Foundations of Natural Language Processing
Lecture 13
Part-of-speech Tagging and HMMs

Ivan Titov

School of informatics

Sequence labeling problems

- Definition:
  - Input: sequences of variable length $\mathbf{x} = (x_1, x_2, \ldots, x_{|x|}), x_i \in \mathcal{X}$
  - Output: every position is associated with a label $\mathbf{y} = (y_1, y_2, \ldots, y_{|x|}), y_i \in \{1, \ldots, N\}$
- An example:
  - Part-of-speech tagging

$\mathbf{x}$ =	**John**	**carried**	**a**	**tin**	**can**	**.**
$\mathbf{y}$ =	NNP	VBD	DT	NN	NN	.

  - Named-entity recognition, shallow parsing ("chunking"), gesture recognition from video-streams, …

Note the change in notation from previous time

Part-of-speech tagging: ambiguity

$\mathbf{x} =$	**John**	**carried**	**a**	**tin**	**can**	**.**
$\mathbf{y} =$	NNP	VBD	DT	NN	NN or MD?	.

In fact, even knowing that the previous word is a noun is not enough

If you just predict the most frequent tag for each word you will make a mistake here

- Labels:
  - NNP – proper singular noun;
  - VBD – verb, past tense
  - DT – determiner
  - NN – singular noun
  - MD - modal
  - . - final punctuation

- Consider

Tin	**can**	**cause**	**poisoning**	**…**
NN	MD	VB	NN	…

One need to model interaction between labels to successfully resolve ambiguities, hence called *structured prediction* problem

Named Entity Recognition

[ORG Chelsea], despite their name, are not based in [LOC Chelsea], but in neighbouring [LOC Fulham] .

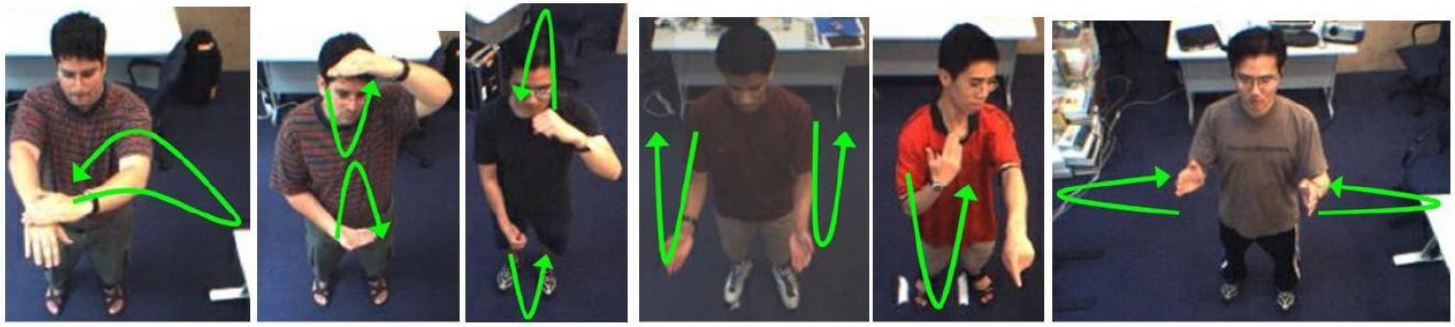
[PER Bill Clinton] embarrassed [PER Chelsea] at her wedding at [LOC Astor Courts]

x =	**Bill**	**Clinton**	**embarrassed**	**Chelsea**	**at**	**her**	**wedding**	**at**	**Astor**	**Courts**
y =	B-PERS	I-PERS	O	B-PERS	O	O	O	O	B-LOC	I-LOC

Vision: Gesture Recognition

- Given a sequence of frames in a video annotate each frame with a gesture type:

- Types of gestures:



Flip back | Shrink vertically | Expand vertically | Double back | Point and back | Expand horizontally

It is hard to predict gestures for each frame in isolation, again need to exploit interaction between gestures in different frames

Hidden Markov Models

- We will consider the part-of-speech (POS) tagging example

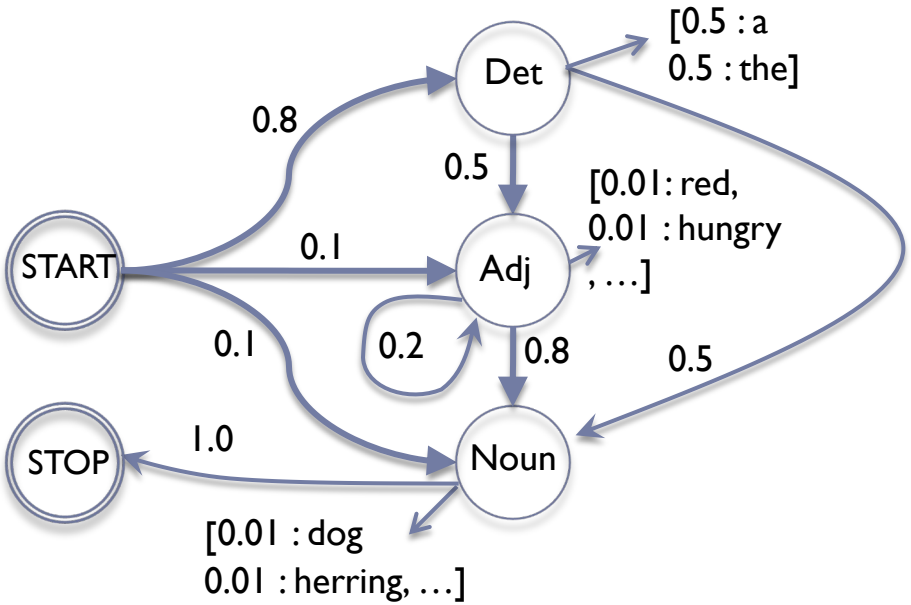
John	**carried**	**a**	**tin**	**can**	**.**
NNP	VBD	DT	NN	NN	.

