Natural Language Models and Interfaces

BSc Artificial Intelligence

Lecturer: Wilker Aziz
Institute for Logic, Language, and Computation

2020, week 3

Problems with n-gram LMs

Estimation

- number of parameters grows exponentially in n

$$O(v^n)$$

- Zipf's law tells us most words will be extremely rare
  n-grams are even sparser

What can we do beyond smoothing and interpolation?

Problems with n-gram LMs

Estimation

- number of parameters grows exponentially in n

$$O(v^n)$$

- Zipf's law tells us most words will be extremely rare
  n-grams are even sparser

What can we do beyond smoothing and interpolation?
Design better models! :D

NLMI

Parts of speech

Hidden Markov Models

Evaluation

Generalisations in language

We can organise words into classes

- semantic criteria: what does the word refer to?
  nouns often refer to 'people', 'places' or 'things'
- formal criteria: what form does the word have?
  -ly makes an adverb out of an adjective
  -tion makes a noun out of a verb
- distributional criteria: in what contexts can the word occur?
  adjectives precede nouns

Adapted from S. Goldwater, S. Cohen, T. Deoskar

Criteria for classifying words

	Semantically	Formally	Distributionally
Nouns	refer to things, concepts	-ness, -tion, -ity, -ance	After determiners, possessives
Verbs	refer to actions, states	-ate, -ize	infinitives: to jump, to learn
Adjectives	properties of nouns	-al, -ble	appear before nouns
Adverbs	properties of actions	-ly	next to verbs, beginning of sentence

Importance of formal and distributional criteria

Often in text, we come across unknown words

And, as in uffish thought he stood,
The Jabberwock, with eyes of flame,
Came whiffling through the tulgey wood,
And burbled as it came!

Formal and distributional criteria help one recognise which class an unknown word belongs to:

Those zorls you splarded were malgy

Parts of Speech

- Open class words (or content words)
  - nouns, verbs, adjectives, adverbs
  - mostly content-bearing
    they refer to objects, actions, and features in the world
  - open class, since there is no limit to what these words are
    new ones are added all the time (email, website, selfie)
- Closed class words (or function words)
  - pronouns, determiners, prepositions, connectives, ...
  - there is a limited number of these
  - mostly functional: to tie the concepts of a sentence together

Adapted from T. Deoskar

But how many parts of speech

- Both linguistic and practical considerations
- Corpus annotators decide. Distinguish between
  - proper nouns (names) and common nouns ?
  - past and present tense verbs?
  - auxiliary and main verbs?

English POS tag sets

Brown corpus (87 tags)

- one of the earliest large corpora collected for computational linguistics (1960s)
- balanced corpus: different genres (fiction, news, academic, editorial, etc)

Penn Treebank corpus (45 tags)

- first large corpus annotated with POS and full syntactic trees (1992)
- possibly the most-used corpus in NLP
- originally, just text from the Wall Street Journal (WSJ)

Universal POS tags

- Simplify the set of tags to lowest common denominator across languages
- Map existing annotations onto universal tags

VBD, VBN, VB, VBG, VBP $\rightarrow$ VERB

- Allows interoperability of systems across languages
- Promoted by Google and others

Universal POS tags

NOUN (nouns)
VERB (verbs)
ADJ (adjectives)
ADV (adverbs)
PRON (pronouns)
DET (determiners and articles)
ADP (prepositions and postpositions)
NUM (numerals)
CONJ (conjunctions)
PRT (particles)
?.? (punctuation marks)
X (anything else, such as abbreviations or foreign words)

Example of POS tagged data

The/*DT* grand/*JJ* jury/*NN* commented/*VBD* on/*IN* a/*DT* number/*NN* of/*IN* other/*JJ* topics/*NNS* ./*.*

There/*EX* was/*VBD* still/*JJ* lemonade/*NN* in/*IN* the/*DT* bottle/*NN* ./*.*

NLMI

Parts of speech

Hidden Markov Models

Evaluation

How does any of that help modelling language?

Linguistic generalisation abstracts away from surface form

- knowing X_i took on an adjective should increase the chance that X_{i+1} takes on a noun
  - regardless of the adjective and of the noun

Role of conditional independence

Suppose A and B take on values in $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$

- how many parameters to represent P_{AB}?

Role of conditional independence

Suppose A and B take on values in $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$

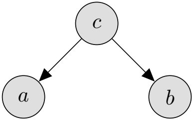
- how many parameters to represent P_{AB}? $O(n \times m)$

Role of conditional independence

Suppose A and B take on values in $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$

- how many parameters to represent P_{AB}? $O(n \times m)$

We can make A and B conditionally independent given C



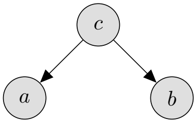
$$P_{AB}(a, b) = \sum_{c=1}^{t} P_{ABC}(a, b, c)$$

Role of conditional independence

Suppose A and B take on values in $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$

- how many parameters to represent P_{AB}? $O(n \times m)$

We can make A and B conditionally independent given C



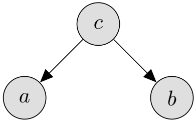
$$
\begin{aligned}
P_{AB}(a, b) &= \sum_{c=1}^{t} P_{ABC}(a, b, c) \\
&= \sum_{c=1}^{t} P_C(c) P_{AB|C}(a, b|c)
\end{aligned}
$$

Role of conditional independence

Suppose A and B take on values in $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$

- how many parameters to represent P_{AB}? $O(n \times m)$

We can make A and B conditionally independent given C



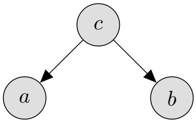
$$\begin{aligned} P_{AB}(a, b) &= \sum_{c=1}^{t} P_{ABC}(a, b, c) \\ &= \sum_{c=1}^{t} P_C(c) P_{AB|C}(a, b|c) \\ &= \sum_{c=1}^{t} P_C(c) P_{A|C}(a|c) P_{B|C}(b|c) \end{aligned}$$

Role of conditional independence

Suppose A and B take on values in $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$

- how many parameters to represent P_{AB}? $O(n \times m)$

We can make A and B conditionally independent given C



$$\begin{aligned} P_{AB}(a, b) &= \sum_{c=1}^{t} P_{ABC}(a, b, c) \\ &= \sum_{c=1}^{t} P_C(c) P_{AB|C}(a, b|c) \\ &= \sum_{c=1}^{t} P_C(c) P_{A|C}(a|c) P_{B|C}(b|c) \end{aligned}$$

and still *marginally dependent*

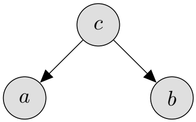
- with how many parameters?

Role of conditional independence

Suppose A and B take on values in $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$

- how many parameters to represent P_{AB}? $O(n \times m)$

We can make A and B conditionally independent given C



$$\begin{aligned} P_{AB}(a, b) &= \sum_{c=1}^{t} P_{ABC}(a, b, c) \\ &= \sum_{c=1}^{t} P_C(c) P_{AB|C}(a, b|c) \\ &= \sum_{c=1}^{t} P_C(c) P_{A|C}(a|c) P_{B|C}(b|c) \end{aligned}$$

and still *marginally dependent*

- with how many parameters? $O(t + t \times n + t \times m)$

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story

We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story

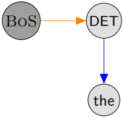
We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story



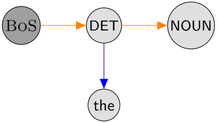
We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story



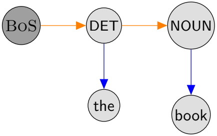
We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story



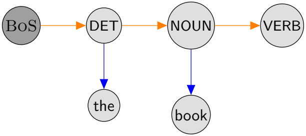
We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story



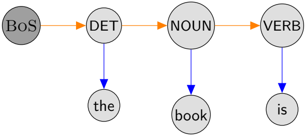
We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story



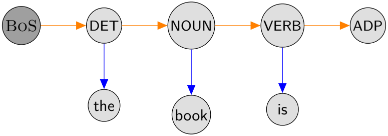
We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story



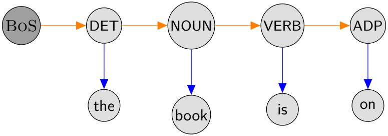
We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story



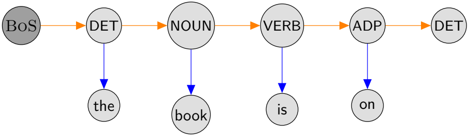
We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story



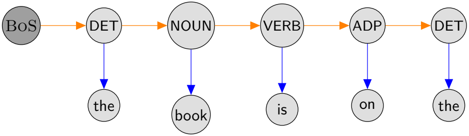
We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story



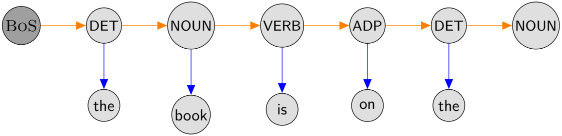
We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story



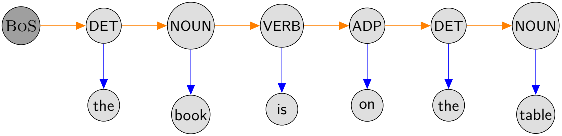
We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story



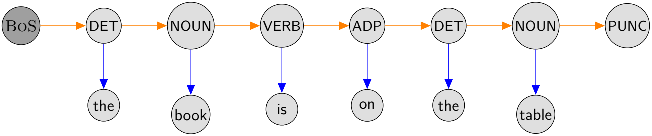
We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story



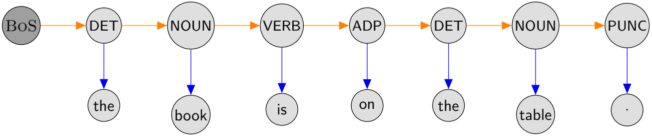
We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story



