COMP5046: From Classification to Tagging

Joel Nothman
joel.nothman@sydney.edu.au

School of Information Technologies
University of Sydney

2018-04-17

Part I

Classification Review

Applications of classification

- Wikipedia article entity types
  page $\mapsto \{person, company, location, \ldots\}$
- Thematic classification
  article $\mapsto \{economics, media, health, \ldots\}$
- Spam filtering
  Email $\mapsto \{spam, notspam\}$
- Sentiment detection
  Product review $\mapsto \{positive, neutral, negative\}$
- ...

Prediction with Naïve Bayes

$$\begin{aligned}
\hat{y} &= \operatorname*{argmax}_y p(y|\mathbf{x}) \\
&= \operatorname*{argmax}_y \frac{p(\mathbf{x}, y)}{p(\mathbf{x})} \\
&\propto \operatorname*{argmax}_y p(\mathbf{x}, y) \\
&\propto \operatorname*{argmax}_y p(y) p(\mathbf{x}|y) \\
&\propto \operatorname*{argmax}_y p(y) \prod_i \underbrace{p(x_i|y)}_{\text{independent features}}
\end{aligned}$$

Prediction with Maximum Entropy

$$
\begin{aligned}
\hat{y} &= \operatorname*{argmax}_y p(y|\mathbf{x}) \\
&= \operatorname*{argmax}_y \frac{\exp \mathbf{w} \cdot f(x, y)}{\sum_{y'} \exp \mathbf{w} \cdot f(x, y')} \\
&= \operatorname*{argmax}_y \frac{\exp \sum_i w_i f_i(x, y)}{\sum_{y'} \exp \sum_i w_i f_i(x, y')} \\
&\propto \operatorname*{argmax}_y \exp \sum_i w_i f_i(x, y) \\
&\propto \operatorname*{argmax}_y \sum_i \underbrace{w_i}_{\text{weights}} \underbrace{f_i(x, y)}_{\text{features}}
\end{aligned}
$$

same as NB

Prediction with Perceptron

weights are learnt differently

$$\hat{y} = \operatorname*{argmax}_y \mathbf{w} \cdot f(\mathbf{x}, y)$$

$$= \operatorname*{argmax}_y \sum_i w_i f_i(x, y)$$

no prob output

Part of Speech (POS) Tagging

Mr.	Vinken	is	chairman	of	Elsevier	N.V.	,
NNP	NNP	VBZ	NN	IN	NNP	NNP	,
the	Dutch	publishing	group	.			
DT	NNP	VBG	NN	.			

- 45 POS tags
- 1 million words Penn Treebank WSJ text
- 97% state of the art accuracy

Penn Treebank tagset

Tag	Desciption
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential “there”
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun

Tag	Desciption
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	“to”
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP$	Possessive wh-pronoun
WRB	Wh-adverb

Is increase a NN or a VB?

noun verb

Input:

`equity will` **increase**

Features:

{}

next word end of doc

i	*tag*	*attribute*	*weight*
1	NN	nw *	0.46998
2	NN	w increase	2.97384
3	VB	nw *	-0.08660
4	VB	pw will	1.47305
5	VB	w increase	2.60052

current word

previous word

Is `increase` a NN or a VB?

Input:

`equity will` **increase**

Features:

{pw will}

i	*tag*	*attribute*	*weight*
1	NN	nw *	0.46998
2	NN	w increase	2.97384
3	VB	nw *	-0.08660
4	VB	pw will	1.47305
5	VB	w increase	2.60052

Is `increase` a NN or a VB?

Input:

`equity will` **increase**

Features:

{pw will, w increase}

i	*tag*	*attribute*	*weight*
1	NN	nw *	0.46998
2	NN	w increase	2.97384
3	VB	nw *	-0.08660
4	VB	pw will	1.47305
5	VB	w increase	2.60052

Is `increase` a NN or a VB?

Input:

`equity will` **increase**

Features:

{pw will, w increase, nw *}

i	*tag*	*attribute*	*weight*
1	NN	nw *	0.46998
2	NN	w increase	2.97384
3	VB	nw *	-0.08660
4	VB	pw will	1.47305
5	VB	w increase	2.60052

Is `increase` a NN or a VB?

Input:

`equity will` **increase**

Features:

{pw will, w increase, nw *}

$\underset{y}{\text{argmax}}\, p(y|\mathbf{x})$:

i	*tag*	*attribute*	*weight*
1	NN	nw *	0.46998
2	NN	w increase	2.97384
3	VB	nw *	-0.08660
4	VB	pw will	1.47305
5	VB	w increase	2.60052

Is increase a NN or a VB?

Input:
`equity will` **increase**

Features:
{pw will, w increase, nw *}
previous word next

argmax$_y$ $p(y|\mathbf{x})$:

$p(NN|\mathbf{x}) \propto$

$p(VB|\mathbf{x}) \propto$

i	*tag*	*attribute*	*weight*
1	NN	nw *	0.46998
2	NN	w increase	2.97384
3	VB	nw *	-0.08660
4	VB	pw will	1.47305
5	VB	w increase	2.60052

Is increase a NN or a VB?

