Topics in Natural Language Processing

Shay Cohen

Institute for Language, Cognition and Computation

University of Edinburgh

Lecture 4

Last class - Bayesian analysis

"posterior" "likelihood" "prior"

$$p(\theta|w) = \frac{p(w|\theta)\,p(\theta)}{p(w)}$$

"evidence"

$$p(w) = \int_{\theta} p(w|\theta)\,p(\theta)\,d\theta$$

Conjugacy of prior and likelihood

$$p(\theta) \propto \theta^{\alpha}(1-\theta)^{\beta} \qquad p(w|\theta) = \theta^{I(w)}(1-\theta)^{(1-I(w))}$$

Beta dist.

Prior is "hyperparametrised". What is the posterior?

$$p(\theta|w) \propto \theta^{a+\alpha}(1-\theta)^{b+\beta}$$

where a is # argh

b is # blah

$$p(\theta) \sim \text{Beta}(\alpha, \beta) \Longrightarrow p(\theta|w) \sim \text{Beta}(\alpha + a, \beta + b)$$

Conjugacy of prior to a likelihood

We are given a prior family and a likelihood family.

Each probability distribution in the likelihood family is parametrised by a parameter in the prior family

We say that they are conjugate to each other if the posterior of a likelihood for a prior in the family stays a member of the prior family

“It all stays in the family...”

Conjugacy – always useful?

Trivial non-useful example of conjugacy

$P = \{$ all distributions over $\Theta\}$

posterior is always in P

Conjugacy – always useful?

Another trivial non-useful example of conjugacy

choose some space $\mathcal{H}$

$$\mathcal{P} = \{ p(\theta) \mid p(\theta) = \begin{cases} 1 & \text{if } \theta = \theta_2 \\ 0 & \text{else} \end{cases} \}$$

Your posterior is going to be

Conjugacy: summary

Conjugacy is useful when:

- The prior is not too poor
- It is easy to calculate the posterior hyperparameters

Minimum Description Length and MAP

What is $-\log_2 p(\theta|w_1, \ldots, w_n)$?

Minimum Description Length and MAP

What is $-\log_2 p(\theta|w_1, \ldots, w_n)$?

What is $-\log_2 p(\theta)$?

Minimum Description Length and MAP

What is $-\log_2 p(\theta|w_1,\ldots,w_n)$?

bits that is required using a "good" code to encode θ given $w_1 \ldots w_n$

What is $-\log_2 p(\theta)$?

bits that are required using a good code to encode θ a priori

What is $-\log_2 p(w_1,\ldots,w_n|\theta)$?

bits that is required to encode the data if we think θ is "correct"

$$\theta^* = \arg\max_\theta p(\theta|w_1 \ldots w_n) = \arg\max_\theta \frac{p(\theta)\, p(w_1 \ldots w_n|\theta)}{\cancel{p(w_1 \ldots w_n)}}$$

take lg

$$= \arg\max_\theta -\log_2 p(\theta) - \log_2 p(w_1 \ldots w_n|\theta)$$

Minimum Description Length and MAP

What is $-\log_2 p(\theta|w_1, \ldots, w_n)$?

What is $-\log_2 p(\theta)$?

What is $-\log_2 p(w_1, \ldots, w_n|\theta)$?

MAP: $\theta^* = \arg\max_\theta \log_2 p(\theta) + \log_2 p(w_1, \ldots, w_n|\theta)$

Encoding θ^* requires separately:

- Encoding the hypothesis according to the prior
- Encoding the data according to the hypothesis

That's the "minimum description length" criterion

Learning from Incomplete Data

Just as a side note:

- Bayesian analysis in NLP is especially uesful for learning from incomplete data
- It presents a flexible framework for introducting latent variables into a model
- Posterior inference becomes more complex

Summary

Bayesian analysis:

- Only uses Bayes' rule to do inference
- Posterior is a *distribution* over parameters
- Can summarise the posterior, e.g. MAP, to get a point estimate
- Need to be careful about choice of prior
- Especially important with small amounts of data
- MAP has a connection to minimum description length (MDL)

Grammars

Grammar: a formal system of rules that govern the production of a language Language can be formal language or natural language

Grammars

Grammar: a formal system of rules that govern the production of a language Language can be formal language or natural language

Probabilistic Grammar: augment a set of rules with probabilities to get distributions over derivations and strings

Context-Free Grammars

Context-free grammars, quick reminder:

Nonterminals N

Terminals T

Rules $A \rightarrow B\ C$ $S \rightarrow NP\ VP$

Start symbol S

But there are many other formalisms, all rely on the idea of "rewriting" with a "context-free" backbone