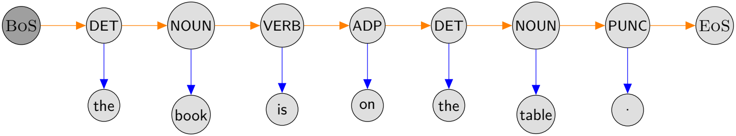
We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story



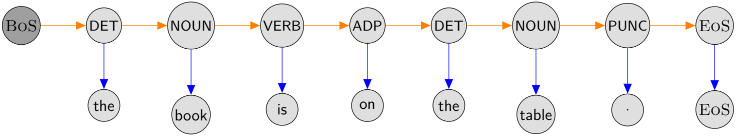
We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story



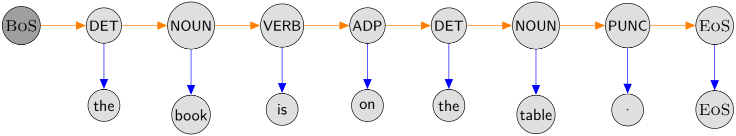
We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data: illustration

Joint observations

the/*DET* book/*NOUN* is/*VERB* on/*ADP* the/*DET* table/*NOUN* ./*PUNC*

Generative story



Joint probability

$$
\begin{aligned}
&P_{C|C_{\text{prev}}}(\text{DET}|\text{BoS})P_{X|C}(\text{the}|\text{DET}) \\
&\times P_{C|C_{\text{prev}}}(\text{NOUN}|\text{DET})P_{X|C}(\text{book}|\text{NOUN}) \\
&\times \ldots \\
&\times P_{C|C_{\text{prev}}}(\text{PUNC}|\text{NOUN})P_{X|C}(.|\text{PUNC}) \\
&\times P_{C|C_{\text{prev}}}(\text{EoS}|\text{PUNC})P_{X|C}(\text{EoS}|\text{EoS})
\end{aligned}
$$

We pad the tag sequence with a BoS symbol. We pad both sequences with a EoS symbol.

Modelling POS-tagged data

Random variables

- X is a random word taking on values in $\mathcal{X} = \{1, \ldots, v\}$
- C is a random tag taking on values in $\mathcal{C} = \{1, \ldots, t\}$

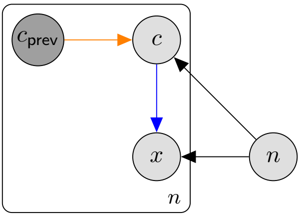
Modelling POS-tagged data

Random variables

- X is a random word taking on values in $\mathcal{X} = \{1, \ldots, v\}$
- C is a random tag taking on values in $\mathcal{C} = \{1, \ldots, t\}$

Generative story

1. $N \sim P_N$
2. For $i = 1, \ldots, n$
   - $C_i | c_{i-1} \sim P_{C|C_{\text{prev}}}$
   - $X_i | c_i \sim P_{X|C}$



Modelling POS-tagged data

Random variables

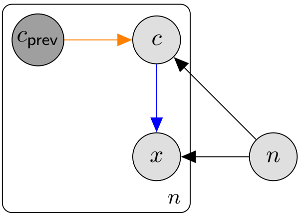
- X is a random word taking on values in $\mathcal{X} = \{1, \ldots, v\}$
- C is a random tag taking on values in $\mathcal{C} = \{1, \ldots, t\}$

Generative story

1. $N \sim P_N$
2. For $i = 1, \ldots, n$
   - $C_i | c_{i-1} \sim P_{C|C_{\text{prev}}}$
   - $X_i | c_i \sim P_{X|C}$

Parameterisation

- Transition distribution
  $C|C_{\text{prev}} = p \sim \mathrm{Cat}(\lambda_1^{(p)}, \ldots, \lambda_t^{(p)})$
- Emission distribution
  $X|C = c \sim \mathrm{Cat}(\theta_1^{(c)}, \ldots, \theta_v^{(c)})$



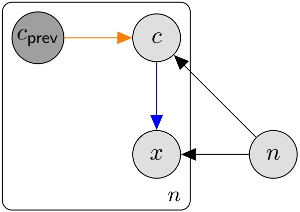
Modelling POS-tagged data

Random variables

- X is a random word taking on values in $\mathcal{X} = \{1, \ldots, v\}$
- C is a random tag taking on values in $\mathcal{C} = \{1, \ldots, t\}$

Generative story

1. $N \sim P_N$
2. For $i = 1, \ldots, n$
   - $C_i | c_{i-1} \sim P_{C|C_{\text{prev}}}$
   - $X_i | c_i \sim P_{X|C}$



Parameterisation

- Transition distribution
  $C|C_{\text{prev}} = p \sim \text{Cat}(\lambda_1^{(p)}, \ldots, \lambda_t^{(p)})$
- Emission distribution
  $X|C = c \sim \text{Cat}(\theta_1^{(c)}, \ldots, \theta_v^{(c)})$

How many parameters?

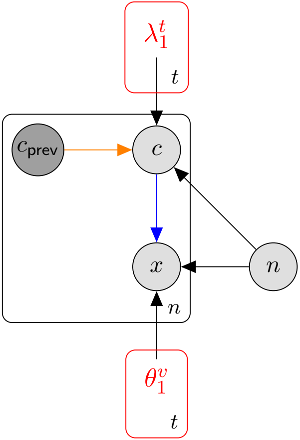
Modelling POS-tagged data

Random variables

- X is a random word taking on values in $\mathcal{X} = \{1, \ldots, v\}$
- C is a random tag taking on values in $\mathcal{C} = \{1, \ldots, t\}$

Generative story

1. $N \sim P_N$
2. For $i = 1, \ldots, n$
   - $C_i | c_{i-1} \sim P_{C|C_{\text{prev}}}$
   - $X_i | c_i \sim P_{X|C}$



Parameterisation

- Transition distribution
  $C|C_{\text{prev}} = p \sim \text{Cat}(\lambda_1^{(p)}, \ldots, \lambda_t^{(p)})$
- Emission distribution
  $X|C = c \sim \text{Cat}(\theta_1^{(c)}, \ldots, \theta_v^{(c)})$

How many parameters? $O(t^2 + tv)$

Maximum likelihood estimation for labelled data

Suppose a data set of m observations

$$\left(\underbrace{\langle x_1^{(k)}, \ldots, x_{n_k}^{(k)} \rangle}_{\text{sentence}}, \underbrace{\langle c_1^{(k)}, \ldots, c_{n_k}^{(k)} \rangle}_{\textit{tag sequence}} \right)_{k=1}^{m}$$

MLE solution

- Transition distribution

Maximum likelihood estimation for labelled data

Suppose a data set of m observations

$$\left(\underbrace{\langle x_1^{(k)}, \ldots, x_{n_k}^{(k)} \rangle}_{\text{sentence}}, \underbrace{\langle c_1^{(k)}, \ldots, c_{n_k}^{(k)} \rangle}_{\textit{tag sequence}} \right)_{k=1}^{m}$$

MLE solution

- Transition distribution

$$\lambda_c^{(p)} = \frac{\sum_{k=1}^{m} \sum_{i=1}^{n_k} [p = c_{i-1}^{(k)} \land c = c_i^{(k)}]}{\sum_{k=1}^{m} \sum_{i=1}^{n_k} [p = c_{i-1}]} = \frac{\text{count}_{C_{\mathsf{prev}} C}(p, c)}{\text{count}_{C_{\mathsf{prev}}}(p)}$$

Maximum likelihood estimation for labelled data

Suppose a data set of m observations

$$\left(\underbrace{\langle x_1^{(k)}, \ldots, x_{n_k}^{(k)} \rangle}_{\text{sentence}}, \underbrace{\langle c_1^{(k)}, \ldots, c_{n_k}^{(k)} \rangle}_{\textit{tag sequence}} \right)_{k=1}^{m}$$

MLE solution

- Transition distribution

$$\lambda_c^{(p)} = \frac{\sum_{k=1}^{m} \sum_{i=1}^{n_k} [p = c_{i-1}^{(k)} \land c = c_i^{(k)}]}{\sum_{k=1}^{m} \sum_{i=1}^{n_k} [p = c_{i-1}]} = \frac{\text{count}_{C_{\text{prev}} C}(p, c)}{\text{count}_{C_{\text{prev}}}(p)}$$

- Emission distribution

Maximum likelihood estimation for labelled data

Suppose a data set of m observations

$$\left(\underbrace{\langle x_1^{(k)}, \ldots, x_{n_k}^{(k)} \rangle}_{\text{sentence}}, \underbrace{\langle c_1^{(k)}, \ldots, c_{n_k}^{(k)} \rangle}_{\text{tag sequence}} \right)_{k=1}^{m}$$

MLE solution

- Transition distribution

$$\lambda_c^{(p)} = \frac{\sum_{k=1}^{m} \sum_{i=1}^{n_k} [p = c_{i-1}^{(k)} \wedge c = c_i^{(k)}]}{\sum_{k=1}^{m} \sum_{i=1}^{n_k} [p = c_{i-1}]} = \frac{\text{count}_{C_{\text{prev}} C}(p, c)}{\text{count}_{C_{\text{prev}}}(p)}$$

- Emission distribution

$$\theta_x^{(c)} = \frac{\sum_{k=1}^{m} \sum_{i=1}^{n_k} [c = c_i^{(k)} \wedge x = x_i^{(k)}]}{\sum_{k=1}^{m} \sum_{i=1}^{n_k} [c = c_i]} = \frac{\text{count}_{CX}(c, x)}{\text{count}_{C}(c)}$$

NLMI

Parts of speech

Hidden Markov Models

Evaluation

Evaluate HMM POS model

Extrinsically *given labelled test set*

- compare best possible tag sequence to tagged test set
- accuracy of tag prediction