- A "generative" model, i.e.:
  - Model: Introduce a parameterized model of how both words and tags are generated $P(\mathbf{x}, \mathbf{y}|\theta)$
  - Learning: use a labeled training set to estimate the most likely parameters of the model $\hat{\theta}$
  - Decoding: $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} P(\mathbf{x}, \mathbf{y}|\theta)$

Hidden Markov Models

A simplistic state diagram for noun phrases:

N – tags, M – vocabulary size



Example:

a hungry dog

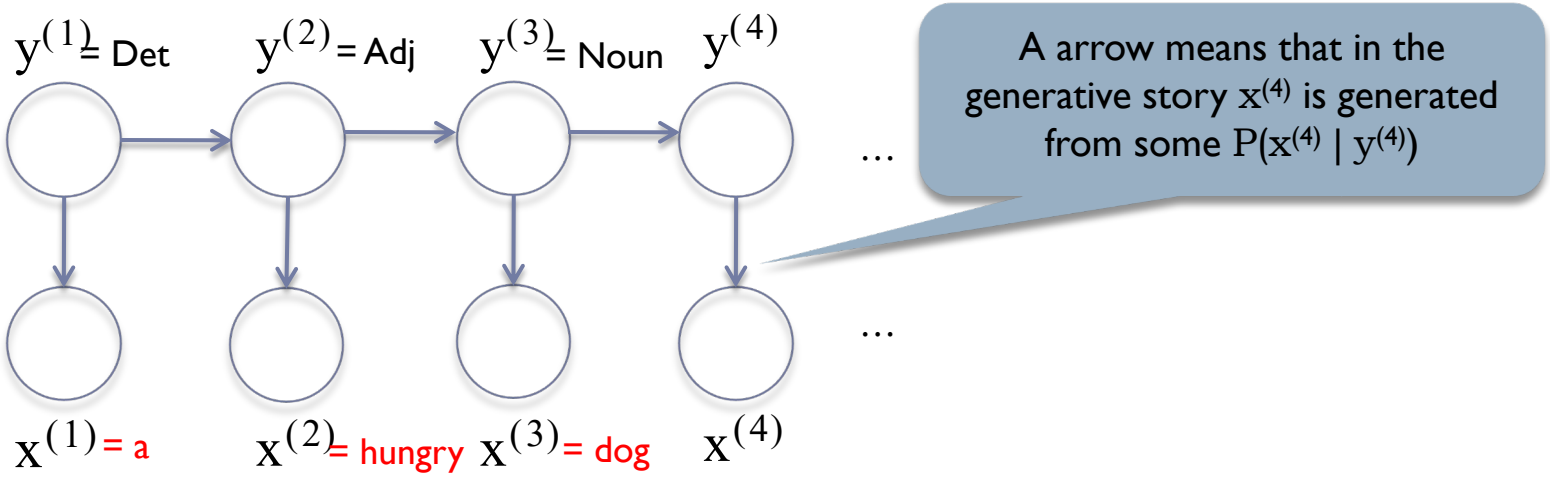
- States correspond to POS tags,
- Words are emitted independently from each POS tag

Stationarity assumption: this probability does not depend on the position in the sequence *t*

- Parameters (to be estimated from the training set):
  - Transition probabilities $P(y^t|y^{t-1})$: [N x N] matrix
  - Emission probabilities $P(x^t|y^t)$: [N x M] matrix

1st order Hidden Markov Models

Representation as an instantiation of *a graphical model:* N – tags, M – vocabulary size



- States correspond to POS tags,
- Words are emitted independently from each POS tag

Stationarity assumption: this probability does not depend on the position in the sequence *t*

- Parameters (to be estimated from the training set):
  - Transition probabilities $P(y^t|y^{t-1})$: [N x N] matrix
  - Emission probabilities $P(x^t|y^t)$: [N x M] matrix

Hidden Markov Models

- We will consider the part-of-speech (POS) tagging example

John	**carried**	**a**	**tin**	**can**	**.**
NNP	VBD	DT	NN	NN	.

- A "generative" model, i.e.:

  - Model: Introduce a parameterized model of how both words and tags are generated $P(\mathbf{x}, \mathbf{y}|\theta)$
  - Learning: use a labeled training set to estimate the most likely parameters of the model $\hat{\theta}$
  - Decoding: $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} P(\mathbf{x}, \mathbf{y}|\theta)$

We do not want to specify the transition system (associated probabilities) but learn it from the data

Hidden Markov Models: Estimation

- N – the number tags, M – vocabulary size
- Parameters (to be estimated from the training set):
  - Transition probabilities $a_{ij} = P(y^t = j \mid y^{t-1} = i)$, A - [N x N] matrix
  - Emission probabilities $b_{ik} = P(x^t = k \mid y^t = i)$, B - [N x M] matrix
- Training corpus:
  - (In, an, Oct., 19, review, of, ….), (IN, DT, NNP, CD, NN, IN, ….)
  - (Ms., Haag, plays, Elianti,.), (NNP, NNP, VBZ, NNP, .)
  - …
  - (The, company, said,…), (DT, NN, VBD, NNP, .)