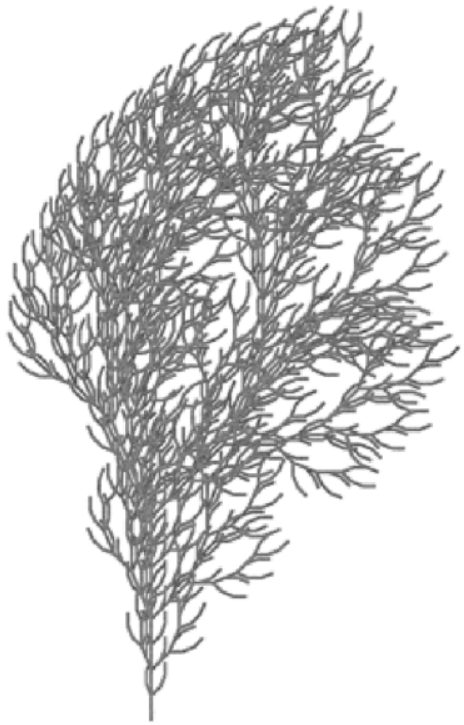
Why Do We Need Grammars?

- They implement the idea of compositionality very elegantly
- They are often interpretable – you can understand why rules are there and what they represent in language or otherwise
- They often have relatively efficient *inference* and *parsing* algorithms to find most likely derivations and structures

Why Do We Need Grammars?

- They implement the idea of compositionality very elegantly
- They are often interpretable – you can understand why rules are there and what they represent in language or otherwise
- They often have relatively efficient *inference* and *parsing* algorithms to find most likely derivations and structures

Originally formal grammars came as an attempt to formalise the rules behind natural language, now they are ubiquitous in computer science and there is an area of research called “formal language theory” that studies them.



Also in computer vision...

Important Notions

Let G be a grammar:

- String language $L(G)$
- Derivation, derivation language $D(G)$
- Structure language, $S(G)$