Input:

equity will **increase**

Features:

{pw will, w increase, nw *}

i	*tag*	*attribute*	*weight*
1	NN	nw *	0.46998
2	NN	w increase	2.97384
3	VB	nw *	-0.08660
4	VB	pw will	1.47305
5	VB	w increase	2.60052

$\underset{y}{\text{argmax}}\, p(y|\mathbf{x})$:

$p(NN|\mathbf{x}) \propto 0.0$

$w_i = NA \qquad f_i(\textbf{pw will}, NN) = 1$

$p(VB|\mathbf{x}) \propto$

Is `increase` a NN or a VB?

Input:
`equity will` **increase**

Features:
{pw will, w increase, nw *}

i	*tag*	*attribute*	*weight*
1	NN	nw *	0.46998
2	NN	w increase	2.97384
3	VB	nw *	-0.08660
4	VB	pw will	1.47305
5	VB	w increase	2.60052

$\underset{y}{\text{argmax}}\, p(y|\mathbf{x})$:

$p(NN|\mathbf{x}) \propto 0.0 + 2.97384$

$w_5 = 2.97384 \qquad f_5(\textbf{w increase}, NN) = 1$

$p(VB|\mathbf{x}) \propto$

Is increase a NN or a VB?

Input:

`equity will` **increase**

Features:

{pw will, w increase, nw *}

i	*tag*	*attribute*	*weight*
1	NN	nw *	0.46998
2	NN	w increase	2.97384
3	VB	nw *	-0.08660
4	VB	pw will	1.47305
5	VB	w increase	2.60052

$\operatorname{argmax}_y p(y|\mathbf{x})$:

$p(NN|\mathbf{x}) \propto 0.0 + 2.97384 + 0.46998$

$w_1 = 0.46998 \qquad f_1(\textbf{nw *}, NN) = 1$

$p(VB|\mathbf{x}) \propto$

Is `increase` a NN or a VB?

Input:

`equity will` **increase**

Features:

{pw will, w increase, nw *}

i	*tag*	*attribute*	*weight*
1	NN	nw *	0.46998
2	NN	w increase	2.97384
3	VB	nw *	-0.08660
4	VB	pw will	1.47305
5	VB	w increase	2.60052

$\underset{y}{\operatorname{argmax}}\, p(y|\mathbf{x})$:

$p(NN|\mathbf{x}) \propto 0.0 + 2.97384 + 0.46998 = 3.44382$

$p(VB|\mathbf{x}) \propto$

Is increase a NN or a VB?

Input:
equity will **increase**

Features:
{pw will, w increase, nw *}

i	*tag*	*attribute*	*weight*
1	NN	nw *	0.46998
2	NN	w increase	2.97384
3	VB	nw *	-0.08660
4	VB	pw will	1.47305
5	VB	w increase	2.60052

$\operatorname{argmax}_y p(y|\mathbf{x})$:

$$p(NN|\mathbf{x}) \propto 0.0 + 2.97384 + 0.46998 = 3.44382$$

$$p(VB|\mathbf{x}) \propto 1.47305$$

$$w_4 = 1.47305 \qquad f_4(\textbf{pw will}, VB) = 1$$

Is increase a NN or a VB?

Input:

`equity will` **increase**

Features:

{pw will, w increase, nw *}

i	*tag*	*attribute*	*weight*
1	NN	nw *	0.46998
2	NN	w increase	2.97384
3	VB	nw *	-0.08660
4	VB	pw will	1.47305
5	VB	w increase	2.60052

$\operatorname{argmax}_y p(y|\mathbf{x})$:

$$p(NN|\mathbf{x}) \propto 0.0 + 2.97384 + 0.46998 = 3.44382$$

$$p(VB|\mathbf{x}) \propto 1.47305 + 2.60052$$

$$w_5 = 2.60052 \qquad f_5(\textbf{w increase}, VB) = 1$$

Is increase a NN or a VB?

Input:
`equity will` **increase**

Features:
{pw will, w increase, nw *}

i	*tag*	*attribute*	*weight*
1	NN	nw *	0.46998
2	NN	w increase	2.97384
3	VB	nw *	-0.08660
4	VB	pw will	1.47305
5	VB	w increase	2.60052

$\operatorname*{argmax}_y p(y|\mathbf{x})$:

$$p(NN|\mathbf{x}) \propto 0.0 + 2.97384 + 0.46998 = 3.44382$$

$$p(VB|\mathbf{x}) \propto 1.47305 + 2.60052 - 0.08660$$

$$w_3 = -0.08660 \qquad f_3(\mathbf{nw\ *}, VB) = 1$$

Is increase a NN or a VB?