Hidden Markov Models: Estimation

- N – the number tags, M – vocabulary size
- Parameters (to be estimated from the training set):
  - Transition probabilities $a_{ij} = P(y^t = j \mid y^{t-1} = i)$, A - [N x N] matrix
  - Emission probabilities $b_{ik} = P(x^t = k \mid y^t = i)$, B - [N x M] matrix
- Training corpus:
  - (In, an, Oct., 19, review, of, ….), (IN, DT, NNP, CD, NN, IN, ….)
  - (Ms., Haag, plays, Elianti,.), (NNP, NNP, VBZ, NNP, .)
  - …
  - (The, company, said,…), (DT, NN, VBD, NNP, .)

$$P(x^t = k|y^t = i) = b_{ik} = \frac{C_E(i,k)}{\sum_{k'} C_E(i,k')}$$

$C_E(i,k)$ is #times word k is emitted by tag i

Hidden Markov Models: Estimation

- N – the number tags, M – vocabulary size
- Parameters (to be estimated from the training set):
  - Transition probabilities $a_{ij} = P(y^t = j \mid y^{t-1} = i)$, A - [N x N] matrix
  - Emission probabilities $b_{ik} = P(x^t = k \mid y^t = i)$, B - [N x M] matrix
- Training corpus:
  - (In, an, Oct., 19, review, of, ….), (IN, DT, NNP, CD, NN, IN, ….)
  - (Ms., Haag, plays, Elianti,.), (NNP, NNP, VBZ, NNP, .)
  - …
  - (The, company, said,…), (DT, NN, VBD, NNP, .)

It is convenient to think that the output sequences are padded with START and STOP:

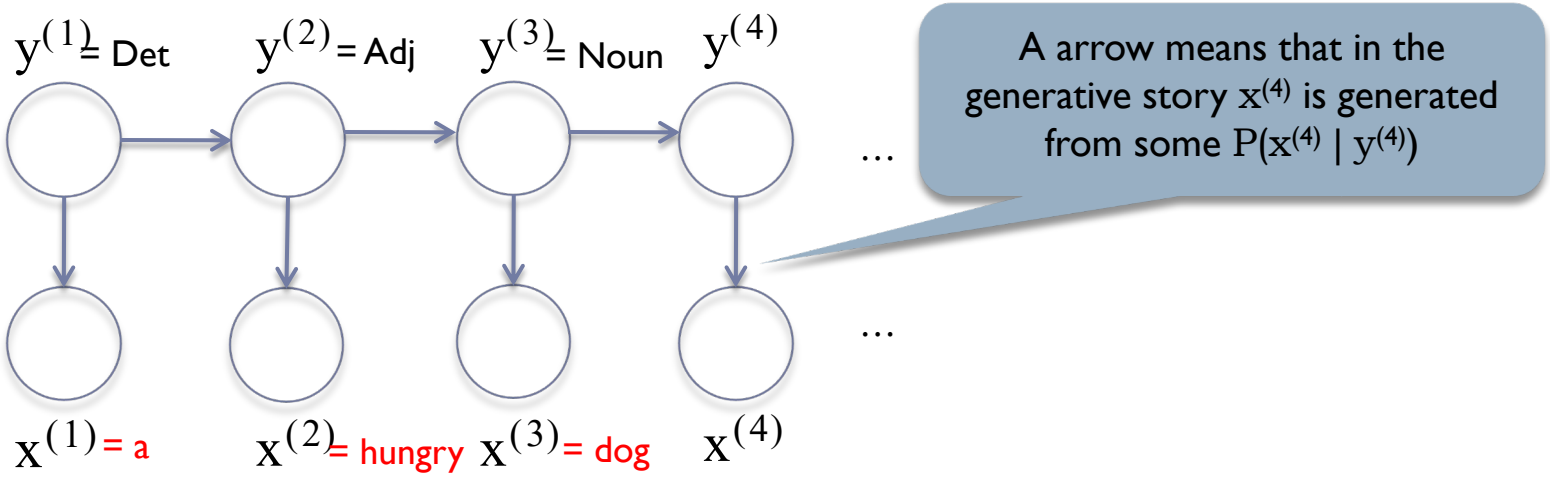
$$y^0 = START,\ y^{|x|+1} = STOP$$

$$P(y^t = j \mid y^{t-1} = i) = a_{ij} = \frac{C_T(i,j)}{\sum_{j'} C_T(i,j')}$$

$C_T(i,j)$ is #times tag i is followed by tag j

1st Order

Representation as an instantiation of a *graphical model*: N – tags, M – vocabulary size



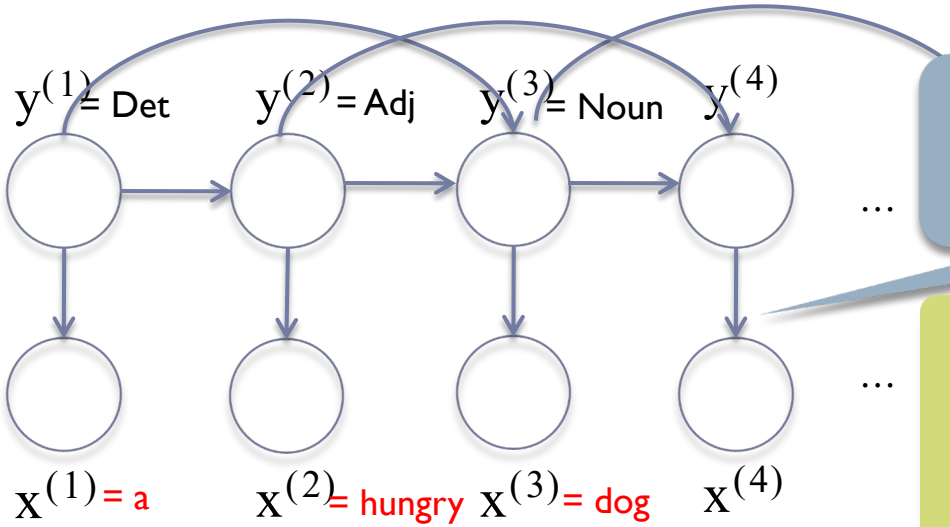
- HMM States correspond to POS tags,
- Words are emitted independently from each POS tag

