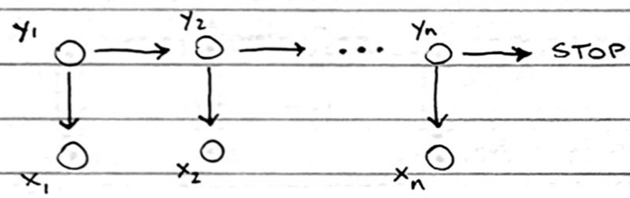
Hidden Markov Models

- Generative Sequence Model

Tags $y_i \in \mathcal{T}$ (tags) Words $x_i \in \mathcal{V}$ (vocabulary)

$$P(\bar{y}, \bar{x}) = P(y_1)\, P(x_1 | y_1)\, P(y_2 | y_1)\, P(x_2 | y_2) \ldots P(\text{stop} | y_n)$$

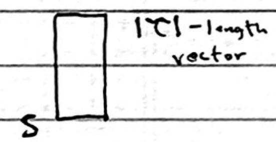


* y's form a <u>markov</u> process: y_i is conditionally independent of $y_1 \cdots y_{i-2}$ given y_{i-1}.

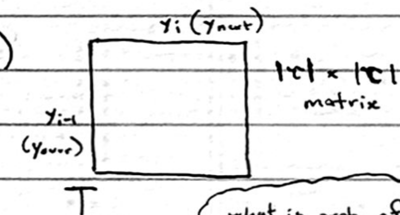
- <u>Two Steps to use model</u>
  1. Parameter Estimation
  2. Inference

Parameters

$P(y_1)$
initial distribution

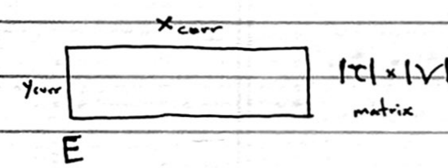


$P(y_i | y_{i-1})$
Transitions



what is prob. of y_{next} given y_{curr}

$P(x_i | y_i)$
Emissions



Given a tag (NN), what is prob of seeing each word in vocab? (fixed vocab)

y = tags
x = words

HMM's : Parameter Estimation

- Labeled data : $\left(\bar{x}^{(i)}, \bar{y}^{(i)}\right)_{i=1}^{D}$

unlike logistic regression which is $P(\bar{y} \mid x)$

- Maximize $\longrightarrow \sum_i \log P(\bar{y}^{(i)}, \bar{x}^{(i)})$ generative (joint) likelihood

$$= \underbrace{\sum_i \log P(y_1^{(i)})}_{\substack{\text{log prob. } y_1 \text{ for each sequence in} \\ \text{training data}}} + \underbrace{\sum_i \sum_j \log P(x_j^{(i)} \mid y_j^{(i)})}_{\substack{i = \text{sum over training data} \\ j = \text{sum over sentence index} \\ * \text{Accumulate prob of seeing that } x \text{ given that } y}} + \underbrace{\sum_i \sum_j \log P(y_j^{(i)} \mid y_{j-1}^{(i)})}_{\substack{\text{Accumulate probability of transition to } y_j \\ \text{given } y_{j-1}}}$$

- MLE with frequency counts

Biased coin w/ probability P of heads
- observe: HHHT
- what is maximum likelihood probability P for this coin? (value of P that maximizes data likelihood)
- 3/4 at first glance

$\operatorname{argmax}(3 \log P + \log(1-P)) = 3/4$ ✓ correct

* HMM parameter estimation doesn't involve gradient descent.
You can estimate parameters by counting and normalizing

Ex.