Best tag sequence

Given a sentence, we want the most likely tag sequence

$$\operatorname*{argmax}_{c_1^n} \; P(c_1^n | x_1^n) \qquad \text{posterior}$$

Best tag sequence

Given a sentence, we want the most likely tag sequence

$$\operatorname*{argmax}_{c_1^n} \; P(c_1^n | x_1^n)$$ posterior

$$= \operatorname*{argmax}_{c_1^n} \; \frac{P(x_1^n, c_1^n)}{P(x_1^n)}$$ conditional probability

Best tag sequence

Given a sentence, we want the most likely tag sequence

$$\operatorname*{argmax}_{c_1^n} \; P(c_1^n | x_1^n) \qquad \text{posterior}$$

$$= \operatorname*{argmax}_{c_1^n} \; \frac{P(x_1^n, c_1^n)}{P(x_1^n)} \qquad \text{conditional probability}$$

$$= \operatorname*{argmax}_{c_1^n} \; P(x_1^n, c_1^n) \qquad \text{proportionality}$$

Best tag sequence

Given a sentence, we want the most likely tag sequence

$$\operatorname*{argmax}_{c_1^n} \; P(c_1^n|x_1^n) \qquad \text{posterior}$$

$$= \operatorname*{argmax}_{c_1^n} \; \frac{P(x_1^n, c_1^n)}{P(x_1^n)} \qquad \text{conditional probability}$$

$$= \operatorname*{argmax}_{c_1^n} \; P(x_1^n, c_1^n) \qquad \text{proportionality}$$

$$= \operatorname*{argmax}_{c_1^n} \; \prod_{i=1}^{n} P_{C|C_{\mathsf{prev}}}(c_i|c_{i-1}) P_{X|C}(x_i|c_i) \qquad \text{factorisation}$$

Best tag sequence

Given a sentence, we want the most likely tag sequence

$$\operatorname*{argmax}_{c_1^n} \; P(c_1^n|x_1^n) \qquad \text{posterior}$$

$$= \operatorname*{argmax}_{c_1^n} \; \frac{P(x_1^n, c_1^n)}{P(x_1^n)} \qquad \text{conditional probability}$$

$$= \operatorname*{argmax}_{c_1^n} \; P(x_1^n, c_1^n) \qquad \text{proportionality}$$

$$= \operatorname*{argmax}_{c_1^n} \; \prod_{i=1}^{n} P_{C|C_{\mathsf{prev}}}(c_i|c_{i-1}) P_{X|C}(x_i|c_i) \qquad \text{factorisation}$$

$$= \operatorname*{argmax}_{c_1^n} \; \prod_{i=1}^{n} \lambda_{c_i}^{(c_{i-1})} \theta_{x_i}^{(c_i)} \qquad \text{Categorical pmf}$$

Best tag sequence

Given a sentence, we want the most likely tag sequence

$$\underset{c_1^n}{\text{argmax}}\ P(c_1^n|x_1^n) \qquad \text{posterior}$$

$$= \underset{c_1^n}{\text{argmax}}\ \frac{P(x_1^n, c_1^n)}{P(x_1^n)} \qquad \text{conditional probability}$$

$$= \underset{c_1^n}{\text{argmax}}\ P(x_1^n, c_1^n) \qquad \text{proportionality}$$

$$= \underset{c_1^n}{\text{argmax}}\ \prod_{i=1}^{n} P_{C|C_{\text{prev}}}(c_i|c_{i-1}) P_{X|C}(x_i|c_i) \qquad \text{factorisation}$$

$$= \underset{c_1^n}{\text{argmax}}\ \prod_{i=1}^{n} \lambda_{c_i}^{(c_{i-1})} \theta_{x_i}^{(c_i)} \qquad \text{Categorical pmf}$$

$$= \underset{c_1^n}{\text{argmax}}\ \sum_{i=1}^{n} \log \lambda_{c_i}^{(c_{i-1})} + \log \theta_{x_i}^{(c_i)} \qquad \text{monotonicity}$$

Space of analyses

Example:

observation $x_1^3 \circ \langle \textsc{EoS} \rangle$

tagset $\{1, 2\} \cup \{0, 4\}$ for $\textsc{BoS}$ and $\textsc{EoS}$ respectively

C_0	C_1	C_2	C_3	C_4	$P(x_1^n, c_1^n \mid n)$

Space of analyses

Example:

observation $x_1^3 \circ \langle \text{EoS} \rangle$

tagset $\{1, 2\} \cup \{0, 4\}$ for BoS and EoS respectively

C_0	C_1	C_2	C_3	C_4	$P(x_1^n, c_1^n \mid n)$
BoS	1	1	1	EoS	
BoS	1	1	2	EoS	
BoS	1	2	1	EoS	
BoS	1	2	2	EoS	
BoS	2	1	1	EoS	
BoS	2	1	2	EoS	
BoS	2	2	1	EoS	
BoS	2	2	2	EoS	

Space of analyses

Example:

observation $x_1^3 \circ \langle \text{EoS} \rangle$

tagset $\{1, 2\} \cup \{0, 4\}$ for BoS and EoS respectively

C_0	C_1	C_2	C_3	C_4	$P(x_1^n, c_1^n \mid n)$
BoS	1	1	1	EoS	$\lambda_1^{(0)} \times \theta_{x_1}^{(1)} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)} \times \lambda_1^{(1)} \theta_{x_3}^{(1)} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}$
BoS	1	1	2	EoS	$\lambda_1^{(0)} \times \theta_{x_1}^{(1)} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)} \times \lambda_2^{(1)} \theta_{x_3}^{(2)} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)}$
BoS	1	2	1	EoS	
BoS	1	2	2	EoS	
BoS	2	1	1	EoS	
BoS	2	1	2	EoS	
BoS	2	2	1	EoS	
BoS	2	2	2	EoS	

Space of analyses

Example:

observation $x_1^3 \circ \langle \textsc{EoS} \rangle$

tagset $\{1, 2\} \cup \{0, 4\}$ for $\textsc{BoS}$ and $\textsc{EoS}$ respectively

C_0	C_1	C_2	C_3	C_4	$P(x_1^n, c_1^n \mid n)$
BoS	1	1	1	EoS	$\lambda_1^{(0)} \times \theta_{x_1}^{(1)} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)} \times \lambda_1^{(1)} \theta_{x_3}^{(1)} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}$
BoS	1	1	2	EoS	$\lambda_1^{(0)} \times \theta_{x_1}^{(1)} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)} \times \lambda_2^{(1)} \theta_{x_3}^{(2)} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)}$
BoS	1	2	1	EoS	$\ldots$
BoS	1	2	2	EoS	
BoS	2	1	1	EoS	
BoS	2	1	2	EoS	
BoS	2	2	1	EoS	
BoS	2	2	2	EoS	

Strategy: enumerate analyses, score them, sort them, pick the best

Space of analyses

Example:

observation $x_1^3 \circ \langle \text{EoS} \rangle$

tagset $\{1, 2\} \cup \{0, 4\}$ for BoS and EoS respectively

C_0	C_1	C_2	C_3	C_4	$P(x_1^n, c_1^n \mid n)$
BoS	1	1	1	EoS	$\lambda_1^{(0)} \times \theta_{x_1}^{(1)} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)} \times \lambda_1^{(1)} \theta_{x_3}^{(1)} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}$
BoS	1	1	2	EoS	$\lambda_1^{(0)} \times \theta_{x_1}^{(1)} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)} \times \lambda_2^{(1)} \theta_{x_3}^{(2)} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)}$
BoS	1	2	1	EoS	...
BoS	1	2	2	EoS	
BoS	2	1	1	EoS	
BoS	2	1	2	EoS	
BoS	2	2	1	EoS	
BoS	2	2	2	EoS	

Strategy: enumerate analyses, score them, sort them, pick the best

Is there a problem here?

Space of analyses

Example:

observation $x_1^3 \circ \langle \text{EoS} \rangle$

tagset $\{1, 2\} \cup \{0, 4\}$ for BoS and EoS respectively

C_0	C_1	C_2	C_3	C_4	$P(x_1^n, c_1^n \| n)$
BoS	1	1	1	EoS	$\lambda_1^{(0)} \times \theta_{x_1}^{(1)} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)} \times \lambda_1^{(1)} \theta_{x_3}^{(1)} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}$
BoS	1	1	2	EoS	$\lambda_1^{(0)} \times \theta_{x_1}^{(1)} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)} \times \lambda_2^{(1)} \theta_{x_3}^{(2)} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)}$
BoS	1	2	1	EoS	...
BoS	1	2	2	EoS	
BoS	2	1	1	EoS	
BoS	2	1	2	EoS	
BoS	2	2	1	EoS	
BoS	2	2	2	EoS	

Strategy: enumerate analyses, score them, sort them, pick the best

Is there a problem here? Yes! There are $O(t^n)$ analyses!