Stationarity assumption: this probability does not depend on the position in the sequence *t*

- Parameters (to be estimated from the training set):
  - Transition probabilities $P(y^t|y^{t-1})$: [N x N] matrix
  - Emission probabilities $P(x^t|y^t)$: [N x M] matrix

2nd Order

Representation as an instantiation of a *graphical model*: N – tags, M – vocabulary size



A arrow means that in the generative story $x^{(4)}$ is generated from some $P(x^{(4)} \mid y^{(4)})$

The higher the order, the more zeros, the more important smoothing of the corresponding distribution is becoming

- HMM States correspond to POS tags,
- Words are emitted independently from each POS tag

Stationarity assumption: this probability does not depend on the position in the sequence *t*

- Parameters (to be estimated from the training set):
  - Transition probabilities $P(y^t|y^{t-1}, y^{t-2})$: [N x N x N] matrix
  - Emission probabilities $P(x^t|y^t)$: [N x M] matrix

Hidden Markov Models

- We will consider the part-of-speech (POS) tagging example

John	**carried**	**a**	**tin**	**can**	**.**
NNP	VBD	DT	NN	NN	.

- A "generative" model, i.e.:

  - Model: Introduce a parameterized model of how both words and tags are generated $P(\mathbf{x}, \mathbf{y}|\theta)$

  - Learning: use a labeled training set to estimate the most likely parameters of the model $\hat{\theta}$

    - Decoding: $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} P(\mathbf{x}, \mathbf{y}|\theta)$

Hidden Markov Models: decoding

- Predict PoS-tags for a sentence

John	carried	a	tin	can	.
?	?	?	?	?	?

- Corresponds to maximization:

$$\hat{\mathbf{y}} = \arg\max_{y} P(\mathbf{y}|\mathbf{x}, A, B) \quad = \arg\max_{y} P(\mathbf{y}, \mathbf{x}|\theta)$$

- Brute force
  - Consider all the sequences y and choose the highest scored one: $O(N^{|\mathbf{x}|})$

- A better alternative -- a dynamic programming algorithm, Viterbi $O(|\mathbf{x}|N^2)$

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

The probability of the most probable sequence up to *t* ending with a tag *i*

$$\max_{y_1 \ldots, y^{t-1}} P(x_1, \ldots, x_t, y_1, \ldots, y_t = i|\theta)$$

	time$_1$	*flies*$_2$	*fast*$_3$	-
NN				
VB				
JJ				
RB				
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \ldots, N;$

	time$_1$	*flies*$_2$	*fast*$_3$	-
NN	0.5x0.1=0.05			
VB				
JJ				
RB				
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \ldots, N;$

	time$_1$	*flies*$_2$	*fast*$_3$	-
NN	0.5x0.1=0.05			
VB	0.25x0.01=0.0025			
JJ				
RB				
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \ldots, N;$

	time$_1$	*flies*$_2$	*fast*$_3$	-
NN	0.5x0.1=0.05			
VB	0.25x0.01=0.0025			
JJ	0			
RB	0			
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \ldots, N;$

	time$_1$	*flies*$_2$	*fast*$_3$	-
NN	0.05			
VB	0.0025			
JJ	0			
RB	0			
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

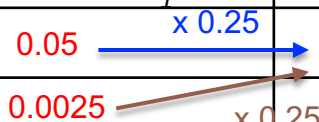
b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization:

$$v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \ldots, N;$$

$$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \ldots, N, \quad t = 2, \ldots, |x|$$

	time$_1$	*flies*$_2$	*fast*$_3$	-
NN	0.05			
VB	0.0025			
JJ	0			
RB	0			
STOP	-	-	-	



Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization:

$$v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \ldots, N;$$

$$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \ldots, N, \quad t = 2, \ldots, |x|$$

	time$_1$	*flies*$_2$	*fast*$_3$	-
NN	0.05	0.05 x 0.25 x 0.01		
VB	0.0025			
JJ	0			
RB	0			
STOP	-	-	-	

x 0.25

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization:

$$v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \ldots, N;$$

$$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \ldots, N, \quad t = 2, \ldots, |x|$$

store the backpointer

	time$_1$	*flies*$_2$	*fast*$_3$	-
NN	0.05	1.25E-4		
VB	0.0025			
JJ	0			
RB	0			
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization:

$$v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \ldots, N;$$

$$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \ldots, N, \quad t = 2, \ldots, |x|$$

	time$_1$	*flies*$_2$	*fast*$_3$	-
NN	0.05	1.25E-4		
VB	0.0025			
JJ	0			
RB	0			
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization:

$$v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \ldots, N;$$

$$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \ldots, N, \quad t = 2, \ldots, |x|$$

	time$_1$	*flies*$_2$	*fast*$_3$	-
NN	0.05	1.25E-4		
VB	0.0025	0.05 x 0.5 x 0.1		
JJ	0			
RB	0			
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization:

$$v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \ldots, N;$$

$$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \ldots, N, \quad t = 2, \ldots, |x|$$

	time$_1$	*flies*$_2$	*fast*$_3$	-
NN	0.05	1.25E-4		
VB	0.0025	0.0025		
JJ	0			
RB	0			
STOP	-	-	-	

again, store the backpointer

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization:

$$v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \ldots, N;$$

$$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \ldots, N, \quad t = 2, \ldots, |x|$$

	time$_1$	*flies*$_2$	*fast*$_3$	-
NN	0.05	1.25E-4	6.25E-6	
VB	0.0025	0.0025	6.25E-7	
JJ	0	0	6.25E-5	
RB	0	0	6.25E-5	
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization:

$$v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \ldots, N;$$

$$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \ldots, N, \quad t = 2, \ldots, |x|$$

