cortex	T_10 → action potential
- Sample skeletal parses:
  - \_3 (T\_5 polynomial size) (T\_15 threshold circuits)
  - \_4 (T\_11 studied) (T\_19 pattern recognition algorithms)
  - \_4 (T\_2 feedforward neural network) (T\_1 implements)
  - \_5 (T\_11 single) (T\_10 ocular dominance stripe) (T\_12 low)  
(T\_3 ocularity) (T\_12 drift rate)

## Some collocations found in NIPS corpus

<b>Count</b>	<b>Topic</b>	<b>Collocation</b>
2	T0	unites states israeli binational science foundation bsf
2	T5	batch k-means empty circles online gradient
12	T1	partially observable markov decision processes
12	T2	defense advanced research projects agency
7	T5	radial basis function rbf network
5	T6	analog vlsi neural network chip
4	T12	national science foundation graduate fellowship
3	T10	globally optimal on-line learning rules
3	T12	radial basis function rbf units
3	T13	non-parametric multi-scale statistical image model
3	T15	weight vector estimate requires knowledge
3	T17	orientation bands intersect ocular dominance
3	T18	optimal brain damage le cun
3	T6	normalized mean squared prediction error
47	T5	markov chain monte carlo
43	T12	radial basis function rbf
41	T12	radial basis function networks
39	T7	independent component analysis ica
35	T11	principal component analysis pca
29	T11	hidden markov models hmms
23	T12	radial basis function network

## Some collocations found in NIPS corpus (cont.)

<b>Count</b>	<b>Topic</b>	<b>Collocation</b>
17	T11	principal components analysis pca
16	T11	hidden markov models hmm
14	T18	artificial neural network ann
13	T15	optimal brain damage obd
12	T4	kanerva sparse distributed memory
11	T14	hybrid monte carlo method
11	T19	artificial neural networks ann
10	T0	mean square error mse
10	T12	radial basis functions rbfs
10	T16	markov decision process pomdp
10	T11	hidden markov model hmm
10	T3	atr human information processing
10	T18	artificial neural networks anns
10	T9	spin spin correlation function
9	T2	naive mean field approximation
9	T0	mean squared error mse
9	T7	support vector machines svms
9	T8	owl sound localization system
8	T1	compatible lateral bipolar transistors
8	T13	nsf presidential young investigator
8	T14	basic differential multiplier method

# Outline

Introduction

Probabilistic context-free grammars

Topic models as PCFGs

Adaptor grammars: a non-parametric extension of PCFGs

Segmentation with adaptor grammars

Finding topical collocations with adaptor grammars

**Efficient implementation with boundary indicator sampling**

Experimental evaluation

Conclusions and future work

# Boundary indicators in word segmentation models

y  $\Delta$  u  $\blacktriangle$  w  $\Delta$  a  $\Delta$  n  $\Delta$  t  $\blacktriangle$  t  $\Delta$  u  $\blacktriangle$  s  $\Delta$  i  $\blacktriangle$  D  $\Delta$  6  $\blacktriangle$  b  $\Delta$  U  $\Delta$  k  
“you want to see the book”

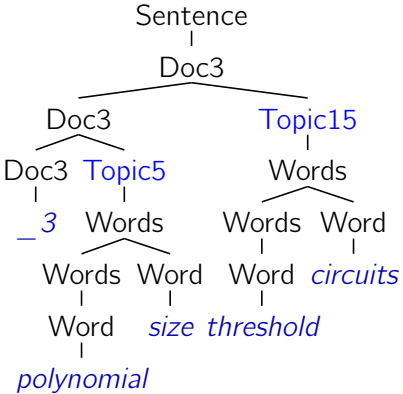
- Boolean *boundary indicator variables* are located between each adjacent pair of elements
  - Isomorphism between assignments to boundary indicator variables and sequence segmentations
  - Goldwater et al. (2006) word segmentation model samples possible segmentations by Gibbs sampling the boundary indicator variables
    - ▶ each Gibbs step only requires the *ratio of the probabilities of segmentations with the boundary present and without the boundary present*
- ⇒ no difficult-to-compute partition function!