Input:

`equity will` **increase**

Features:

{pw will, w increase, nw *}

i	*tag*	*attribute*	*weight*
1	NN	nw *	0.46998
2	NN	w increase	2.97384
3	VB	nw *	-0.08660
4	VB	pw will	1.47305
5	VB	w increase	2.60052

$\operatorname{argmax}_y p(y|\mathbf{x})$:

$$p(NN|\mathbf{x}) \propto 0.0 + 2.97384 + 0.46998 = 3.44382$$

$$p(VB|\mathbf{x}) \propto 1.47305 + 2.60052 - 0.08660 = 3.98697$$

Problems with classification

- Classification ignores structure
  $\Longrightarrow$ **No model of dependence between outputs**
- POS: VB (e.g., `increase`) more likely after MD (e.g., `will`)
- NER: I-PER (e.g., `Gillard`) more likely after B-PER (e.g., `Julia`)

Tagging

shallow syntactic tagging

Mr.	Vinken	is	chairman	of	Elsevier	N.V.	,
NNP	NNP	VBZ	NN	IN	NNP	NNP	,
B-NP	I-NP	B-VP	B-NP	B-PP	B-NP	I-NP	O
B-PER	I-PER	O	O	O	B-ORG	I-ORG	O

phrase tagging

named entity tagging

the	Dutch	publishing	group	.
DT	NNP	VBG	NN	.
B-NP	I-NP	I-NP	I-NP	O
O	O	O	O	O

not interesting entity features

Tagging

- Find the *best sequence*:
  - words
  - tags
  - base pairs
  - ...
- **Which sequence** is the best sequence?
  $\Longrightarrow$ the most probable sequence

$$\underset{y_1 \ldots y_n}{\operatorname{argmax}} \, p(y_1 \ldots y_n)$$

- we need a **probability model of language**

Language modelling vs. tagging

- LMs used to measure likelihood of a given sentence $p(W)$
  $\Longrightarrow$ output is a probability
- Taggers used to predict best tag sequence for sentence $p(T|W)$
  $\Longrightarrow$ output is a distribution over possible sequences

Part II

Tagging

Outline

- sequence tagging
  - Hidden Markov Models
  - Maximum Entropy Markov Models
- finding the optimal sequence
  - using a single history
  - Viterbi
  - Beam search
- features used for:
  - POS tagging (Ratnaparkhi, C&C, Toutanova et al)
  - Named Entity Recognition (Borthwick, C&C, Klein et al)
- Conditional Random Fields

Language Modelling

- Find the best sequence (words, tags, base pairs, ...)
  $\Longrightarrow$ **the most probable sequence**

$$\underset{y_1 \dots y_n}{\operatorname{argmax}}\, p(y_1 \dots y_n)$$

tags | words

- Chain rule expansion:

$$p(y_1 \dots y_n) = p(y_1)p(y_2|y_1)p(y_3|y_1, y_2) \cdots p(y_n|y_1, \dots, y_{n-1})$$

predict y_1
predict y_2 given y_1
predict y_3 given y_1 and y_2
...

Markov Assumption

- **Each prediction cannot depend on entire history!**
- Markov model approximation:

$$\begin{aligned} p(y_1 \ldots y_n) &= p(y_1)p(y_2|y_1)p(y_3|y_1, y_2)\cdots p(y_n|y_1, \ldots, y_{n-1}) \\ &\approx p(y_1)p(y_2|y_1)p(y_3|y_2)\cdots p(y_n|y_{n-1}) \end{aligned}$$

- Current prediction only based on previous prediction
- In theory can use any fixed length history
- In practice a history of 2 is typically used (for English)

Andrei Markov (1856–1922)

http://en.wikipedia.org/wiki/Andrey_Markov

An example of statistical investigation in the text of 'Eugene Onyegin' illustrating coupling of 'tests' in chains. In *Proceedings of the Academy of Sciences*, St. Petersburg, 7:153–162, 1913.

Tagging with Probabilities

- **Find the best tag sequence** ***given the words*** (cond. probability):

$$\operatorname*{argmax}_{t_1 \ldots t_n} p(t_1 \ldots t_n | w_1 \ldots w_n)$$

prob of tags and words

- Alternatively maximise $p(t_1 \ldots t_n, w_1 \ldots w_n)$ (joint probability):

$$\operatorname*{argmax}_{t_1 \ldots t_n} p(t_1 \ldots t_n | w_1 \ldots w_n) = \operatorname*{argmax}_{t_1 \ldots t_n} \frac{p(t_1 \ldots t_n, w_1 \ldots w_n)}{p(w_1 \ldots w_n)}$$
$$= \operatorname*{argmax}_{t_1 \ldots t_n} p(t_1 \ldots t_n, w_1 \ldots w_n)$$

discriminative → does not ac for prob of language ↳ more flexible to include other features

of tag seq given word seq

- **MaxEnt taggers directly maximise conditional probability**
- **Hidden Markov Model taggers maximise joint probability** (easier to estimate)

generative ↳ needs to account for prob of language

Hidden Markov Model Tagging

- Maximise the joint probability:

$$p(t_1 \ldots t_n, w_1 \ldots w_n) = p(t_1 \ldots t_n)p(w_1 \ldots w_n | t_1 \ldots t_n)$$

- Tag sequence probability (first order Markov Model):

$$p(t_1 \ldots t_n) \approx p(t_1)p(t_2|t_1)p(t_3|t_2) \cdots p(t_n|t_{n-1})$$

markov approx

- Word sequence probability (given the tags):

$$p(w_1 \ldots w_n | t_1 \ldots t_n) \approx p(w_1|t_1)p(w_2|t_2) \cdots p(w_n|t_n)$$

↳ each w is only dependent on tags

- **Using $p(w_1 \ldots w_n | t_1 \ldots t_n)$ is counter-intuitive but correct**
  since we're maximising the joint probability

Three questions for HMMs

- *language modelling*: how compute likelihood of a sentence $p(W)$?