Final:

$$v_{STOP}^{|\mathbf{x}|+1} = \max_i v_i^{|\mathbf{x}|} a_{i,STOP}$$

	time$_1$	*flies*$_2$	*fast*$_3$	-
NN	0.05	1.25E-4	6.25E-6	-
VB	0.0025	0.0025	6.25E-7	-
JJ	0	0	6.25E-5	-
RB	0	0	6.25E-5	-
STOP	-	-	-	

x 0.25 x 0.25 x 0.0 x 0.5

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization:

$$v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \ldots, N;$$

$$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \ldots, N, \quad t = 2, \ldots, |x|$$

Final:

$$v_{STOP}^{|\mathbf{x}|+1} = \max_i v_i^{|\mathbf{x}|} a_{i,STOP}$$

	time$_1$	*flies*$_2$	*fast*$_3$	-
NN	0.05	1.25E-4	6.25E-6	-
VB	0.0025	0.0025	6.25E-7	-
JJ	0	0	6.25E-5	-
RB	0	0	6.25E-5 x 0.5	-
STOP	-	-	-	6.25E-5 x 0.5

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization:

$$v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \ldots, N;$$

$$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \ldots, N, \quad t = 2, \ldots, |x|$$

Final:

$$v_{STOP}^{|\mathbf{x}|+1} = \max_i v_i^{|\mathbf{x}|} a_{i,STOP}$$

	time$_1$	*flies*$_2$	*fast*$_3$	-
NN	0.05	1.25E-4	6.25E-6	-
VB	0.0025	0.0025	6.25E-7	-
JJ	0	0	6.25E-5	-
RB	0	0	6.25E-5	-
STOP	-	-	-	3.125E-5

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization:

$$v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \ldots, N;$$

$$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \ldots, N, \quad t = 2, \ldots, |x|$$

Final:

$$v_{STOP}^{|\mathbf{x}|+1} = \max_i v_i^{|\mathbf{x}|} a_{i,STOP}$$

Retrace the backpointers to recover

The probability of the most probable tagged word sequence

	time$_1$	*flies*$_2$	*fast*$_3$	
NN	0.05	1.25E-4	6.25E-6	-
VB	0.0025	0.0025	6.25E-7	-
JJ	0	0	6.25E-5	-
RB	0	0	6.25E-5	-
STOP	-	-	-	3.125E-5

Hidden Markov Models: decoding

The most probable sequence can be recovered by back-tracing the pointers starting with "final":

NN VB RB

Final: $v_{STOP}^{|\mathbf{x}|+1} = \max_i v_i^{|\mathbf{x}|} a_{i,STOP}$

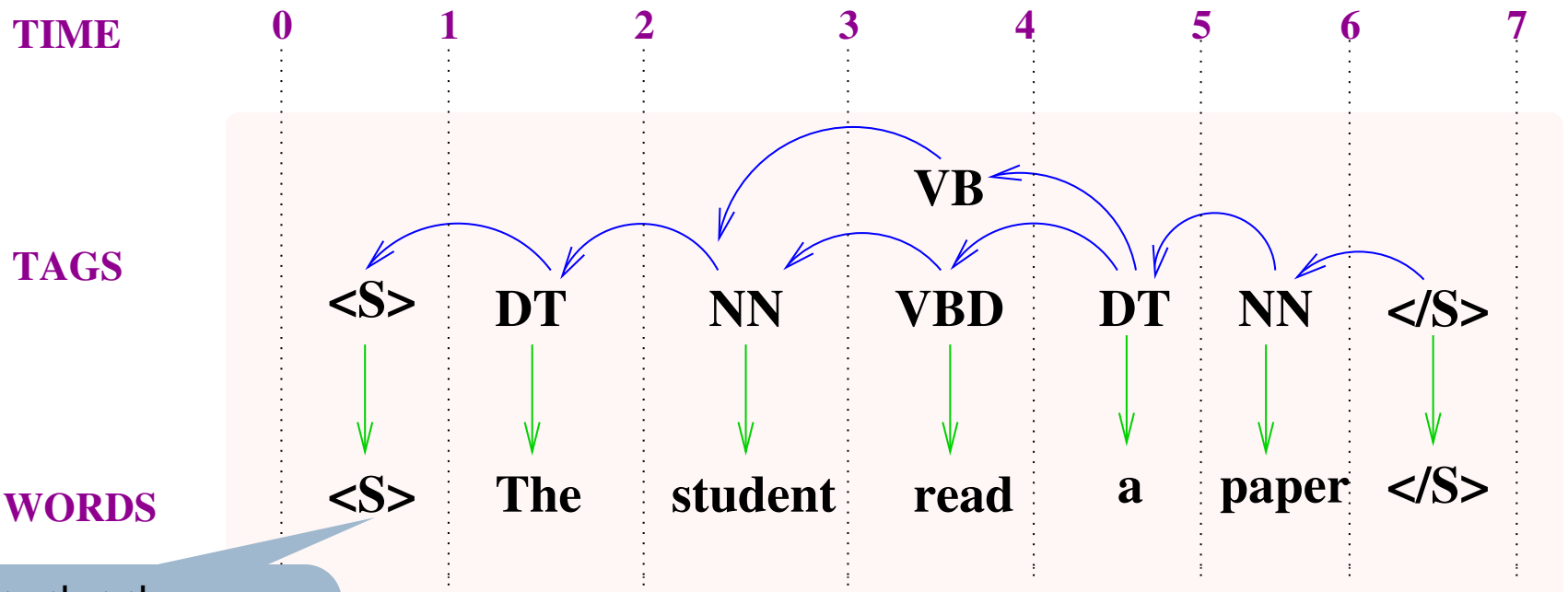
Retrace the backpointers to recover

The probability of the most probable tagged word sequence

	time$_1$	*flies*$_2$	*fast*$_3$	
NN	0.05	1.25E-4	6.25E-6	-
VB	0.0025	0.0025	6.25E-7	-
JJ	0	0	6.25E-5	-
RB	0	0	6.25E-5	-
STOP	-	-	-	3.125E-5