# Boundary indicator representation of topical collocations

- Boundary indicator variables range over possible topics, plus a special “null topic” 0

polynomial  $\overset{\Delta}{\underset{0}{}}$  size  $\overset{\Delta}{\underset{5}{}}$  threshold  $\overset{\Delta}{\underset{0}{}}$  circuits  $\overset{\Delta}{\underset{15}{}}$

- An assignment to the boundary indicator variables uniquely determines a parse tree for the string
- We use Gibbs sampling over these boundary indicators instead of sampling parse trees
  - $\Rightarrow$  avoids cubic time complexity of PCFG parsing



# Boundary sampling algorithm for topical collocation models

- Because of the isomorphism between adaptor grammar parses and boundary indicator variable assignments, we can sample parses by sampling boundary indicator variable values
- Gibbs sampling algorithm for boundary indicator variables:  
repeat forever:
  - ▶ pick a random boundary indicator variable
  - ▶ compute relative probabilities of all parses corresponding to possible values of variable
    - most of parse tree is fixed  $\Rightarrow$  strictly local computation
  - ▶ sample a new value for boundary indicator variable according to these relative probabilities
- Basically same as Griffiths et al. (2004) Gibbs sampler for LDA, except for the “null topics”



## Speeding inference for topical collocation models

- Because our sampler is so similar to standard LDA sampler, we can use most of the implementation tricks developed for LDA
- Document  $\rightarrow$  topic and topic  $\rightarrow$  word distributions are sparse  
 $\Rightarrow$  use sparse sampling techniques of Yao et al. (2009) that divide topic probabilities into three “buckets”:
  - ▶ *Smoothing only* bucket: base distribution
  - ▶ *Document topic* bucket: non-zero count document-topic pairs
  - ▶ *Topic word* bucket: non-zero count topic-word pairs
- We *parallelise our inference algorithm* by *generalising the multi-threaded algorithm used in Distributed LDA* (Newman et al., 2009)
  - ▶ we improve their algorithm by *parallelising the reduction operation*

# Outline

Introduction

Probabilistic context-free grammars

Topic models as PCFGs

Adaptor grammars: a non-parametric extension of PCFGs

Segmentation with adaptor grammars

Finding topical collocations with adaptor grammars

Efficient implementation with boundary indicator sampling

**Experimental evaluation**

Conclusions and future work

# Overview of experiments

- We evaluate our model in four ways:
  - ▶ *Document classification*: evaluates how well topics are assigned to documents
  - ▶ *Topic coherence*: evaluates how well topics are assigned to words
  - ▶ *Information retrieval*: evaluates how well topics are assigned to both documents and words
  - ▶ *Efficiency*: measures how fast an implementation is
- We compare the Topical Collocation Model (TCM) to the following models:
  - ▶ LDA (Mallet implementation)
  - ▶ Pipeline Approach (PA) (Lau et al., 2013)
  - ▶ The LDA collocation model (LDACOL) (Griffiths et al., 2007)
  - ▶ Topic N-gram model (TNG) (Wang et al., 2007)
  - ▶ The Adaptor Grammar topical collocation model (AG-colloc) (Johnson, 2010)

Only the first two models can be run on larger data sets.

## Document classification and information retrieval on small corpora

Task	Classification accuracy	IR MAP
Dataset	MReview	SJMN-2k
Mallet-LDA	71.30	18.85
LDACOL	71.75	19.03
TNG	71.40	19.06
PA	72.74	19.16
AG-colloc	<b>73.15</b>	<b>19.37</b>
Non-sparse TCM	<b>73.14</b>	<b>19.30</b>
Sparse TCM	<b>73.13</b>	<b>19.31</b>

- The *movie review* (**MReviews**) corpus (Pang and Lee, 2012) consists of 1,000 positive and 1,000 negative movie reviews
- The *San Jose Mercury News* (**SJMN-2k**) corpus consists of 2,000 news articles
- All non-boldface scores are significantly different ( $p < 0.05$ ) to best

## Classification accuracy on larger corpora

	Mallet-LDA	PA	TCM
Politics	<b>89.1</b>	<b>89.2</b>	<b>89.2</b>
Comp	86.3	87.4	87.9
Sci	92.0	93.2	93.4
Sports	91.6	91.7	92.6
Reuters-21578	97.3	<b>97.5</b>	<b>97.6</b>