  (Manning and Schütze, Section 9.3)

- *training*: **how find model that best explains the data?**
- *tagging*: how choose a tag sequence for a given sentence $p(T|W)$?

Maximum Likelihood Estimation for Markov Models

train

- Probabilities are estimated from annotated data
- Estimates are simple relative frequencies (MLE):

$$p^*(t_i|t_{i-1}) = \frac{\text{count}(t_{i-1}, t_i)}{\text{count}(t_{i-1})}$$

$$p^*(w_i|t_i) = \frac{\text{count}(w_i, t_i)}{\text{count}(t_i)}$$

CPT excerpt for tags (transition probabilities)

http://www.clips.ua.ac.be/conll2000/chunking/

$\text{count}(T_{i-1}, T_i) =$

t_{i-1} \ t_i	*	MD	NN	VB	Total
*	0	5	404	43	8937
MD	0	0	0	1706	2167
NN	14	548	3546	43	30147
VB	0	5	394	41	6017

CPT excerpt for tags (transition probabilities)

http://www.clips.ua.ac.be/conll2000/chunking/

$\text{count}(T_{i-1}, T_i) =$

t_{i-1} \ t_i	*	MD	NN	VB	Total
*	0	5	404	43	8937
MD	0	0	0	1706	2167
NN	14	548	3546	43	30147
VB	0	5	394	41	6017

$p^*(T_i | T_{i-1}) =$

t_{i-1} \ t_i	*	MD	NN	VB
*	0.00000	0.00056	0.04521	0.00481
MD				
NN				
VB				

CPT excerpt for tags (transition probabilities)

http://www.clips.ua.ac.be/conll2000/chunking/

$\text{count}(T_{i-1}, T_i) =$

t_{i-1} \ t_i	*	MD	NN	VB	Total
*	0	5	404	43	8937
MD	0	0	0	1706	2167
NN	14	548	3546	43	30147
VB	0	5	394	41	6017

$p^*(T_i | T_{i-1}) =$

t_{i-1} \ t_i	*	MD	NN	VB
*	0.00000	0.00056	0.04521	0.00481
MD	0.00000	0.00000	0.00000	0.78726
NN				
VB				

CPT excerpt for tags (transition probabilities)

http://www.clips.ua.ac.be/conll2000/chunking/

$\text{count}(T_{i-1}, T_i) =$

t_{i-1} \ t_i	*	MD	NN	VB	Total
*	0	5	404	43	8937
MD	0	0	0	1706	2167
NN	14	548	3546	43	30147
VB	0	5	394	41	6017

5 / 8937

$p^*(T_i | T_{i-1}) =$

t_{i-1} \ t_i	*	MD	NN	VB
*	0.00000	0.00056	0.04521	0.00481
MD	0.00000	0.00000	0.00000	0.78726
NN	0.00046	0.01818	0.11762	0.00143
VB	0.00000	0.00083	0.06548	0.00681

CPT excerpt for words (emission probabilities)

http://www.clips.ua.ac.be/conll2000/chunking/

all words

$\text{count}(W_i, T_i) =$

t_i \ w_i	equity	increase	will	Total
MD	0	0	658	2167
NN	33	78	1	30147
VB	0	28	0	6017

all tags

CPT excerpt for words (emission probabilities)

http://www.clips.ua.ac.be/conll2000/chunking/

$\text{count}(W_i, T_i) =$

t_i \ w_i	equity	increase	will	Total
MD	0	0	658	2167
NN	33	78	1	30147
VB	0	28	0	6017

$p^*(W_i | T_i) =$

t_i \ w_i	equity	increase	will
MD	0.00000	0.00000	0.30365
NN			
VB			

CPT excerpt for words (emission probabilities)

http://www.clips.ua.ac.be/conll2000/chunking/

$\text{count}(W_i, T_i) =$

t_i \ w_i	equity	increase	will	Total
MD	0	0	658	2167
NN	33	78	1	30147
VB	0	28	0	6017

$p^*(W_i | T_i) =$

t_i \ w_i	equity	increase	will
MD	0.00000	0.00000	0.30365
NN	0.00109	0.00259	0.00003
VB			

CPT excerpt for words (emission probabilities)

http://www.clips.ua.ac.be/conll2000/chunking/

$\text{count}(W_i, T_i) =$

t_i \ w_i	equity	increase	will	Total
MD	0	0	658	2167
NN	33	78	1	30147
VB	0	28	0	6017

$p^*(W_i | T_i) =$

t_i \ w_i	equity	increase	will
MD	0.00000	0.00000	0.30365
NN	0.00109	0.00259	0.00003
VB	0.00000	0.00465	0.00000

Generative process for a state-emission HMM

finds the most probable seq of tags

Given a HMM instance, we can generate sentences:

1. $i := 1$
2. $t_i :=$ sample from $p^*(T|*)$
3. **do**
4. $\quad w_i :=$ sample from $p^*(W|t_i)$ emission prob matrix
5. $\quad i := i + 1$
6. $\quad t_i :=$ sample from $p^*(T|t_{i-1})$
7. **until** $t_i \equiv *$

random

transition prob matrix

but this isn't tagging. . .

how choose a tag sequence for a given sentence $p(T|W)$?