Viterbi Algorithm

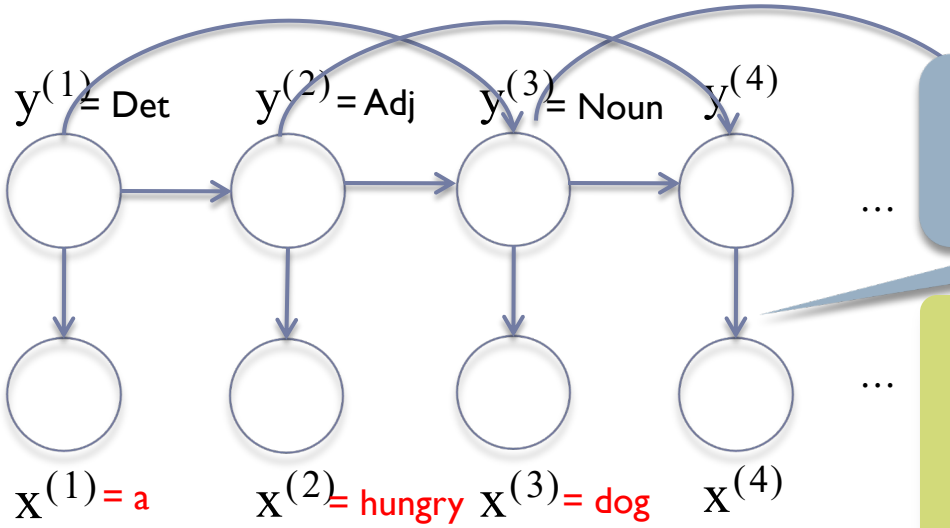
- A state lattice for a given sentence: we need to traverse only relevant states



Note that the sentence is padded with <s> and </s> symbols

2nd Order

Representation as an instantiation of a *graphical model*: N – tags, M – vocabulary size



A arrow means that in the generative story $x^{(4)}$ is generated from some $P(x^{(4)} \mid y^{(4)})$

What the state in the Viterbi lattice should be here? Is it just the previous PoS tag?

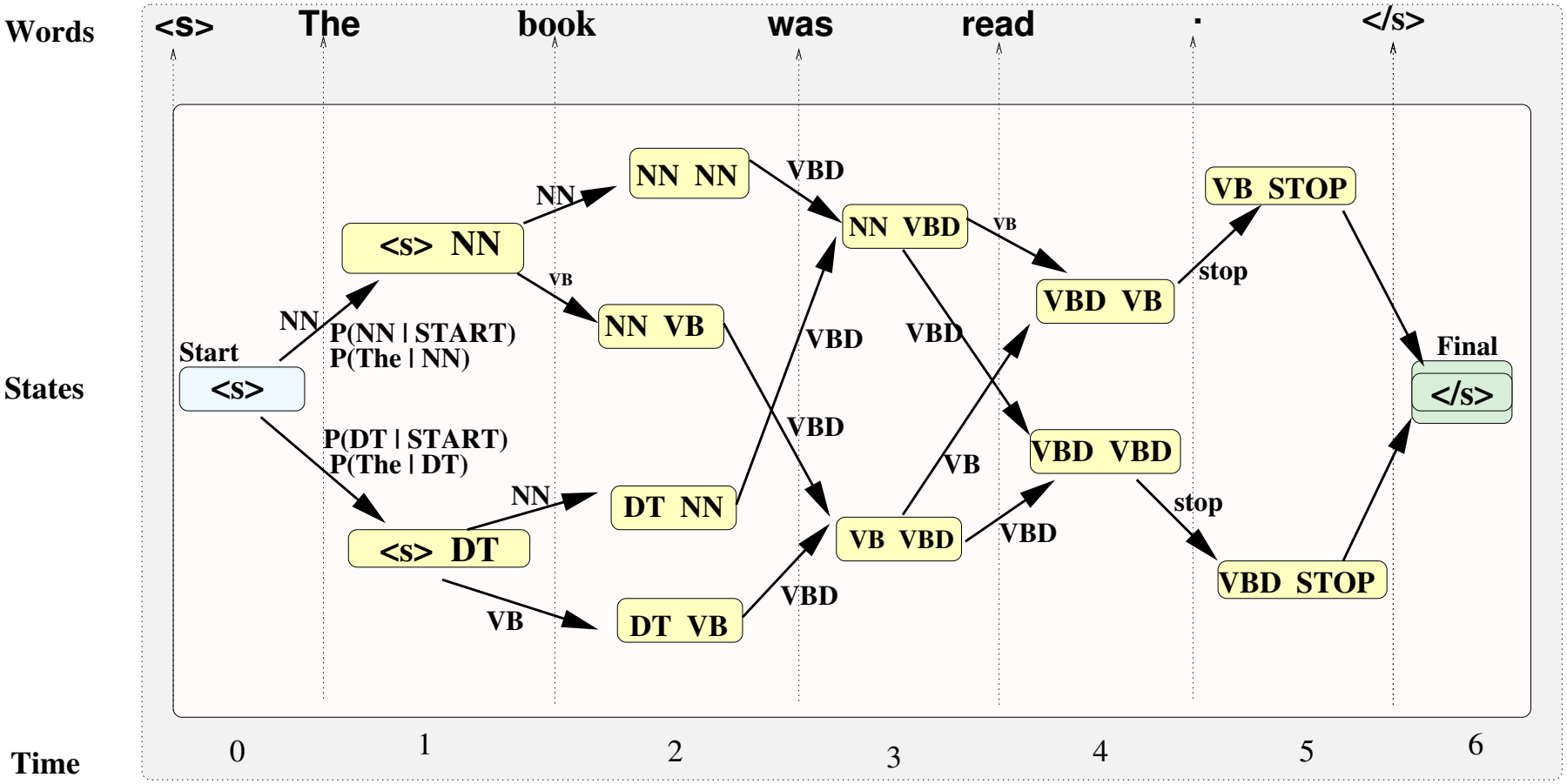
- HMM States correspond to POS tags,
- Words are emitted independently from each POS tag