Dynamic programming

There are $O(t^n)$ possible tag sequences, but

- small changes only affect small parts of the scoring function

C_0	C_1	C_2	C_3	C_4	$P(x_1^n, c_1^n \mid n)$
BoS	1	1	1	EoS	$\lambda_1^{(0)} \times \theta_{x_1}^{(1)} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)} \times \lambda_1^{(1)} \theta_{x_3}^{(1)} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}$
BoS	1	1	2	EoS	$\lambda_1^{(0)} \times \theta_{x_1}^{(1)} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)} \times \lambda_2^{(1)} \theta_{x_3}^{(2)} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)}$

Dynamic programming

There are $O(t^n)$ possible tag sequences, but

- small changes only affect small parts of the scoring function

C_0	C_1	C_2	C_3	C_4	$P(x_1^n, c_1^n \mid n)$
BoS	1	1	1	EoS	$\lambda_1^{(0)} \times \theta_{x_1}^{(1)} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)} \times \lambda_1^{(1)} \theta_{x_3}^{(1)} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}$
BoS	1	1	2	EoS	$\lambda_1^{(0)} \times \theta_{x_1}^{(1)} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)} \times \lambda_2^{(1)} \theta_{x_3}^{(2)} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)}$

- divide and conquer: identify independent subproblems and reuse partial solutions