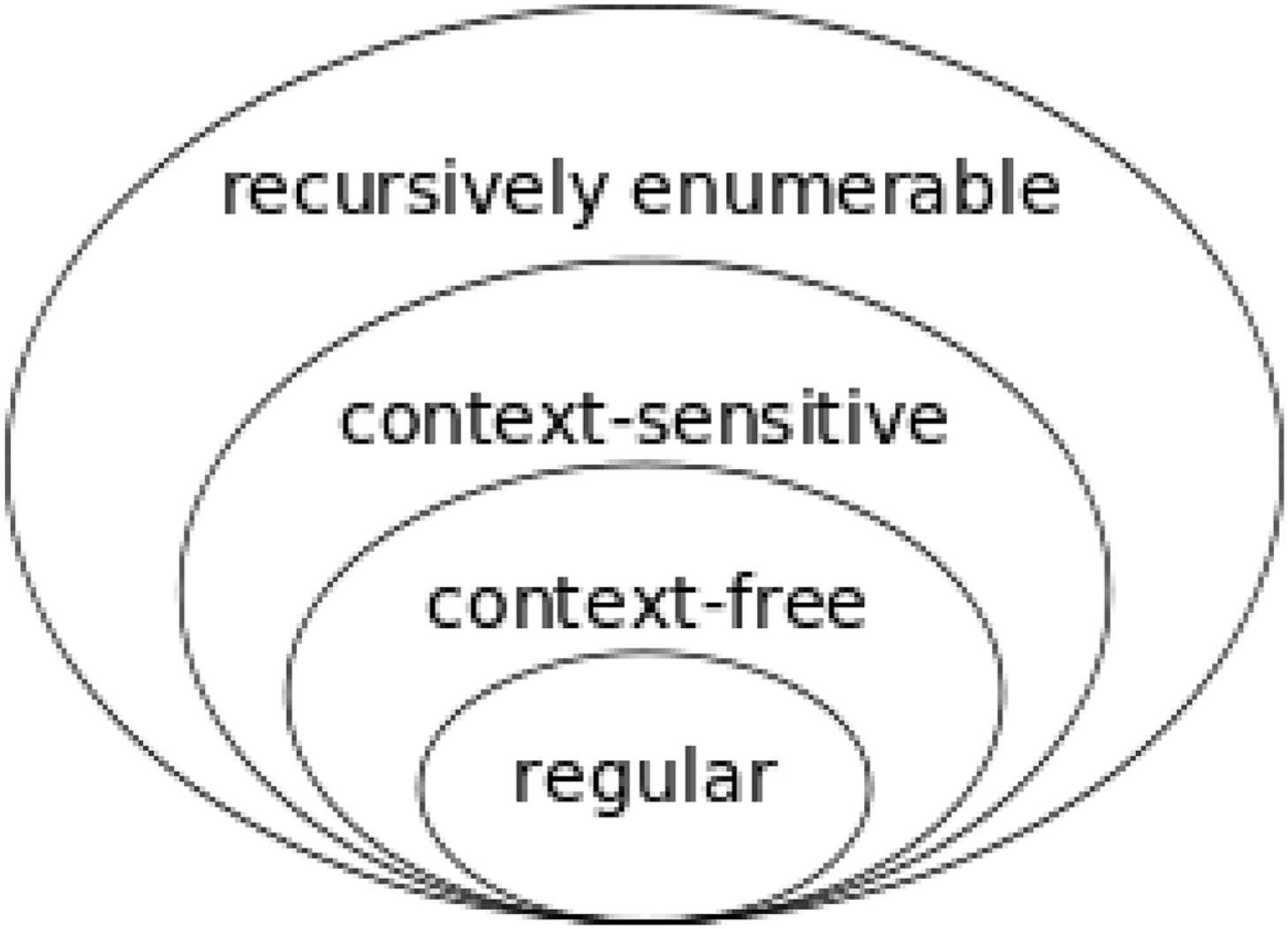
T^*

$L(G) \subseteq T^*$

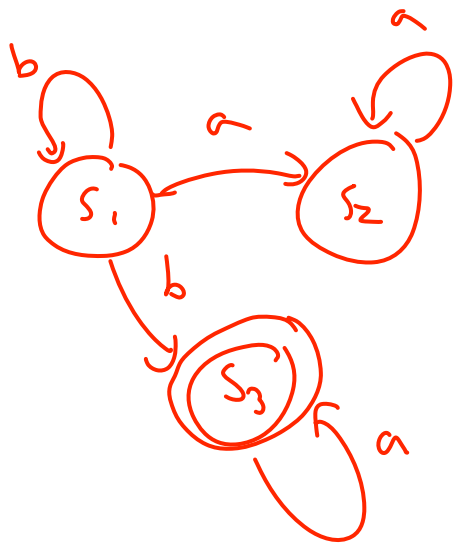
$T = \{th, dg\}$

$T^* = \{the\ th, d\ d, t\ t \ldots\}$

The Chomsky Hierarchy



Regular Languages as Context-Free Languages



NTs to be states

Terminals to be the alphabet

$S_1 \rightarrow a\ S_2$

$S_1 \rightarrow b\ S_1$

$S_3 \rightarrow \varepsilon$

Regular Languages as Context-Free Languages

The states become nonterminals. We have rules of the form $\text{InState} \to x\,\text{OutState}$ for every edge in the FSA that transitions from InState to OutState emitting x.

This means we get a *linear* grammar – a grammar with a single nonterminal on the righthand side

Is Language Regular?

Center embedding implies language is not regular:

- This is the rat that ate the malt.

Is Language Regular?

Center embedding implies language is not regular:

- This is the rat that ate the malt.
- This is the malt that the rat ate.

- This is the cat that bit the rat that ate the malt.
- This is the malt that the rat that the cat bit ate.

- This is the dog that chased the cat that bit rat that ate the malt.
- This is the malt that the rat that the cat that the dog chased bit ate.

Is Language Regular?

Let $A = \{a_1, \ldots, a_m\}$ be the set of nouns and $B = \{b_1, \ldots, b_m\}$ the set of matching verbs.

Assume English was regular. Intersect it with the regular language **This is the malt (that the** A**)*** B^*.

Then we get **This is the malt (that the** A**)**^{n} B^n. Clearly not regular.

But the intersection of any regular language with another is also regular. Hence English cannot be regular.

Is Language Context-Free?

Let G be a grammar.

$T(G) =$ tree language

$L(G) =$ string language

- Dutch - there are structures in Dutch which do not appear in any $T(G)$ for any G context-free grammar

Is Language Context-Free?

Let G be a grammar.

$T(G) =$

$L(G) =$

- Dutch - there are structures in Dutch which do not appear in any $T(G)$ for any G context-free grammar
- Swiss-German - there are strings in Swiss-German which do not appear in any $L(G)$ (and hence in any $T(G)$) for any G CFG

The constructions are similar to demonstrate that. Swiss-German uses case markers.

Non-projectivity

mer	em	Hans	s	huus	hälfed	aastriiche
we		Hans	the	house	helped	paint

... and “we have wanted to let the children help Hans paint the house.”