Stationarity assumption: this probability does not depend on the position in the sequence *t*

- Parameters (to be estimated from the training set):
  - Transition probabilities $P(y^t|y^{t-1}, y^{t-2})$: [N x N x N] matrix
  - Emission probabilities $P(x^t|y^t)$: [N x M] matrix

Viterbi Algorithm for 2nd order model

- A state in the lattice now include both current and the previous state



Hidden Markov Models: decoding

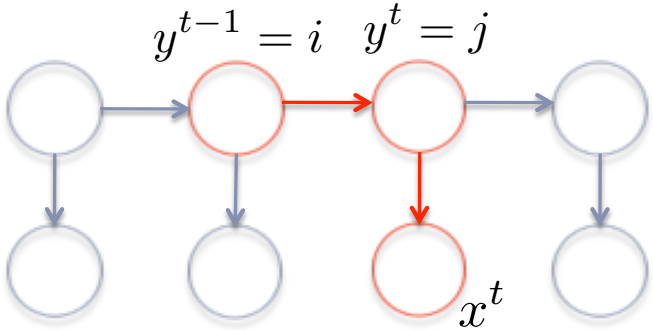
Initialization: $v_j^1 = a_{START,j} b_{j,x^1}, \quad j = 1, \ldots, N;$

Recomputation: $v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \; j = 1, \ldots, N, \; t = 2, \ldots, |x|$

Final: $v_{STOP}^{|\mathbf{x}|+1} = \max_i v_i^{|\mathbf{x}|} a_{i,STOP}$

Equivalently, in the log-space (and in a more generalized form):

Define: $g^t(\mathbf{x}, i, j) = \begin{cases} \log a_{ij} + \log b_{j,x^t} & t = 1, \ldots, |\mathbf{x}| \\ \log a_{ij} & t = |\mathbf{x}| + 1 \end{cases}$



The score is associated with a fragment responsible for one transition and one word generation

Initialization: $v_j^1 = g^1(\mathbf{x}, START, j) \quad j = 1, \ldots, N;$

Recomputation: $v_j^t = \max_i \left(v_i^{t-1} + g^t(\mathbf{x}, i, j) \right), \; j = 1, \ldots, N, \; t = 2, \ldots, |x|$

Final: $v_{STOP}^{|\mathbf{x}|+1} = \max_i \left(v_i^{|\mathbf{x}|} + g^{|x|+1}(\mathbf{x}, i, STOP) \right)$

Hidden Markov Models

- We will consider the part-of-speech (POS) tagging example

John	**carried**	**a**	**tin**	**can**	**.**
NNP	VBD	DT	NN	NN	.

- A "generative" model, i.e.:

- Model: Introduce a parameterized model of how both words and tags are generated $P(\mathbf{x}, \mathbf{y}|\theta)$

- Learning: use a labeled training set to estimate the most likely parameters of the model $\hat{\theta}$

- Decoding: $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} P(\mathbf{x}, \mathbf{y}|\theta)$

Is generative modeling necessarily the best approach to tagging? In the classification case, the Naïve Bayes was not necessarily the best choice

HMMs: what else?

- Using Viterbi, we can find the best tags for a sentence (decoding), and get $P(y, x|\theta)$
- We might also want to
  - compute the likelihood, i.e., the probability of a sentence regardless of its tags (a language model!) $P(x|\theta)$
  - learn the best set of parameters $\hat{\theta}$ given only an unannotated corpus of sentences.

Computing the likelihood

- From the probability theory we know

$$P(x|\theta) = \sum_y P(x, y|\theta)$$

- But there are an exponential number of sequences *y*
- Again, by computing and storing partial results, we can solve efficiently.

Computing the likelihood

- From the probability theory we know

$$P(x|\theta) = \sum_y P(x, y|\theta)$$

- But there are an exponential number of sequences *y*
- Again, by computing and storing partial results, we can solve efficiently.

Viterbi

Initialization: $v_j^1 = a_{START,j} b_{j,x^1}, \quad j = 1, \ldots, N;$

Recomputation: $v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \ j = 1, \ldots, N, \ t = 2, \ldots, |x|$

Final: $v_{STOP}^{|\mathbf{x}|+1} = \max_i v_i^{|\mathbf{x}|} a_{i,STOP}$

Forward algorithm

Initialization: $v_j^1 = a_{START,j} b_{j,x^1}, \quad j = 1, \ldots, N;$

Recomputation: $v_j^t = \left(\sum_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \ j = 1, \ldots, N, \ t = 2, \ldots, |x|$

Final: $v_{STOP}^{|\mathbf{x}|+1} = \sum_i v_i^{|\mathbf{x}|} a_{i,STOP}$

Let's revisit Naïve Bayes

- For Naïve Bayes, we discussed that we can use Self-Training to make use of unlabeled data
- We now, get to NB for a bit and see how we can fix it (Expectation Maximization, EM for NB)
- Then, we generalize EM for HMM
  - Consider even a harder case where we have only unlabeled data