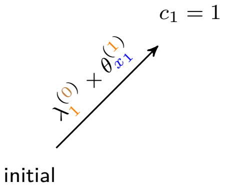
Pack solutions in a directed graph

Example: observation x_1^3 tagset $\{1, 2\}$

initial

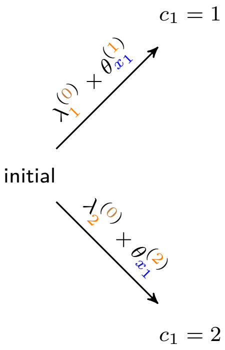
Pack solutions in a directed graph

Example: observation x_1^3 tagset $\{1, 2\}$



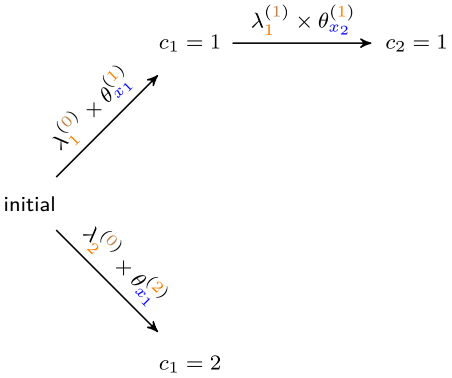
Pack solutions in a directed graph

Example: observation x_1^3 tagset $\{1, 2\}$



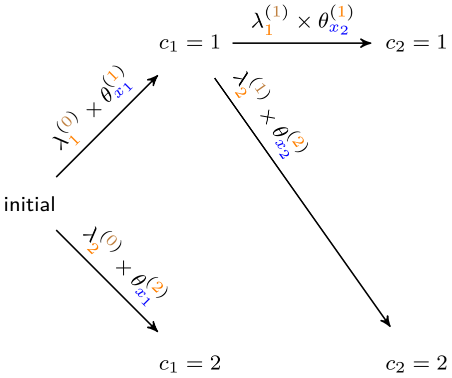
Pack solutions in a directed graph

Example: observation x_1^3 tagset $\{1, 2\}$



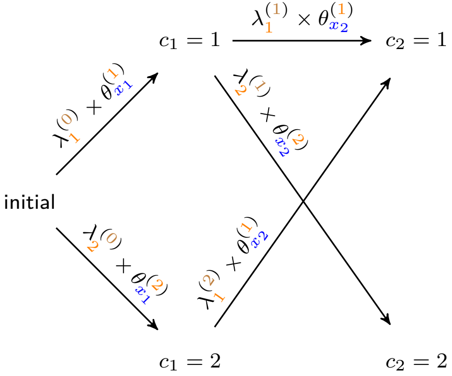
Pack solutions in a directed graph

Example: observation x_1^3 tagset $\{1, 2\}$



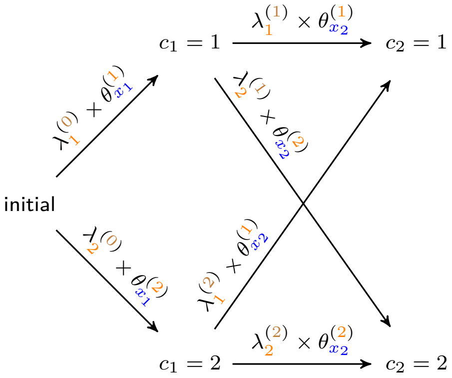
Pack solutions in a directed graph

Example: observation x_1^3 tagset $\{1, 2\}$



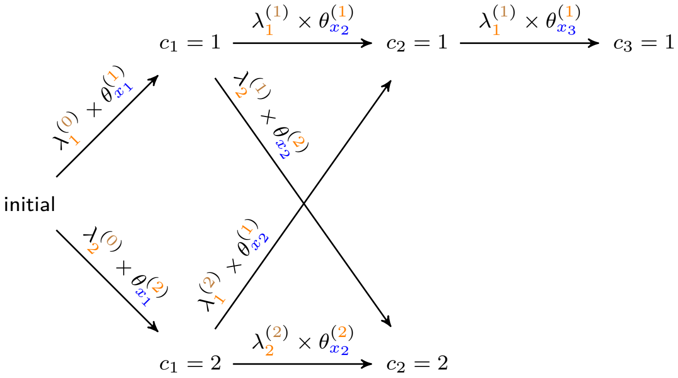
Pack solutions in a directed graph

Example: observation x_1^3 tagset $\{1, 2\}$



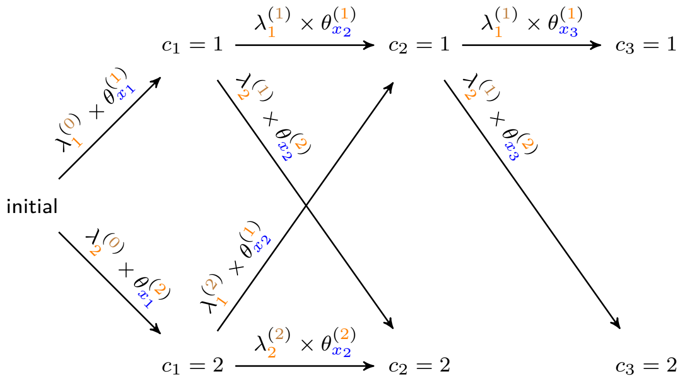
Pack solutions in a directed graph

Example: observation x_1^3 tagset $\{1, 2\}$



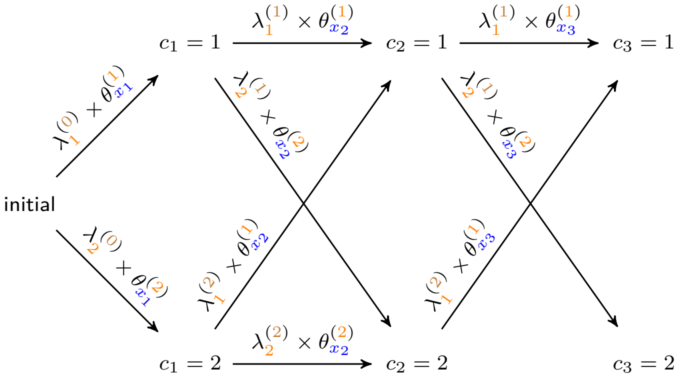
Pack solutions in a directed graph

Example: observation x_1^3 tagset $\{1, 2\}$



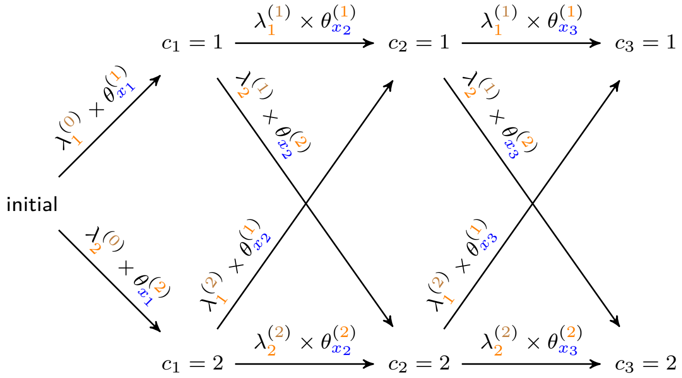
Pack solutions in a directed graph

Example: observation x_1^3 tagset $\{1, 2\}$



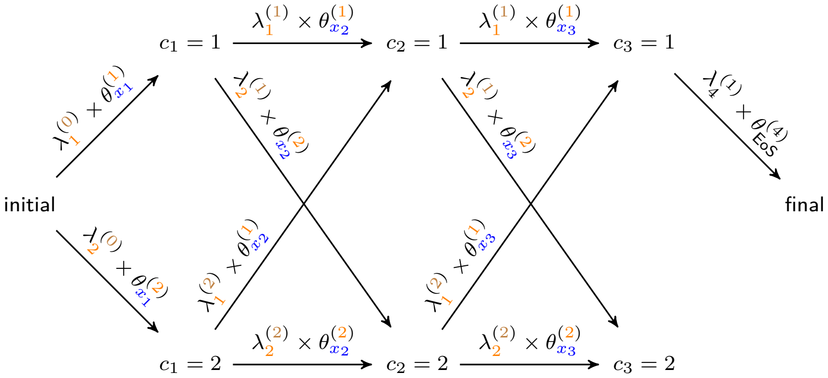
Pack solutions in a directed graph

Example: observation x_1^3 tagset $\{1, 2\}$



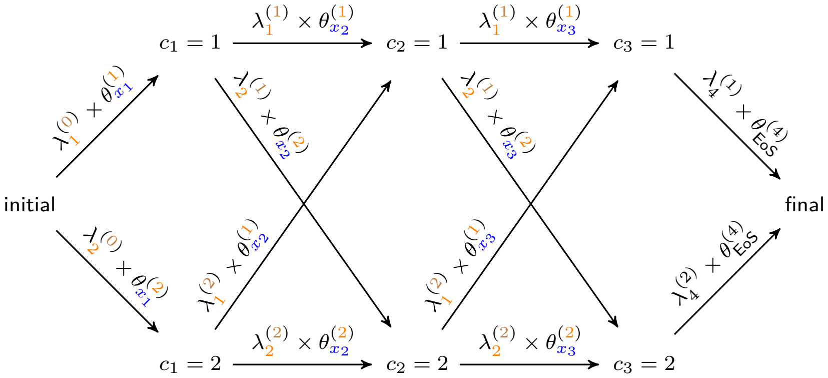
Pack solutions in a directed graph

Example: observation x_1^3 tagset $\{1, 2\}$



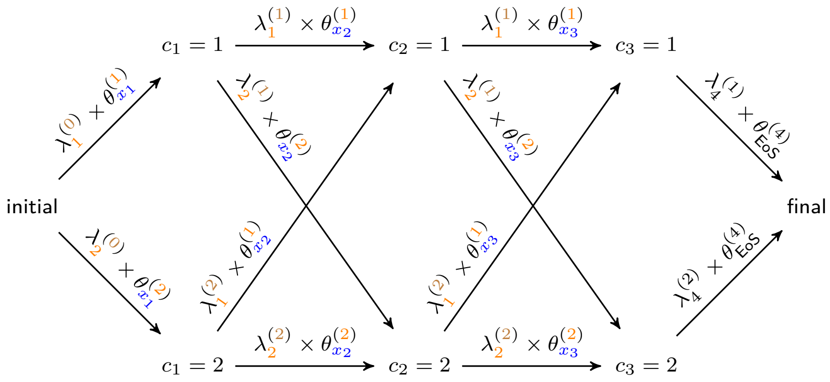
Pack solutions in a directed graph

Example: observation x_1^3 tagset $\{1, 2\}$



Pack solutions in a directed graph

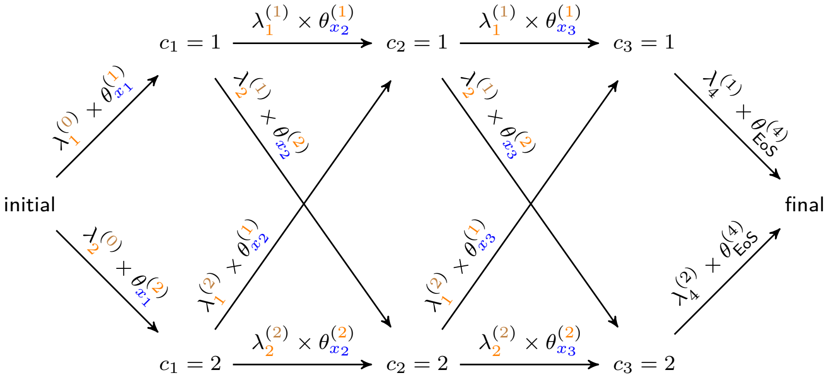
Example: observation x_1^3 tagset $\{1, 2\}$



Compact representation:

Pack solutions in a directed graph

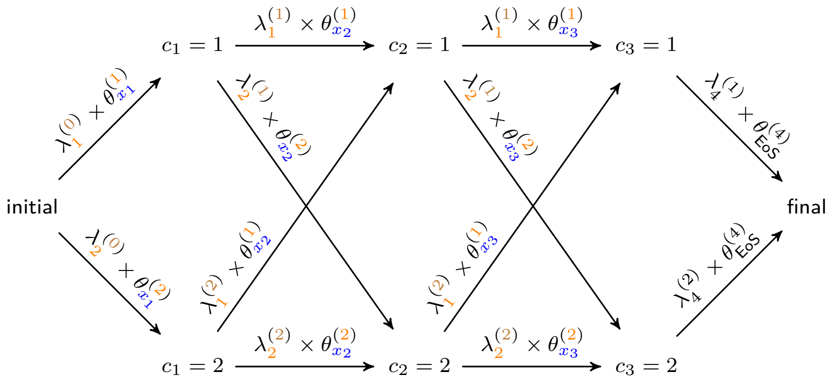
Example: observation x_1^3 tagset $\{1, 2\}$



Compact representation: $O(n \times t)$ nodes and $O(n \times t^2)$ edges

Pack solutions in a directed graph

Example: observation x_1^3 tagset $\{1, 2\}$



Compact representation: $O(n \times t)$ nodes and $O(n \times t^2)$ edges
Best sequence: path with highest probability

Viterbi algorithm

Enumeration is intractable:

Dynamic programming Recursion It's numerically convenient to compute $\log \alpha$ instead!

Viterbi algorithm

Enumeration is intractable: $O(t^n)$ paths

Dynamic programming Recursion It's numerically convenient to compute $\log \alpha$ instead!

Viterbi algorithm

Enumeration is intractable: $O(t^n)$ paths

- but the scoring function factorises

Dynamic programming Recursion It's numerically convenient to compute $\log \alpha$ instead!

Viterbi algorithm

Enumeration is intractable: $O(t^n)$ paths

- but the scoring function factorises

Dynamic programming

- identify optimal substructure and overlapping subproblems
- the ith decision only affects the score of the $(i+1)$th decision or conversely, the ith decision is only a function of the $(i-1)$th decision

Dynamic programming Recursion It's numerically convenient to compute $\log \alpha$ instead!

Viterbi algorithm

Enumeration is intractable: $O(t^n)$ paths

- but the scoring function factorises

Dynamic programming

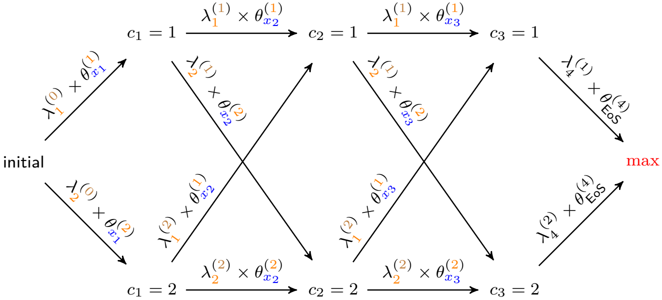
- identify optimal substructure and overlapping subproblems
- the ith decision only affects the score of the $(i+1)$th decision or conversely, the ith decision is only a function of the $(i-1)$th decision

Viterbi recursion

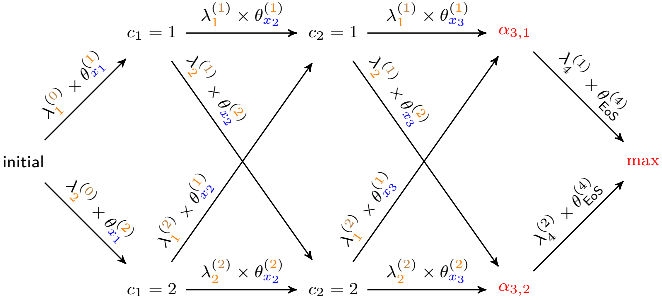
$$\alpha(i, j) = \begin{cases} 1 & \text{if } i = 0 \\ \max\limits_{p \in \{1, \ldots, t\}} \alpha(i-1, p)\lambda_j^{(p)}\theta_{x_i}^{(j)} & \text{otherwise} \end{cases}$$

$\alpha(i, j)$ is the maximum value of any sequence $\langle C_1, \ldots, C_i = j \rangle$

Dynamic programming Recursion It's numerically convenient to compute $\log \alpha$ instead!

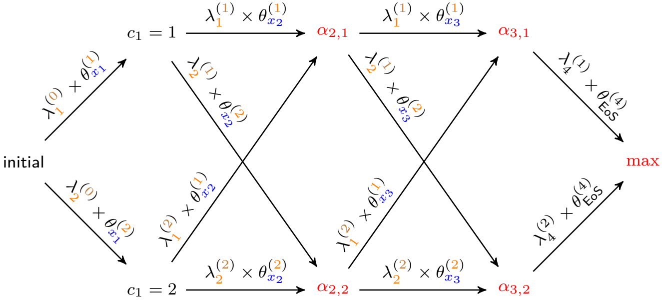


We want to know the maximum of the joint distribution



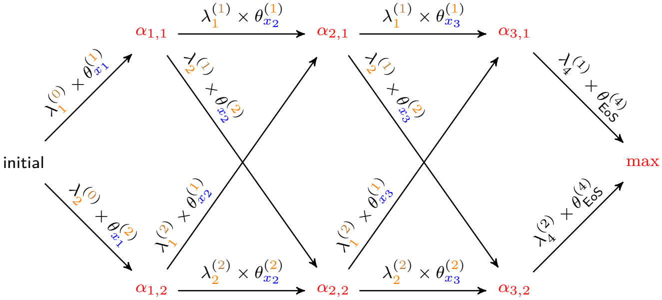
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)})$

The maximum complete assignment depends on the maximum for $\langle c_1, c_2, c_3 = 1\rangle$ and $\langle c_1, c_2, c_3 = 2\rangle$



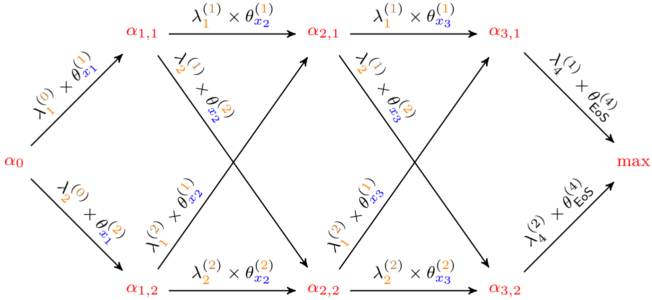
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$

Similarly, the maximum for $\langle c_1, c_2, c_3 = 1\rangle$ depends on $\langle c_1, c_2 = 1\rangle$ and $\langle c_1, c_2 = 2\rangle$



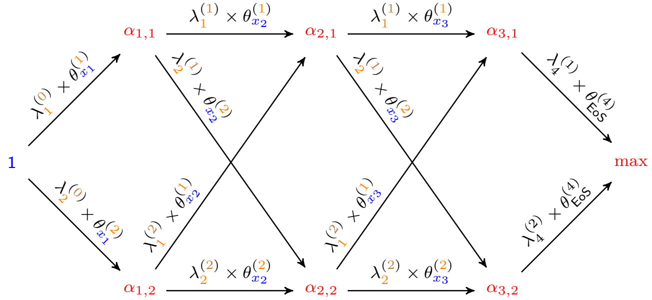
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\mathsf{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\mathsf{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x_2}^{(1)})$

Again, the maximum for $\langle c_1, c_2 = 1 \rangle$ depends on $\langle c_1 = 1 \rangle$ and $\langle c_1 = 2 \rangle$