Finding the most probable sequence

- Current decision depends on previous decision(s)
- Cannot simply take the most probable tag for each word
- Brute force search would take $\mathbf{O}(n^{\#\text{toks}})$

- **Viterbi algorithm finds the shortest path through the tag lattice**
  - $\mathbf{O}(n^2)$ in the number of tags (e.g. POS tags 45^2)

- An instance of dynamic programming
  - found throughout NLP for estimation and decoding

CPT excerpts from CoNLL 2000 data

http://www.clips.ua.ac.be/conll2000/chunking/

$p^*(T_i|T_{i-1}) =$

t_{i-1} \ t_i	*	MD	NN	VB
*	0.00000	0.00056	0.04521	0.00481
MD	0.00000	0.00000	0.00000	0.78726
NN	0.00046	0.01818	0.11762	0.00143
VB	0.00000	0.00083	0.06548	0.00681

$p^*(W_i|T_i) =$

t_i \ w_i	equity	increase	will
MD	0.00000	0.00000	0.30365
NN	0.00109	0.00259	0.00003
VB	0.00000	0.00465	0.00000

CPT excerpts from CoNLL 2000 data: log-transformed

http://www.clips.ua.ac.be/conll2000/chunking/

$\log_{10} p^*(T_i|T_{i-1}) =$

t_{i-1} \ t_i	*	MD	NN	VB
*	-∞	-3.25	-1.34	-2.32
MD	-∞	-∞	-∞	-0.10
NN	-3.34	-1.74	-0.93	-2.84
VB	-∞	-3.08	-1.18	-2.17

should be smoothed

$\log_{10} p^*(W_i|T_i) =$

t_i \ w_i	equity	increase	will
MD	-∞	-∞	-0.52
NN	-2.96	-2.59	-4.52
VB	-∞	-2.33	-∞

Viterbi trace: $\max p(t_0, t_1, \mathbf{w})$

↳ upto 1st tag

equity will increase

t_0	t_1	Transition		Emission
*	MD	$p(t_1 = \mathrm{MD} \mid t_0 = *)$	$\times$	$p(w_1 = \text{equity} \mid t_1 = \mathrm{MD})$
*	NN	$p(t_1 = \mathrm{NN} \mid t_0 = *)$	$\times$	$p(w_1 = \text{equity} \mid t_1 = \mathrm{NN})$
*	VB	$p(t_1 = \mathrm{VB} \mid t_0 = *)$	$\times$	$p(w_1 = \text{equity} \mid t_1 = \mathrm{VB})$

t_0	t_1	$\log p$
*	MD	$-3.25 + -\infty$
*	NN	$-1.34 + -2.96 = -4.30$
*	VB	$-2.32 + -\infty$

Viterbi trace: $\max p(t_0, t_1, t_2, \mathbf{w})$

↳ upto 2nd tag

t_1	t_2	Transition		Emission		Best history
MD	MD	$p(t_2 = \mathrm{MD} \mid t_1 = \mathrm{MD})$	$\times$	$p(w_2 = \text{will} \mid t_2 = \mathrm{MD})$	$\times$	$\max_{t_0} p(t_1 = \mathrm{MD}, t_0)$
NN	MD	$p(t_2 = \mathrm{MD} \mid t_1 = \mathrm{NN})$	$\times$	$p(w_2 = \text{will} \mid t_2 = \mathrm{MD})$	$\times$	$\max_{t_0} p(t_1 = \mathrm{NN}, t_0)$
VB	MD	$p(t_2 = \mathrm{MD} \mid t_1 = \mathrm{VB})$	$\times$	$p(w_2 = \text{will} \mid t_2 = \mathrm{MD})$	$\times$	$\max_{t_0} p(t_1 = \mathrm{VB}, t_0)$
MD	NN	$p(t_2 = \mathrm{NN} \mid t_1 = \mathrm{MD})$	$\times$	$p(w_2 = \text{will} \mid t_2 = \mathrm{NN})$	$\times$	$\max_{t_0} p(t_1 = \mathrm{MD}, t_0)$
NN	NN	$p(t_2 = \mathrm{NN} \mid t_1 = \mathrm{NN})$	$\times$	$p(w_2 = \text{will} \mid t_2 = \mathrm{NN})$	$\times$	$\max_{t_0} p(t_1 = \mathrm{NN}, t_0)$
VB	NN	$p(t_2 = \mathrm{NN} \mid t_1 = \mathrm{VB})$	$\times$	$p(w_2 = \text{will} \mid t_2 = \mathrm{NN})$	$\times$	$\max_{t_0} p(t_1 = \mathrm{VB}, t_0)$
MD	VB	$p(t_2 = \mathrm{VB} \mid t_1 = \mathrm{MD})$	$\times$	$p(w_2 = \text{will} \mid t_2 = \mathrm{VB})$	$\times$	$\max_{t_0} p(t_1 = \mathrm{MD}, t_0)$
NN	VB	$p(t_2 = \mathrm{VB} \mid t_1 = \mathrm{NN})$	$\times$	$p(w_2 = \text{will} \mid t_2 = \mathrm{VB})$	$\times$	$\max_{t_0} p(t_1 = \mathrm{NN}, t_0)$
VB	VB	$p(t_2 = \mathrm{VB} \mid t_1 = \mathrm{VB})$	$\times$	$p(w_2 = \text{will} \mid t_2 = \mathrm{VB})$	$\times$	$\max_{t_0} p(t_1 = \mathrm{VB}, t_0)$

