Sistemas Inteligentes

Hidden Markov Models(Columbia-Collins)

José Eduardo Ochoa Luna
Dr. Ciencias - Universidade de São Paulo

Maestría C.C. Universidad Católica San Pablo
Sistemas Inteligentes

6 de diciembre 2018

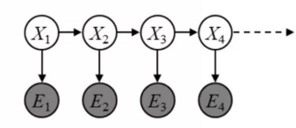
Reasoning over Time or Space

We want to reason about a sequence of observations

- Speech recognition
- Robot localization
- Medical monitoring
- Machine translation

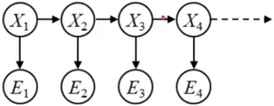
Hidden Markov Models

- Underlying Markov chain over states S
- You observe outputs (effects) at each time step
- $P(X_1), P(X_t|X_{t-1})$ and $P(E_t|X_t)$



Properties

- Hidden Markov process, future depends on past via the present (e.g. if X_3 is observed then $X_2 \perp X_4$)
- Current observation independent of all else given current state (e.g. if X_2 is observed then $E_2 \perp Z$)



HMM Examples

Speech Recognition HMMs:

- Observations are acoustic signals
- States are specific positions in specific words

Machine Translation HMMs:

- Observations are words
- States are translation options

Robot Tracking:

- Observations are range readings
- States are positions on a map

HMM for Tagging

Part-of-Speech Tagging

INPUT:
Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.

OUTPUT:
Profits/N soared/V at/P Boeing/N Co./N ,/, easily/ADV topping/V forecasts/N on/P Wall/N Street/N ,/, as/P their/POSS CEO/N Alan/N Mulally/N announced/V first/ADJ quarter/N results/N ./.

N	=	Noun
V	=	Verb
P	=	Preposition
Adv	=	Adverb
Adj	=	Adjective

...

Named Entity Recognition

INPUT: Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.

OUTPUT: Profits soared at [Company Boeing Co.], easily topping forecasts on [Location Wall Street], as their CEO [Person Alan Mulally] announced first quarter results.

Named Entity Recognition as Tagging

INPUT:

Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.

OUTPUT:

Profits/NA soared/NA at/NA Boeing/SC Co./CC ,/NA easily/NA topping/NA forecasts/NA on/NA Wall/SL Street/CL ,/NA as/NA their/NA CEO/NA Alan/SP Mulally/CP announced/NA first/NA quarter/NA results/NA ./NA

NA = No entity
SC = Start Company
CC = Continue Company
SL = Start Location
CL = Continue Location
...

The Goal

Training set:

1 Pierre/NNP Vinken/NNP ,/, 61/CD years/NNS old/JJ ,/, will/MD join/VB the/DT board/NN as/IN a/DT nonexecutive/JJ director/NN Nov./NNP 29/CD ./.

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

3 Rudolph/NNP Agnew/NNP ,/, 55/CD years/NNS old/JJ and/CC chairman/NN of/IN Consolidated/NNP Gold/NNP Fields/NNP PLC/NNP ,/, was/VBD named/VBN a/DT nonexecutive/JJ director/NN of/IN this/DT British/JJ industrial/JJ conglomerate/NN ./.

...

38,219 It/PRP is/VBZ also/RB pulling/VBG 20/CD people/NNS out/IN of/IN Puerto/NNP Rico/NNP ,/, who/WP were/VBD helping/VBG Huricane/NNP Hugo/NNP victims/NNS ,/, and/CC sending/VBG them/PRP to/TO San/NNP Francisco/NNP instead/RB ./.

From the training set, induce a function / algorithm that maps new sentences to their tag sequences

$x^{(1)}$ = The dog laughs, $y^{(1)}$ = DT NN VB

Two Types of Constraints

Influential/JJ members/NNS of/IN the/DT House/NNP Ways/NNP and/CC Means/NNP Committee/NNP introduced/VBD legislation/NN that/WDT would/MD restrict/VB how/WRB the/DT new/JJ savings-and-loan/NN bailout/NN agency/NN can/MD raise/VB capital/NN ./.

- Local:e.g. *can* is more likely to be a modal verb MD rather than a noun NN
- Contextual: e.g., a noun is much more likely than a verb to follow a determiner (DT NN)
- Sometimes these preferences are in conflict: *The trash can is in the garage*

Supervised Learning Problem

- We have training examples $x^{(i)}, y^{(i)}$ for $i = 1 \dots m$. Each $x^{(i)}$ is an input, each $y^{(i)}$ is a label
- Task is to learn a function f mapping inputs x to labels $f(x)$
- Conditional models:
  - Learn a distribution $p(y|x)$ from training examples
  - For any test input x, define $f(x) = \arg\max_y p(y|x)$

Decoding with Generative Models

- We have training examples $x^{(i)}, y^{(i)}$ for $i = 1 \ldots m$. Task is to learn a function f mapping inputs x to labels $f(x)$
- Generative models:
  - Learn a distribution $p(x, y)$ from training examples
  - $p(x, y) = p(y)p(x|y)$
- Output from the model:

$$\begin{aligned} f(x) &= \arg\max_y p(y|x) \\ &= \arg\max_y \frac{p(y)p(x|y)}{p(x)} (p(x) \text{ does not vary with y}) \\ &= \arg\max_y p(y)p(x|y) \end{aligned}$$