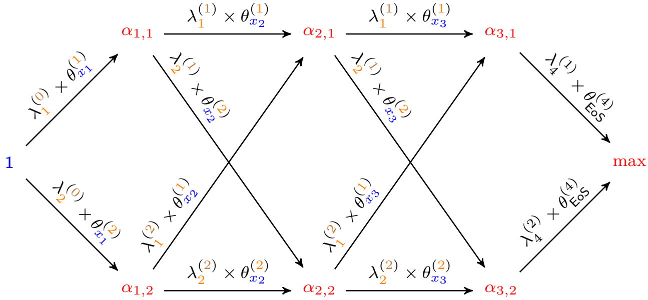
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x_2}^{(1)})$
- $\alpha_{1,1} = \alpha_0 \times \lambda_1^{(0)} \times \theta_{x_1}^{(1)}$

Finally, the maximum for $\langle c_1 = 1 \rangle$ depends on tagging x_1 with $c_1 = 1$ from the initial state



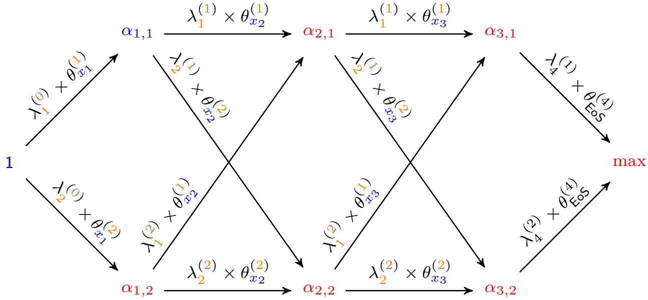
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x_2}^{(1)})$
- $\alpha_{1,1} = \alpha_0 \times \lambda_1^{(0)} \times \theta_{x_1}^{(1)}$
- $\alpha_0 = 1$

Which by convention has probability 1



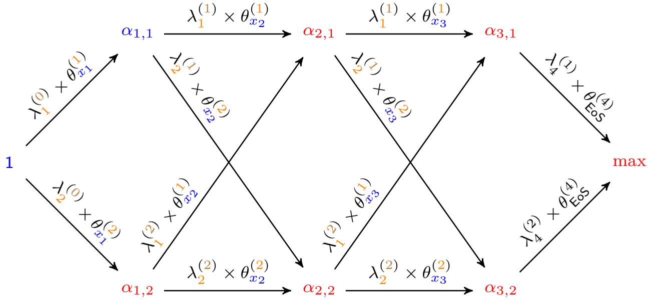
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x_2}^{(1)})$
- $\alpha_{1,1} = \alpha_0 \times \lambda_1^{(0)} \times \theta_{x_1}^{(1)}$
- $\alpha_0 = 1$

We backtrack with the value 1



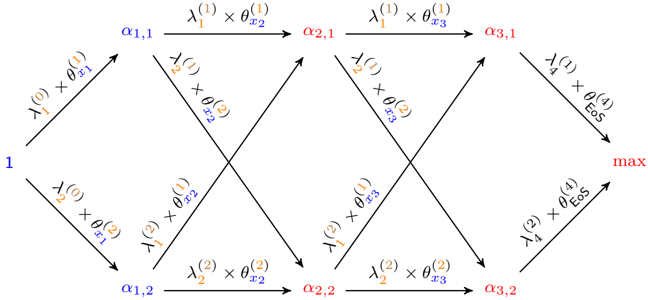
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x_2}^{(1)})$
- $\alpha_{1,1} = \alpha_0 \times \lambda_1^{(0)} \times \theta_{x_1}^{(1)}$
- $\alpha_0 = 1$

And obtain the maximum for $\langle c_1 = 1 \rangle$



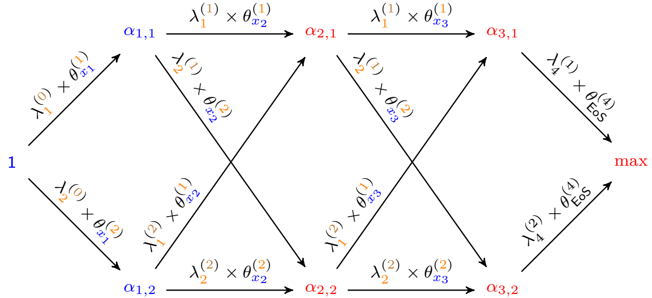
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x_2}^{(1)})$
- $\alpha_{1,1} = \alpha_0 \times \lambda_1^{(0)} \times \theta_{x_1}^{(1)}$
- $\alpha_0 = 1$

Again we backtrack substituting the value just computed



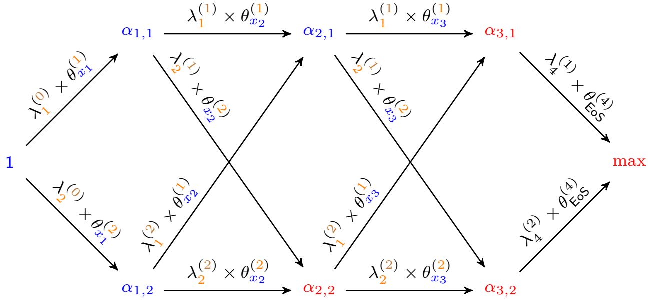
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x_2}^{(1)})$
- $\alpha_{1,1} = \alpha_0 \times \lambda_1^{(0)} \times \theta_{x_1}^{(1)}$
- $\alpha_0 = 1$
- $\alpha_{1,2} = \alpha_0 \times \lambda_2^{(0)} \times \theta_{x_1}^{(2)}$

And proceed to compute the maximum for $\langle c_1 = 2 \rangle$ — note that we already know α_0



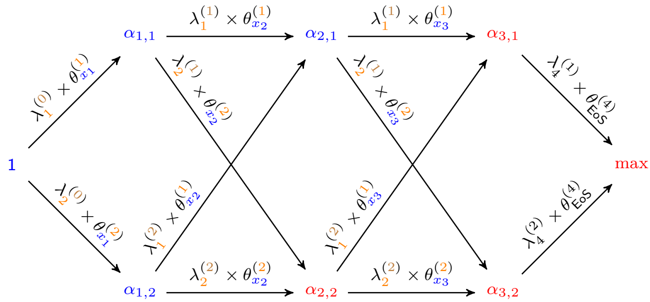
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\mathsf{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\mathsf{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x_2}^{(1)})$
- $\alpha_{1,1} = \alpha_0 \times \lambda_1^{(0)} \times \theta_{x_1}^{(1)}$
- $\alpha_0 = 1$
- $\alpha_{1,2} = \alpha_0 \times \lambda_2^{(0)} \times \theta_{x_1}^{(2)}$

We backtrack substituting the value just computed



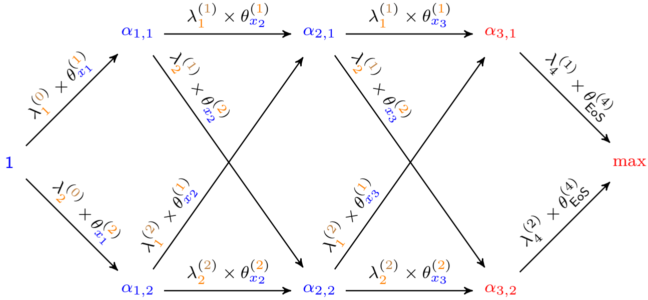
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x_2}^{(1)})$
- $\alpha_{1,1} = \alpha_0 \times \lambda_1^{(0)} \times \theta_{x_1}^{(1)}$
- $\alpha_0 = 1$
- $\alpha_{1,2} = \alpha_0 \times \lambda_2^{(0)} \times \theta_{x_1}^{(2)}$

And now that all relevant quantities are known, we can compute the maximum for $\langle c_1, c_2 = 1 \rangle$



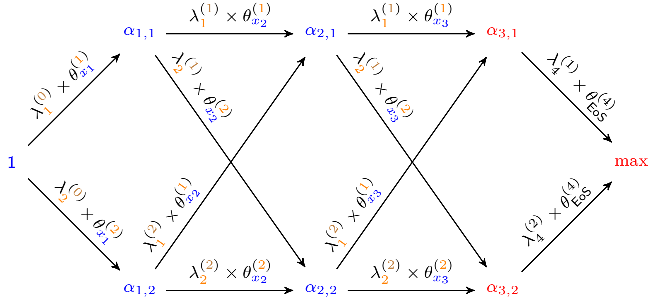
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x_2}^{(1)})$
- $\alpha_{1,1} = \alpha_0 \times \lambda_1^{(0)} \times \theta_{x_1}^{(1)}$
- $\alpha_0 = 1$
- $\alpha_{1,2} = \alpha_0 \times \lambda_2^{(0)} \times \theta_{x_1}^{(2)}$

Again we backtrack substituting the value just computed



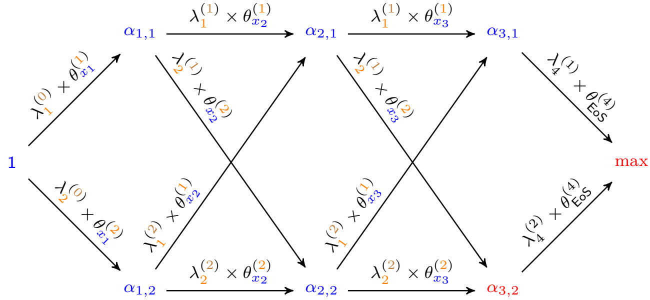
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x_2}^{(1)})$
- $\alpha_{1,1} = \alpha_0 \times \lambda_1^{(0)} \times \theta_{x_1}^{(1)}$
- $\alpha_0 = 1$
- $\alpha_{1,2} = \alpha_0 \times \lambda_2^{(0)} \times \theta_{x_1}^{(2)}$
- $\alpha_{2,2} = \max(\alpha_{1,1} \times \lambda_2^{(1)} \times \theta_{x_2}^{(2)}, \alpha_{1,2} \times \lambda_2^{(2)} \times \theta_{x_2}^{(2)})$