- The *Politics*, *Comp*, *Sci* and *Sports* are subsets of the 20 Newsgroups corpus with 4,891, 3,952, 1,993 and 2,625 documents respectively
- The *Reuters-21578* corpus has 21,578 Reuters news stories
- Evaluation procedure:
  - ▶ find document → topic assignments for each model and corpus
  - ▶ randomly split corpus into train (80%) and test (20%)
  - ▶ train SVM to predict document label

## Information retrieval on larger corpora

	Mallet-LDA	PA	TCM
SJMN	20.7	20.9	<b>21.2</b>
AP News	24.0	24.5	<b>24.8</b>

- The *SJMN* corpus has 90,257 documents
- The *AP News* corpus has 242,918 documents
- Experimental procedure:
  - ▶ use the Wei and Croft (2006) information retrieval system, where the topic model is used (together with a unigram language model) to predict the probability of the query given the document
  - ▶ for the collocation models, the query is retokenised using collocations
  - ▶ we report Mean Averaged Precision (MAP) scores

## Topic coherence evaluation

Models	$p(w t)$	$p(t w)$
Mallet-LDA	71.9	73.2
PA	72.8	76.7
TCM	<b>73.2</b>	<b>79.7</b>

- The *intrusion detection* task detects how well Mechanical Turkers can spot “intruders” in lists of topical words (Chang et al., 2009)
  - ▶ train models on the San Jose Mercury News corpus
  - ▶ select 10 words or collocations that maximise  $p(w|t)$  or  $p(t|w)$
  - ▶ randomly select a high-probability word or collocation from another topic
  - ▶ measure the accuracy with which the Turkers spot the intruder

## Running time per iteration

Dataset	MReview		SJMN-2k	
Number of Topics	100	800	100	800
AG-colloc	84.9	1305	37.5	692
Non-sparse TCM	13.8	233	6.6	85.7
Sparse TCM	0.28	0.35	0.14	0.2

- The non-sparse TCM sampler performs each iteration about *6 times faster* than the adaptor grammar sampler
  - ▶ but blocked samplers (e.g., the adaptor grammar sampler) often need fewer iterations than pointwise samplers (e.g., the TCM sampler)
- The *sparse sampler* is more than 50 times faster!



# Evaluating the parallelisation speedup

- Experiments on a machine with 80 Xeon E7-4850 processors (2.0GHz) and 96 GB memory.

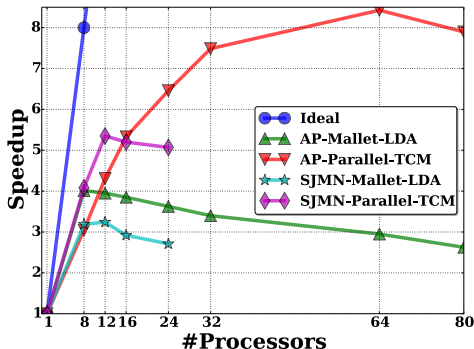


Figure: Plot of speedup in running time for the Mallet-LDA and our TCM.

# Outline

Introduction

Probabilistic context-free grammars

Topic models as PCFGs

Adaptor grammars: a non-parametric extension of PCFGs

Segmentation with adaptor grammars

Finding topical collocations with adaptor grammars

Efficient implementation with boundary indicator sampling

Experimental evaluation

Conclusions and future work

# Conclusions

- Grammars can encode topic models and a wide range of generalisations of them
  - ▶ The *topical collocation model* jointly identifies topics and collocations
- By re-expressing the models in terms of *boundary indicator variables* we can derive a fast, parallelisable Gibbs sampler for the Topical Collocation Model (TCM)
  - ▶ we have also used boundary indicator sampling in *document segmentation* and *phonology induction* models
- The TCM performs well on *document classification*, *information retrieval* and *topic coherence* evaluations.
- The *sparse sampler* significantly speeds inference for topical collocations

## Future work

- Can we exploit sparsity more generally in the adaptor grammar sampler?
  - ▶ the adaptor grammar sampler uses *block sampling*, which samples an entire parse at a time, rather than the *point-wise sampling* used in LDA and here
- Investigate other structural sensitivity in topical collocations
  - ▶ Johnson (2010) uses adaptor grammars to learn and classify named entities
  - ▶ perhaps topical collocations also have an asymmetric structure?
- Learn and exploit latent feature representations for words and collocations