t_1	t_2	$\log p$	$\operatorname{argmax}_{t_0}$	max?
MD	MD	$-\infty - 0.52 - \infty = -\infty$	–	
NN	MD	$-1.74 - 0.52 - 4.30 = -6.56$	*	yes
VB	MD	$-3.08 - 0.52 - \infty = -\infty$	–	
MD	NN	$-\infty - 4.52 - \infty = -\infty$	–	
NN	NN	$-0.93 - 4.52 - 4.30 = -9.75$	*	yes
VB	NN	$-1.18 - 4.52 - \infty = -\infty$	–	
MD	VB	$-0.10 - \infty - \infty = -\infty$	–	
NN	VB	$-2.84 - \infty - 4.30 = -\infty$	*	
VB	VB	$-2.17 - \infty - \infty = -\infty$	–	

Viterbi trace: $\max p(t_0, t_1, t_2, t_3, \mathbf{w})$

t_2	t_3	Transition		Emission		Best history
MD	MD	$p(t_3 = \text{MD} \mid t_2 = *)$	$\times$	$p(w_3 = \text{increase} \mid t_3 = \text{MD})$	$\times$	$\max_{t_0,t_1} p(t_2 = \text{MD}, t_0, t_1)$
NN	MD	$p(t_3 = \text{MD} \mid t_2 = *)$	$\times$	$p(w_3 = \text{increase} \mid t_3 = \text{MD})$	$\times$	$\max_{t_0,t_1} p(t_2 = \text{NN}, t_0, t_1)$
VB	MD	$p(t_3 = \text{MD} \mid t_2 = *)$	$\times$	$p(w_3 = \text{increase} \mid t_3 = \text{MD})$	$\times$	$\max_{t_0,t_1} p(t_2 = \text{VB}, t_0, t_1)$
MD	NN	$p(t_3 = \text{NN} \mid t_2 = \text{MD})$	$\times$	$p(w_3 = \text{increase} \mid t_3 = \text{NN})$	$\times$	$\max_{t_0,t_1} p(t_2 = \text{MD}, t_0, t_1)$
NN	NN	$p(t_3 = \text{NN} \mid t_2 = \text{NN})$	$\times$	$p(w_3 = \text{increase} \mid t_3 = \text{NN})$	$\times$	$\max_{t_0,t_1} p(t_2 = \text{NN}, t_0, t_1)$
VB	NN	$p(t_3 = \text{NN} \mid t_2 = \text{VB})$	$\times$	$p(w_3 = \text{increase} \mid t_3 = \text{NN})$	$\times$	$\max_{t_0,t_1} p(t_2 = \text{VB}, t_0, t_1)$
MD	VB	$p(t_3 = \text{VB} \mid t_2 = \text{MD})$	$\times$	$p(w_3 = \text{increase} \mid t_3 = \text{VB})$	$\times$	$\max_{t_0,t_1} p(t_2 = \text{MD}, t_0, t_1)$
NN	VB	$p(t_3 = \text{VB} \mid t_2 = \text{NN})$	$\times$	$p(w_3 = \text{increase} \mid t_3 = \text{VB})$	$\times$	$\max_{t_0,t_1} p(t_2 = \text{NN}, t_0, t_1)$
VB	VB	$p(t_3 = \text{VB} \mid t_2 = \text{VB})$	$\times$	$p(w_3 = \text{increase} \mid t_3 = \text{VB})$	$\times$	$\max_{t_0,t_1} p(t_2 = \text{VB}, t_0, t_1)$

t_2	t_3	$\log p$	$\operatorname{argmax}_{t_0,t_1}$	max?
MD	MD	$-\infty - \infty - 6.56 = -\infty$	*, NN	
NN	MD	$-1.74 - \infty - 9.75 = -\infty$	*, NN	
VB	MD	$-3.08 - \infty - \infty = -\infty$	–	
MD	NN	$-\infty - 2.59 - 6.56 = -\infty$	*, NN	
NN	NN	$-0.93 - 2.59 - 9.75 = -13.27$	*, NN	yes
VB	NN	$-1.18 - 2.59 - \infty = -15.17$	–	
MD	VB	$-0.10 - 2.33 - 6.56 = -8.99$	*, NN	yes
NN	VB	$-2.84 - 2.33 - 9.75 = -14.92$	*, NN	
VB	VB	$-2.17 - 2.33 - \infty = -\infty$	–	