And proceed to compute the maximum for $\langle c_1, c_2 = 2 \rangle$. In this case, all relevant quantities are known



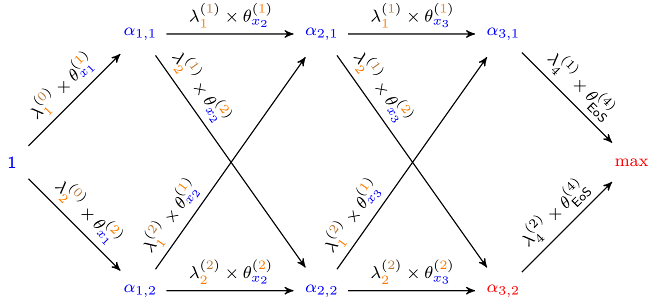
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x_2}^{(1)})$
- $\alpha_{1,1} = \alpha_0 \times \lambda_1^{(0)} \times \theta_{x_1}^{(1)}$
- $\alpha_0 = 1$
- $\alpha_{1,2} = \alpha_0 \times \lambda_2^{(0)} \times \theta_{x_1}^{(2)}$
- $\alpha_{2,2} = \max(\alpha_{1,1} \times \lambda_2^{(1)} \times \theta_{x_2}^{(2)}, \alpha_{1,2} \times \lambda_2^{(2)} \times \theta_{x_2}^{(2)})$

Thus we backtrack substituting the relevant maximum



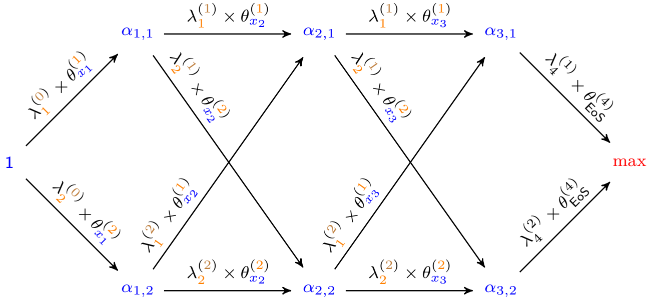
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x_2}^{(1)})$
- $\alpha_{1,1} = \alpha_0 \times \lambda_1^{(0)} \times \theta_{x_1}^{(1)}$
- $\alpha_0 = 1$
- $\alpha_{1,2} = \alpha_0 \times \lambda_2^{(0)} \times \theta_{x_1}^{(2)}$
- $\alpha_{2,2} = \max(\alpha_{1,1} \times \lambda_2^{(1)} \times \theta_{x_2}^{(2)}, \alpha_{1,2} \times \lambda_2^{(2)} \times \theta_{x_2}^{(2)})$

And obtain the maximum for $\langle c_1, c_2, c_3 = 1 \rangle$



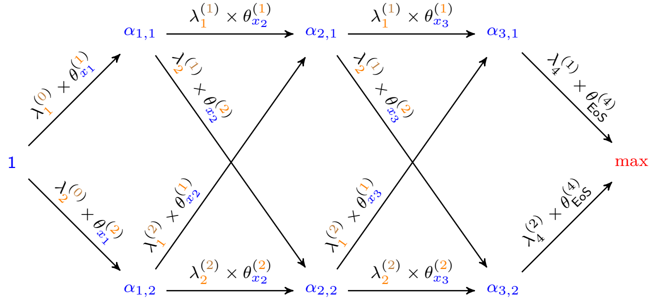
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\mathsf{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\mathsf{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x_2}^{(1)})$
- $\alpha_{1,1} = \alpha_0 \times \lambda_1^{(0)} \times \theta_{x_1}^{(1)}$
- $\alpha_0 = 1$
- $\alpha_{1,2} = \alpha_0 \times \lambda_2^{(0)} \times \theta_{x_1}^{(2)}$
- $\alpha_{2,2} = \max(\alpha_{1,1} \times \lambda_2^{(1)} \times \theta_{x_2}^{(2)}, \alpha_{1,2} \times \lambda_2^{(2)} \times \theta_{x_2}^{(2)})$

We backtrack with that value



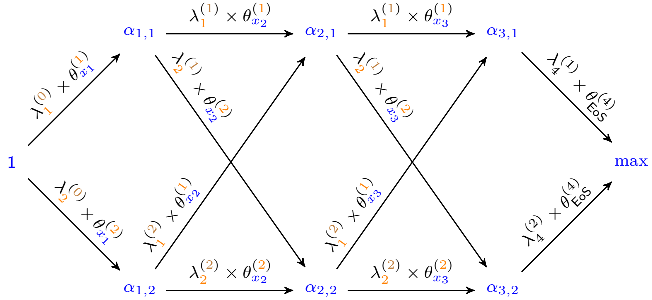
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x_2}^{(1)})$
- $\alpha_{1,1} = \alpha_0 \times \lambda_1^{(0)} \times \theta_{x_1}^{(1)}$
- $\alpha_0 = 1$
- $\alpha_{1,2} = \alpha_0 \times \lambda_2^{(0)} \times \theta_{x_1}^{(2)}$
- $\alpha_{2,2} = \max(\alpha_{1,1} \times \lambda_2^{(1)} \times \theta_{x_2}^{(2)}, \alpha_{1,2} \times \lambda_2^{(2)} \times \theta_{x_2}^{(2)})$
- $\alpha_{3,2} = \max(\alpha_{2,1} \times \lambda_2^{(1)} \times \theta_{x_3}^{(2)}, \alpha_{2,2} \times \lambda_2^{(2)} \times \theta_{x_3}^{(2)})$

And proceed to compute the maximum for $\langle c_1, c_2, c_3 = 2 \rangle$. Again, all necessary quantities are known.



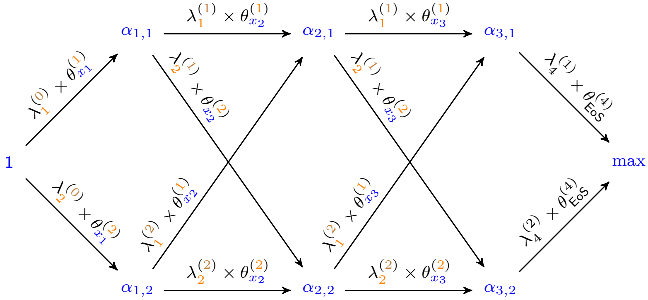
- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x_2}^{(1)})$
- $\alpha_{1,1} = \alpha_0 \times \lambda_1^{(0)} \times \theta_{x_1}^{(1)}$
- $\alpha_0 = 1$
- $\alpha_{1,2} = \alpha_0 \times \lambda_2^{(0)} \times \theta_{x_1}^{(2)}$
- $\alpha_{2,2} = \max(\alpha_{1,1} \times \lambda_2^{(1)} \times \theta_{x_2}^{(2)}, \alpha_{1,2} \times \lambda_2^{(2)} \times \theta_{x_2}^{(2)})$
- $\alpha_{3,2} = \max(\alpha_{2,1} \times \lambda_2^{(1)} \times \theta_{x_3}^{(2)}, \alpha_{2,2} \times \lambda_2^{(2)} \times \theta_{x_3}^{(2)})$

We backtrack the maximum



- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x2}^{(1)})$
- $\alpha_{1,1} = \alpha_0 \times \lambda_1^{(0)} \times \theta_{x1}^{(1)}$
- $\alpha_0 = 1$
- $\alpha_{1,2} = \alpha_0 \times \lambda_2^{(0)} \times \theta_{x1}^{(2)}$
- $\alpha_{2,2} = \max(\alpha_{1,1} \times \lambda_2^{(1)} \times \theta_{x2}^{(2)}, \alpha_{1,2} \times \lambda_2^{(2)} \times \theta_{x2}^{(2)})$
- $\alpha_{3,2} = \max(\alpha_{2,1} \times \lambda_2^{(1)} \times \theta_{x3}^{(2)}, \alpha_{2,2} \times \lambda_2^{(2)} \times \theta_{x3}^{(2)})$

And have the overall maximum!



- $\max(\alpha_{3,1} \times \lambda_4^{(1)} \times \theta_{\text{EoS}}^{(4)}, \alpha_{3,2} \times \lambda_4^{(2)} \times \theta_{\text{EoS}}^{(4)})$
- $\alpha_{3,1} = \max(\alpha_{2,1} \times \lambda_1^{(1)} \times \theta_{x_3}^{(1)}, \alpha_{2,2} \times \lambda_1^{(2)} \times \theta_{x_3}^{(1)})$
- $\alpha_{2,1} = \max(\alpha_{1,1} \times \lambda_1^{(1)} \times \theta_{x_2}^{(1)}, \alpha_{1,2} \times \lambda_1^{(2)} \times \theta_{x_2}^{(1)})$
- $\alpha_{1,1} = \alpha_0 \times \lambda_1^{(0)} \times \theta_{x_1}^{(1)}$
- $\alpha_0 = 1$
- $\alpha_{1,2} = \alpha_0 \times \lambda_2^{(0)} \times \theta_{x_1}^{(2)}$
- $\alpha_{2,2} = \max(\alpha_{1,1} \times \lambda_2^{(1)} \times \theta_{x_2}^{(2)}, \alpha_{1,2} \times \lambda_2^{(2)} \times \theta_{x_2}^{(2)})$
- $\alpha_{3,2} = \max(\alpha_{2,1} \times \lambda_2^{(1)} \times \theta_{x_3}^{(2)}, \alpha_{2,2} \times \lambda_2^{(2)} \times \theta_{x_3}^{(2)})$

Finding an argmax is a simple matter of traversing in reverse direction tracking the best path.