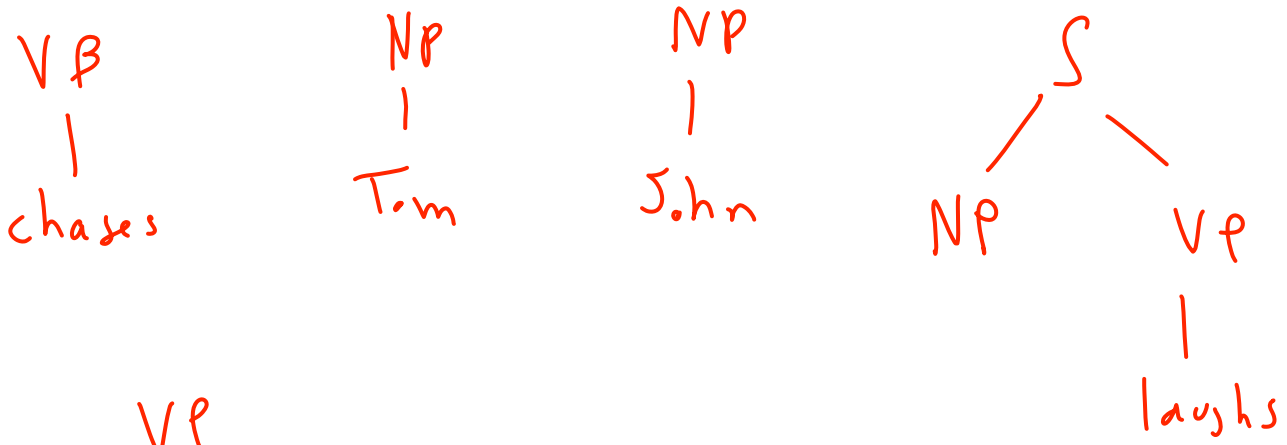
Intersect Swiss-German with a regular language and you get a non-context-free language.

But intersection of context-free languages with regular languages is context-free.

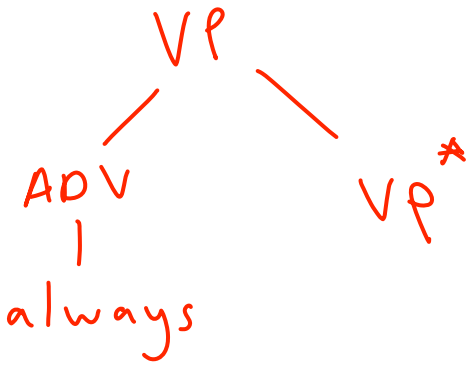
More about this in the assignment!

Tree Adjoining Grammars

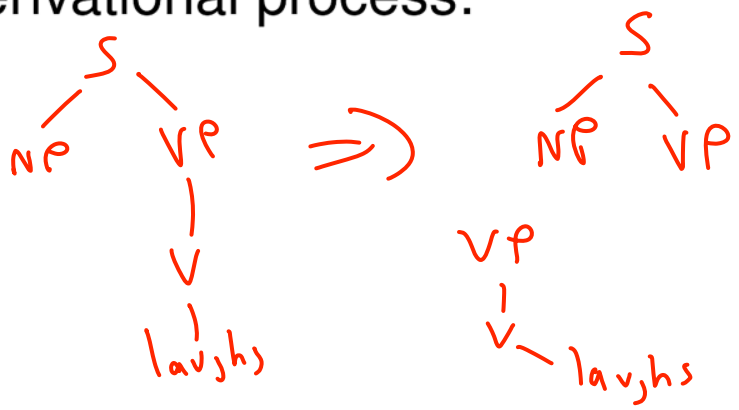
Initial trees:

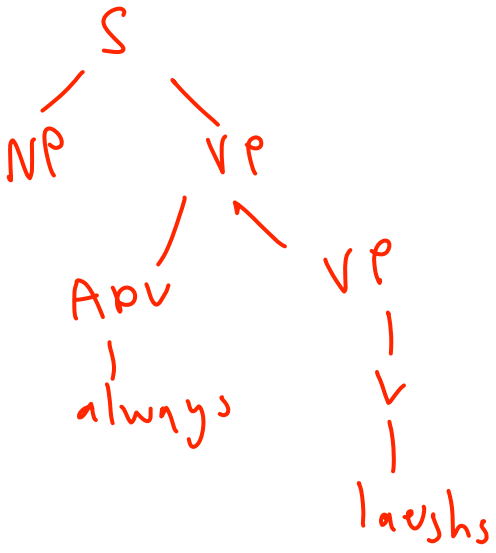


Auxiliary trees:



Derivational process:





Tree Adjoining Grammars

- The initial trees and auxiliary trees together are also called “elementary trees.”
- Can have a constraint on the nodes where adjunction is allowed and where it is not allowed.

Tree Adjoining Grammars

Quick question: is $\{ww^R \mid w \in \Sigma^*\}$ a context-free language where w^R is the reverse string of w?

$S \rightarrow aSb$ $\quad ab$

$S \rightarrow \varepsilon$

$S \rightarrow aSa$ $\quad abba$

$S \rightarrow bSb$ $\quad aabbaa$

$S \rightarrow \varepsilon$

Tree Adjoining Grammars

Quick question: is $\{ww \mid w \in \Sigma^*\}$ a context-free language?