$\mathcal{Y} = \{N, V, \text{STOP}\}$ $\quad V = \{\text{They}, \text{can}, \text{fish}\}$

Data:

N	V	STOP
They	can	

N	V	STOP
They	fish	

$\underline{S} =$

N	2
V	0

$\downarrow$ normalize

N	1
V	0

$\underline{T} =$

	N	V	STOP
N	0	2	0
V	0	0	2

$\downarrow$ normalize

0	1	0
0	0	1

$\underline{E} =$

	They	can	fish
N	2	0	0
V	0	1	1

$\longrightarrow$

	T	C	F
N	1	0	0
V	0	1/2	1/2

$P(\text{fish} \mid V)$

HMM's Parameter Estimation (continued)

- <u>Smoothing</u>

  Add counts (fake data) to avoid 0's

$$T \longrightarrow \begin{array}{|c|c|c|}\hline 1 & 3 & 1 \\ \hline 1 & 1 & 3 \\ \hline \end{array} \xrightarrow[\text{normal.}]{} \begin{array}{ccc} 1/5 & 3/5 & 1/5 \\ 1/5 & 1/5 & 3/5 \end{array}$$

Ex)

$$\begin{array}{ccc} N & V & V \\ \text{They} & \text{can} & \text{fish} \end{array} \Longrightarrow P(\bar{y}, \bar{x}) = \begin{array}{cccc} y_1 & y_2|y_1 & y_3|y_2 & s|y_3 \\ x_1|y_1 & x_2|y_2 & x_3|y_3 & \end{array}$$

$$\downarrow$$

$$\begin{array}{cccc} 1 & 3/5 & 1/5 & 3/5 \\ 1 & 1/2 & 1/2 & \end{array}$$

HMM's : Viterbi Algorithm

- HMM's : model of $P(\bar{y}, \bar{x}) = P(y_1)\, P(x_1 | y_1)\, P(y_2 | y_1) \dots$
- Inference : $\underset{\bar{y}}{\mathrm{argmax}}\; p(\bar{y} | \bar{x}) \rightarrow$ Given a sentence, what is most likely POS tag sequence that could've produced that sentence

$$\underset{\bar{y}}{\mathrm{argmax}}\; P(\bar{y} | \bar{x}) = \underset{\bar{y}}{\mathrm{argmax}}\; \frac{P(\bar{y}, \bar{x})}{\underset{\substack{\text{constant} \\ \text{w.r.t. } \bar{y}}}{P(\bar{x})}} = \underset{\bar{y}}{\mathrm{argmax}}\; \log P(\bar{y}, \bar{x})$$

$$= \underset{\hat{y}_1 \dots \hat{y}_n}{\mathrm{argmax}}\; \log P(\hat{y}_1) + \log P(x_1 | \hat{y}_1) + \log(\hat{y}_2 | \hat{y}_1) + \dots$$

- <u>Viterbi Dynamic Program</u>

Define $v_i(\hat{y}) = n \times |\Upsilon|$ n = sentence length, $|\Upsilon|$ = number of tags

score of best path ending in $\hat{y}$ at time i

Base: $v_1(\hat{y}) = \log P(x_1 | \hat{y}) + \log P(\hat{y})$

Recurrence: $v_i(\hat{y}) = \log P(x_i | \hat{y}) + \max_{\hat{y}_{prev}} \log P(\hat{y} | \hat{y}_{prev}) + v_{i-1}(\hat{y}_{prev})$

<u>Viterbi</u>

for $i = 1 \dots n$:

    for $\hat{y}$ in $|\Upsilon|$:

        compute $v_i(\hat{y})$

Compute $v_{n+1}(\text{STOP})$, this $= \max_{\bar{y}} \log P(\bar{x}, \bar{y})$

Track "backpointers"

$\boxed{\text{Ex.}}$

$S =$

N	-1
V	-1

$T =$ (log probs)

	N	V	STOP
N	-2	-1	-1
V	-1	-1	-2

$E =$

	They	con	fish
N	-1	-3	-1
V	-3	-1	-1

$v_i(\hat{y})$

	they	can	can	fish	STOP
N	-2	$-2-2-3=\boxed{-7}$ $-4-1-3=-8$			
V	-4	$-2-1-1=\boxed{-4}$			

STOP

$\underset{\text{prev}}{-2}\; \underset{\text{tr}}{-2}\; \underset{\text{em}}{-3} = -7$