Viterbi trace: $\max p(t_0, t_1, t_2, t_3, \mathbf{w})$

t_2	t_3	Transition		Emission		Best history
MD	MD	$p(t_3 = \mathrm{MD} \mid t_2 = *)$	×	$p(w_3 = \text{increase} \mid t_3 = \mathrm{MD})$	×	$\max_{t_0,t_1} p(t_2 = \mathrm{MD}, t_0, t_1)$
NN	MD	$p(t_3 = \mathrm{MD} \mid t_2 = *)$	×	$p(w_3 = \text{increase} \mid t_3 = \mathrm{MD})$	×	$\max_{t_0,t_1} p(t_2 = \mathrm{NN}, t_0, t_1)$
VB	MD	$p(t_3 = \mathrm{MD} \mid t_2 = *)$	×	$p(w_3 = \text{increase} \mid t_3 = \mathrm{MD})$	×	$\max_{t_0,t_1} p(t_2 = \mathrm{VB}, t_0, t_1)$
MD	NN	$p(t_3 = \mathrm{NN} \mid t_2 = \mathrm{MD})$	×	$p(w_3 = \text{increase} \mid t_3 = \mathrm{NN})$	×	$\max_{t_0,t_1} p(t_2 = \mathrm{MD}, t_0, t_1)$
NN	NN	$p(t_3 = \mathrm{NN} \mid t_2 = \mathrm{NN})$	×	$p(w_3 = \text{increase} \mid t_3 = \mathrm{NN})$	×	$\max_{t_0,t_1} p(t_2 = \mathrm{NN}, t_0, t_1)$
VB	NN	$p(t_3 = \mathrm{NN} \mid t_2 = \mathrm{VB})$	×	$p(w_3 = \text{increase} \mid t_3 = \mathrm{NN})$	×	$\max_{t_0,t_1} p(t_2 = \mathrm{VB}, t_0, t_1)$
MD	VB	$p(t_3 = \mathrm{VB} \mid t_2 = \mathrm{MD})$	×	$p(w_3 = \text{increase} \mid t_3 = \mathrm{VB})$	×	$\max_{t_0,t_1} p(t_2 = \mathrm{MD}, t_0, t_1)$
NN	VB	$p(t_3 = \mathrm{VB} \mid t_2 = \mathrm{NN})$	×	$p(w_3 = \text{increase} \mid t_3 = \mathrm{VB})$	×	$\max_{t_0,t_1} p(t_2 = \mathrm{NN}, t_0, t_1)$
VB	VB	$p(t_3 = \mathrm{VB} \mid t_2 = \mathrm{VB})$	×	$p(w_3 = \text{increase} \mid t_3 = \mathrm{VB})$	×	$\max_{t_0,t_1} p(t_2 = \mathrm{VB}, t_0, t_1)$

t_2	t_3	$\log p$	$\operatorname{argmax}_{t_0,t_1}$	max?
MD	MD	$-\infty - \infty - 6.56 = -\infty$	*, NN	
NN	MD	$-1.74 - \infty - 9.75 = -\infty$	*, NN	
VB	MD	$-3.08 - \infty - \infty = -\infty$	–	
MD	NN	$-\infty - 2.59 - 6.56 = -\infty$	*, NN	
NN	NN	$-0.93 - 2.59 - 9.75 = -13.27$	*, NN	yes
VB	NN	$-1.18 - 2.59 - \infty = -15.17$	–	
MD	VB	$-0.10 - 2.33 - 6.56 = -8.99$	*, NN	yes
NN	VB	$-2.84 - 2.33 - 9.75 = -14.92$	*, NN	
VB	VB	$-2.17 - 2.33 - \infty = -\infty$	–	

The essence of the Viterbi algorithm

for word i
for each tag t
we keep track of the best score so far
that labels i with that tag,
and the previous tag that led to it

sequence of smaller problems

Finds the most probable tag sequence
under a Markov assumption for tag bigrams

$$p(t_1, t_2, \ldots, t_n) = p(t_1)p(t_2|t1) \cdots p(t_n|t_{n-1})$$

by solving the problem for t_1, then t_1, t_2, then t_1, t_2, t_3, ...

Notes on Viterbi

- $\mathbf{O}(n^2)$ in the number of tags (e.g. POS tags 45^2)
- finds max and argmax of $\text{score}(t_1, \ldots, t_n, w_1, \ldots w_n)$
  - HMM is generative and probabilistic: score $= p$ is factored into transition and emission
  - can use Viterbi where score is derived from features discriminatively
  - some features would encode previous tag
  - technically can condition on previous k tags for some fixed k

- Beam search works well in practice: approximate search
- $\mathbf{O}(n^2)$ in the beam width (typically 5^2) only use best 5 tags

Part of Speech (POS) Tagging

Mr.	Vinken	is	chairman	of	Elsevier	N.V.	,
NNP	**NNP**	**VBZ**	**NN**	**IN**	**NNP**	**NNP**	**,**

the	Dutch	publishing	group	.
DT	**NNP**	**VBG**	**NN**	**.**

- 45 POS tags
- 1 million words Penn Treebank WSJ text
- 97% state of the art accuracy

Chunk Tagging

Mr.	Vinken	is	chairman	of	Elsevier	N.V.	,
B-NP	**I-NP**	**B-VP**	**B-NP**	**B-PP**	**B-NP**	**I-NP**	**O**
the	Dutch	publishing	group	.			
B-NP	**I-NP**	**I-NP**	**I-NP**	**O**			

- 18 phrase tags
- 1 million words Penn Treebank WSJ text
- 94% state of the art accuracy
- Alternative: B-XX only used to separate adjacent phrases of same type

Named Entity Tagging

Mr.	Vinken	is	chairman	of	Elsevier	N.V.	,
B-PER	**I-PER**	**O**	**O**	**O**	**B-ORG**	**I-ORG**	**O**

the	Dutch	publishing	group	.
O	**B-MISC**	**O**	**O**	**O**

- 4 named entity tags
- 400,000 words CoNLL 2003 shared task data
- Over Reuters newswire text
- 90% state of the art accuracy

Problems with Markov Model Taggers

- unreliable zero or very low counts
  - does a zero count indicate an impossible event?