Recap: self-training

		Bayes	your	model	cash	Viagra	class	orderz	spam?
labeled data	lab doc 1	0	1	3	0	0	2	0	-
	lab doc 2	0	2	0	4	0	0	0	+
	lab doc 3	0	2	2	0	0	3	0	-
	lab doc 4	0	3	2	1	3	0	1	+
	lab doc 5	0	1	0	2	0	0	1	+
unlabeled data	unlab doc 1	1	1	1	0	0	2	1	Labels missing
	unlab doc 2	2	2	0	0	0	0	0	
	unlab doc 3	0	1	0	0	1	0	1	

1. Train NB on labeled data alone
2. Predict labels on on unlabelled data
3. Re-estimate NB (in the usual way), but now using also self-labelled data

Recap: self-training

		Bayes	your	model	cash	Viagra	class	orderz	spam?
labeled data	lab doc 1	0	1	3	0	0	2	0	-
	lab doc 2	0	2	0	4	0	0	0	+
	lab doc 3	0	2	2	0	0	3	0	-
	lab doc 4	0	3	2	1	3	0	1	+
	lab doc 5	0	1	0	2	0	0	1	+
unlabeled data	unlab doc 1	1	1	1	0	0	2	1	-
	unlab doc 2	2	2	0	0	0	0	0	+
	unlab doc 3	0	1	0	0	1	0	1	+

mistake

Recap: self-training doesn't account for uncertainty

		Bayes	your	model	cash	Viagra	class	orderz	spam?
labeled data	lab doc 1	0	1	3	0	0	2	0	-
	lab doc 2	0	2	0	4	0	0	0	+
	lab doc 3	0	2	2	0	0	3	0	-
	lab doc 4	0	3	2	1	3			+
	lab doc 5	0	1	0	2	0			+
unlabeled data	unlab doc 1	1	1	1	0	0	2	1	
	unlab doc 2	2	2	0	0	0	0	0	+
	unlab doc 3	0	1	0	0	1	0	1	

mistake

Unlab doc 2:

$$\hat{P}(\text{spam}|d) \approx 0.53$$

The initial model was not confident in this prediction, but - on the re-estimation step - self-training does not account for this, and treats the label as "gold standard"

Summary: self-training

- Advantages:
  - Simplicity and applicable to any classifier (not only NB)
- Disadvantages:
  - Does not account for uncertainty of a classifier
  - No theoretical motivation (kind of…)
- To make it work, well requires
  - discarding low-confidence predictions
  - curriculum (start with examples similar to labeled data)
  - …

Also, self-training does not make much sense for completely unsupervised estimation (i.e. no labeled data = no sensible initial model)

Expectation Maximization

		Bayes	your	model	cash	Viagra	class	orderz	spam?
labeled data	lab doc 1	0	1	3	0	0	2	0	-
	lab doc 2	0	2	0	4	0	0	0	+
	lab doc 3	0	2	2	0	0	3	0	-
	lab doc 4	0	3	2	1	3	0	1	+
	lab doc 5	0	1	0	2	0	0	1	+
unlabeled data	unl doc 2	2	2	0	0	0	0	0	

Unlab doc 2:

$$\hat{P}(\text{spam}|d) \approx 0.53$$

Use soft label: 0.53 of the data point is labelled as "+", 0.47 as "-"

Expectation Maximization

		Bayes	your	model	cash	Viagra	class	orderz	spam?
labeled data	lab doc 1	0	1	3	0	0	2	0	-
	lab doc 2	0	2	0	4	0	0	0	+
	lab doc 3	0	2	2	0	0	3	0	-
	lab doc 4	0	3	2	1	3	0	1	+
	lab doc 5	0	1	0	2	0	0	1	+
unlabeled data	unl doc 2	2 x 0.53	2 x0.53	0	0	0	0	0	+ (.53)
		2 x 0.47	2 x 0.47	0	0	0	0	0	- (.47)

Unlab doc 2:

$$\hat{P}(\text{spam}|d) \approx 0.53$$

Use soft label: 0.53 of the data point is labelled as "+", 0.47 as "-"

Expectation Maximization

		Bayes	your	model	cash	Viagra	class	orderz	spam?
labeled data	lab doc 1	0	1	3	0	0	2	0	-
	lab doc 2	0	2	0	4	0	0	0	+
	lab doc 3	0	2	2	0	0	3	0	-
	lab doc 4	0	3	2	1	3	0	1	+
	lab doc 5	0	1	0	2	0	0	1	+
unlabeled data	unl doc 2	2 x 0.53	2 x0.53	0	0	0	0	0	+ (.53)
		2 x 0.47	2 x 0.47	0	0	0	0	0	- (.47)

$$\hat{P}(\text{your}|+) = (6 + 2 \times 0.53 + \alpha)/(20 + 4 \times 0.53 + \alpha * F)$$
$$\hat{P}(\text{your}|-) = (3 + 2 \times 0.47 + \alpha)/(13 + 3 \times 0.47 + \alpha * F)$$

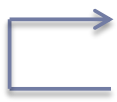
$$\hat{P}(\text{Bayes}|+) = (2 \times 0.53 + \alpha)/(20 + 4 \times 0.53 + \alpha * F)$$
$$\hat{P}(\text{Bayes}|-) = (2 \times 0.47 + \alpha)/(13 + 4 \times 0.47 + \alpha * F)$$

$$\hat{P}(\text{spam}) = \frac{3 + 0.53}{5 + 1}$$

This is just for one data point

EM for Semi-supervised Learning

1. Train NB on labeled data alone
2. Make soft prediction on on unlabelled data ("E-step")
3. Recompute NB parameters using the soft counts



We defined the method algorithmically, but it can be shown to optimize the likelihood of observed data (i.e. a combination labelled and unlabeled portions)

- EM is very general, and some of its generalizations (e.g., Variational Autoencoders / VAE) are standard tools in Deep Learning
- Self-training for NB is known as "hard EM"

justifying the name, "Expectation maximization"