Viterbi implementation

Viterbi recursion

$$\alpha(i, j) = \begin{cases} 1 & \text{if } i = 0 \\ \max_{p \in \{1, \ldots, t\}} \alpha(i-1, p) \lambda_j^{(p)} \theta_{x_i}^{(j)} & \text{otherwise} \end{cases}$$

Implementation without recursion:

- for $i = 1, \ldots, n$
  - for $j = 1, \ldots, t$
    - solve $\alpha(i, j)$ and store its value in cell $\mathtt{V}[i, j]$

Viterbi implementation

Viterbi recursion

$$\alpha(i, j) = \begin{cases} 1 & \text{if } i = 0 \\ \max_{p \in \{1, \ldots, t\}} \alpha(i-1, p) \lambda_j^{(p)} \theta_{x_i}^{(j)} & \text{otherwise} \end{cases}$$

Implementation without recursion:

- for $i = 1, \ldots, n$
  - for $j = 1, \ldots, t$
    - solve $\alpha(i, j)$ and store its value in cell $\mathtt{V}[i, j]$

Complexity

- space:

Viterbi implementation

Viterbi recursion

$$\alpha(i, j) = \begin{cases} 1 & \text{if } i = 0 \\ \max_{p \in \{1,\ldots,t\}} \alpha(i-1, p)\lambda_j^{(p)}\theta_{x_i}^{(j)} & \text{otherwise} \end{cases}$$

Implementation without recursion:

- for $i = 1, \ldots, n$
  - for $j = 1, \ldots, t$
    - solve $\alpha(i, j)$ and store its value in cell $\mathtt{V}[i, j]$

Complexity

- space: $O(n \times t)$ cells in `V`

Viterbi implementation

Viterbi recursion

$$\alpha(i, j) = \begin{cases} 1 & \text{if } i = 0 \\ \max_{p \in \{1, \ldots, t\}} \alpha(i-1, p) \lambda_j^{(p)} \theta_{x_i}^{(j)} & \text{otherwise} \end{cases}$$

Implementation without recursion:

- for $i = 1, \ldots, n$
  - for $j = 1, \ldots, t$
    - solve $\alpha(i, j)$ and store its value in cell $\texttt{V}[i, j]$

Complexity

- space: $O(n \times t)$ cells in `V`
- time:

Viterbi implementation

Viterbi recursion

$$\alpha(i, j) = \begin{cases} 1 & \text{if } i = 0 \\ \max\limits_{p \in \{1,\ldots,t\}} \alpha(i-1, p)\lambda_j^{(p)}\theta_{x_i}^{(j)} & \text{otherwise} \end{cases}$$

Implementation without recursion:

- for $i = 1, \ldots, n$
  - for $j = 1, \ldots, t$
    - solve $\alpha(i, j)$ and store its value in cell $\mathtt{V}[i, j]$

Complexity

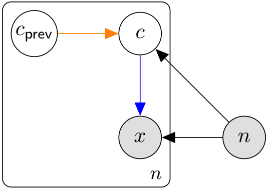
- space: $O(n \times t)$ cells in $\mathtt{V}$
- time: there are $O(n \times t)$ calls to $\alpha(i, j)$
  each requires solving a $\max$ over t pre-computed values
  thus $O(n \times t^2)$

Evaluate our HMM language model

Intrinsically *no need for POS tag sequences*

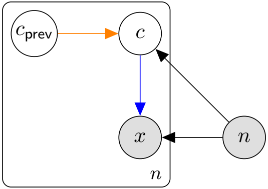
- test set perplexity
- perplexity requires computing $P_{S|n}(x_1^n|n)$ by marginalising over tag sequences
- what's the complexity?

Probability of a sentence



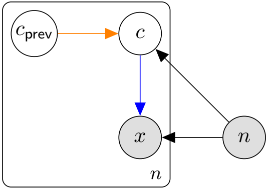
$$P_S(x_1^n) = P_N(n) P_{X_1^n|N}(x_1^n|n)$$

Probability of a sentence



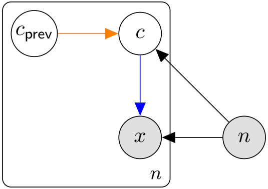
$$
\begin{aligned}
P_S(x_1^n) &= P_N(n) P_{X_1^n|N}(x_1^n|n) \\
&= P_N(n) \sum_{c_1=1}^{t} \cdots \sum_{c_n=1}^{t} P_{X_1^n C_1^n}(x_1^n, c_1^n|n)
\end{aligned}
$$

Probability of a sentence



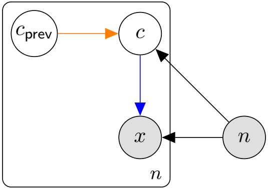
$$
\begin{aligned}
P_S(x_1^n) &= P_N(n) P_{X_1^n|N}(x_1^n|n) \\
&= P_N(n) \sum_{c_1=1}^{t} \cdots \sum_{c_n=1}^{t} P_{X_1^n C_1^n}(x_1^n, c_1^n|n) \\
&= P_N(n) \sum_{c_1=1}^{t} \cdots \sum_{c_n=1}^{t} \prod_{i=1}^{n} P_{XC|C_{\text{prev}}}(x_i, c_i|c_{i-1})
\end{aligned}
$$

Probability of a sentence



$$
\begin{aligned}
P_S(x_1^n) &= P_N(n) P_{X_1^n|N}(x_1^n|n) \\
&= P_N(n) \sum_{c_1=1}^{t} \cdots \sum_{c_n=1}^{t} P_{X_1^n C_1^n}(x_1^n, c_1^n|n) \\
&= P_N(n) \sum_{c_1=1}^{t} \cdots \sum_{c_n=1}^{t} \prod_{i=1}^{n} P_{X C|C_{\text{prev}}}(x_i, c_i|c_{i-1}) \\
&= P_N(n) \sum_{c_1=1}^{t} \cdots \sum_{c_n=1}^{t} \prod_{i=1}^{n} P_{C|C_{\text{prev}}}(c_i|c_{i-1}) P_{X|C}(x_i|c_i)
\end{aligned}
$$

Probability of a sentence



$$P_{S|N}(x_1^n|n) = \alpha_{n+1}(\text{EoS})$$

$$\alpha_{i>1}(c) = P_{X|C}(x_i|c) \sum_{c_{i-1}=1}^{t} \alpha_{i-1}(c_{i-1}) \times P_{C|C_{\text{prev}}}(c|c_{i-1})$$

$$\alpha_1(c) = P_{X|C}(x_1|c) P_{C|C_{\text{prev}}}(c|\text{BoS})$$

where we conveniently pad the sequences with

- $C_0 = \text{BoS}$ and $C_{n+1} = \text{EoS}$
- $X_0 = \text{BoS}$ and $X_{n+1} = \text{EoS}$

For complete derivation see

- pdf or notebook

Identities of summation

Forward algorithm

Forward recursion

$$\alpha(i, j) = \begin{cases} \theta_{x_i}^{(j)} \times \lambda_j^{(0)} & \text{if } i = 1 \\ \theta_{x_i}^{(j)} \times \sum_{p \in \{1, \ldots, t\}} \alpha(i-1, p) \lambda_j^{(p)} & \text{otherwise} \end{cases}$$

Implementation without recursion:

- for $i = 1, \ldots, n$
  - for $j = 1, \ldots, t$
    - solve $\alpha(i, j)$ and store its value in cell $\texttt{M}[i, j]$

Forward algorithm

Forward recursion

$$\alpha(i, j) = \begin{cases} \theta_{x_i}^{(j)} \times \lambda_j^{(0)} & \text{if } i = 1 \\ \theta_{x_i}^{(j)} \times \sum_{p \in \{1,\ldots,t\}} \alpha(i-1, p)\lambda_j^{(p)} & \text{otherwise} \end{cases}$$

Implementation without recursion:

- for $i = 1, \ldots, n$
  - for $j = 1, \ldots, t$
    - solve $\alpha(i, j)$ and store its value in cell $\mathtt{M}[i, j]$

Complexity

- space:

Forward algorithm

Forward recursion

$$\alpha(i, j) = \begin{cases} \theta_{x_i}^{(j)} \times \lambda_j^{(0)} & \text{if } i = 1 \\ \theta_{x_i}^{(j)} \times \sum_{p \in \{1,\ldots,t\}} \alpha(i-1, p)\lambda_j^{(p)} & \text{otherwise} \end{cases}$$

Implementation without recursion:

- for $i = 1, \ldots, n$
  - for $j = 1, \ldots, t$
    - solve $\alpha(i, j)$ and store its value in cell $\mathtt{M}[i, j]$

Complexity

- space: $O(n \times t)$ cells in `M`

Forward algorithm

Forward recursion

$$\alpha(i, j) = \begin{cases} \theta_{x_i}^{(j)} \times \lambda_j^{(0)} & \text{if } i = 1 \\ \theta_{x_i}^{(j)} \times \sum\limits_{p \in \{1,\ldots,t\}} \alpha(i-1, p)\lambda_j^{(p)} & \text{otherwise} \end{cases}$$

Implementation without recursion:

- for $i = 1, \ldots, n$
  - for $j = 1, \ldots, t$
    - solve $\alpha(i, j)$ and store its value in cell $\texttt{M}[i, j]$

Complexity

- space: $O(n \times t)$ cells in `M`
- time:

Forward algorithm

Forward recursion

$$\alpha(i, j) = \begin{cases} \theta_{x_i}^{(j)} \times \lambda_j^{(0)} & \text{if } i = 1 \\ \theta_{x_i}^{(j)} \times \sum_{p \in \{1, \ldots, t\}} \alpha(i-1, p)\lambda_j^{(p)} & \text{otherwise} \end{cases}$$

Implementation without recursion:

- for $i = 1, \ldots, n$
  - for $j = 1, \ldots, t$
    - solve $\alpha(i, j)$ and store its value in cell $\texttt{M}[i, j]$

Complexity

- space: $O(n \times t)$ cells in M
- time: there are $O(n \times t)$ calls to $\alpha(i, j)$
  each requires solving a $\sum$ over t pre-computed values
  thus $O(n \times t^2)$

References I