$\Longrightarrow$ *smoothing* the counts solves this problem

- Words not seen in the data are especially problematic

$\Longrightarrow$ would like to include word internal information
e.g. capitalisation or suffix information

- Cannot incorporate diverse pieces of evidence for predicting tags
  e.g. global document information

Tagging with Maximum Entropy Markov Models

- The conditional probability of a tag sequence $t_1 \dots t_n$ is

$$p(t_1 \dots t_n | w_1 \dots w_n) \approx \prod_{i=1}^{n} p(t_i | C_i)$$

given a sentence $w_1 \dots w_n$ and contexts $C_1 \dots C_n$

- The context includes previously assigned tags (for a fixed history)
- Beam search is used to find the most probable sequence in practice

Ratnaparkhi POS-tagging Contextual Predicates

Condition	Contextual predicate
$\text{freq}(w_i) < 5$	X is prefix/suffix of w_i, $\lvert X \rvert \leq 4$
	w_i contains a digit
	w_i contains uppercase character
	w_i contains a hyphen
$\forall w_i$	$w_i = X$
	$w_{i-1} = X$, $w_{i-2} = X$
	$w_{i+1} = X$, $w_{i+2} = X$
$\forall w_i$	$\text{KLASS}_{i-1} = X$
	$\text{KLASS}_{i-2}\text{KLASS}_{i-1} = XY$

C&C NER Contextual Predicates

Condition	Contextual predicate
$\mathrm{freq}(w_i) < 5$	X is prefix/suffix of w_i, $\|X\| \leq 4$
	w_i contains a digit
	w_i contains uppercase character
	w_i contains a hyphen
$\forall w_i$	$w_i = X$
	$w_{i-1} = X$, $w_{i-2} = X$
	$w_{i+1} = X$, $w_{i+2} = X$
$\forall w_i$	$\mathrm{POS}_i = X$
	$\mathrm{POS}_{i-1} = X$, $\mathrm{POS}_{i-2} = X$
	$\mathrm{POS}_{i+1} = X$, $\mathrm{POS}_{i+2} = X$
$\forall w_i$	$\mathrm{KLASS}_{i-1} = X$
	$\mathrm{KLASS}_{i-2}\mathrm{KLASS}_{i-1} = XY$

C&C NER Additional Contextual Predicates

Condition	Contextual predicate
$\text{freq}(w_i) < 5$	w_i contains period
	w_i contains punctuation
	w_i is only digits
	w_i is a number
	w_i is $\{$upper,lower,title,mixed$\}$ case
	w_i is alphanumeric
	length of w_i
	w_i has only Roman numerals
	w_i is an initial (X.)
	w_i is an acronym (ABC, A.B.C.)

C&C NER Additional Contextual Predicates

Condition	Contextual predicate
$\forall w_i$	memory NE tag for w_i
	unigram tag of w_{i+1}
	unigram tag of w_{i+2}
$\forall w_i$	w_i in a gazetteer
	w_{i-1} in a gazetteer
	w_{i+1} in a gazetteer
$\forall w_i$	w_i not lowercase and $f_{\text{lc}} > f_{\text{uc}}$
$\forall w_i$	unigrams of word type
	bigrams of word types
	trigrams of word types

Example Word Types (Collins, 2002)

- `Moody` $\Longrightarrow$ `Aa`
- `A.B.C.` $\Longrightarrow$ `A.A.A.`
- `1,345.00` $\Longrightarrow$ `0,0.0`

- `Mr. Smith` $\Longrightarrow$ `Aa. Aa`

Fancier NER predicates (Kazama and Torisawa, 2008)

contextual predicate for tagging w_i:
X is a Wikipedia category for A
$\forall j \leq i \leq k$
such that the phrase $\{w_j, \cdots w_k\}$
is the title of Wikipedia article A

Thus our discriminative learner can learn an association between Wikipedia categories and named entity types

Linear Chain Conditional Random Fields

- assign probability to entire sequence as a single classification
- use probabilities of tag-bigrams
- overcomes the *label bias* problem
    - bias towards tags with few possible successors in HMM/MEMM
    - but in practice this doesn't seem to be the major difficulty
- there are many tasks where CRF is now state-of-the-art
- recently combined with learnt word sequence representations (with BiLSTMs)

Take away

- Sequence tagging classifies each token
- with dependencies between tags
- Phrase labelling through sequence tagging
- Applications to identify syntax and reference
- Common features for sequence labelling in English
- Viterbi algorithm: why and how