Trigram HMM

Hidden Markov Models

- We have an input sentence $x = x_1, x_2, \ldots, x_n$ (x_i is the i'th word in the sentence)
- We have a tag sequence $y = y_1, y_2, \ldots, y_n$ (y_i is the i'th tag in the sentence)
- We'll use an HMM to define

$$p(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n)$$

  for any sentence $x_1 \ldots x_n$ and tag sequence $y_1 \ldots y_n$ of the same length
- Then the most likely tag sequence for x is

$$\arg\max_{y_1 \ldots y_n} p(x_1, \ldots, x_n, y_1, y_2 \ldots, y_n)$$

Trigram HMM

For any sentence $x_1 \ldots x_n$ where $x_i \in \mathcal{V}$ for $i = 1 \ldots n$, and any tag sequence $y_1 \ldots y_{n+1}$, where $y_i \in \mathcal{S}$ for $i = 1 \ldots n$, and $y_{n+1} =$STOP (e.g. $\mathcal{V} = \{the, dog, cat, \ldots\}$, $\mathcal{S} =$DT,NN,P,ADV,...), the joint probability of the sentence and tag sequence is

$$p(x_1, \ldots, x_n, y_1, \ldots, y_{n+1}) = \prod_{i=1}^{n+1} q(y_i | y_{i-2}, y_{i-1}) \prod_{i=1}^{n} e(x_i | y_i)$$

We have assumed that $y_0 = y_{-1} = *$
Parameters of the model:

- $q(s|u, v)$ for any $s \in \mathcal{S} \cup$ STOP, $u, v \in \mathcal{S} \cup *$
- $e(x|s)$ for any $s \in \mathcal{S}, x \in \mathcal{V}$

Example

If we have $n = 3$, $x_1 \dots x_3$ equal to the sentence *the dog laughs*, and $y_1 \dots y_4$ equal to the tag sequence D N V STOP, then

$$\begin{aligned} & p(x_1, \dots, x_n, y_1, \dots, y_{n+1}) \\ = \; & q(D|*,*) \times q(N|*,D) \times q(V|D,N) \times q(STOP|N,V) \\ & \times e(the|D) \times e(dog|N) \times e(laughs|V) \end{aligned}$$

- STOP is a special tag that terminates the sequence
- We take $y_0 = y_{-1} = *$, where * is a special padding symbol

Parameters estimation

Smoothed Estimation

$$
\begin{aligned}
q(V_t|DT, JJ) \quad = \quad & \lambda_1 \times \frac{Count(DT, JJ, V_t)}{Count(DT, JJ)} \\
& \lambda_2 \times \frac{Count(JJ, V_t)}{Count(JJ)} \\
& \lambda_3 \times \frac{Count(V_t)}{Count()}
\end{aligned}
$$

$\lambda_1 + \lambda_2 + \lambda_3$, and for all $i, \lambda_i \geq 0$

$$
e(base|V_t) = \frac{Count(V_t, base)}{Count(V_t)}
$$

Dealing with Low-Frequency Words

$e(rare|y) = 0$ for all tags y then $p(x, y) = 0$ for all tag sequences $y_1 \cdots y_{n+1}$

- Step 1: Split vocabulary into two sets: a) frequent words (words occurring ≥ 5 times in training) ; b) low frequency words (all other words)
- Step 2: map low frequency words into a small, finite set, depending on prefixes, suffixes, etc

Dealing with Low-Frequency Words

Profits/NA soared/NA at/NA Boeing/SC Co./CC ,/NA easily/NA topping/NA forecasts/NA on/NA Wall/SL Street/CL ,/NA as/NA their/NA CEO/NA Alan/SP Mulally/CP announced/NA first/NA quarter/NA results/NA ./NA

⇓

firstword/NA soared/NA at/NA initCap/SC Co./CC ,/NA easily/NA lowercase/NA forecasts/NA on/NA initCap/SL Street/CL ,/NA as/NA their/NA CEO/NA Alan/SP initCap/CP announced/NA first/NA quarter/NA results/NA ./NA

NA	= No entity
SC	= Start Company
CC	= Continue Company
SL	= Start Location
CL	= Continue Location
. . .	

$e(\text{firstword} \mid NA)$

$e(\text{initCap} \mid SC)$

The Viterbi Algorithm

The Viterbi Algorithm

Problem: for an input $x_1 \dots x_n$, find

$$\arg \max_{y_1 \dots y_{n+1}} p(x_1, \dots, x_n, y_1, \dots, y_{n+1})$$

where the arg max is taken over all sequences $y_1 \dots y_{n+1}$ such that $y_i \in \mathcal{S}$ (e.g D,N,V) for $i = 1 \dots n$, and $y_{n+1} = \text{STOP}$
We assume that p again takes the form

$$p(x_1, \dots, x_n, y_1, \dots, y_{n+1}) = \prod_{i=1}^{n+1} q(y_i | y_{i-2}, y_{i-1}) \prod_{i=1}^{n} e(x_i | y_i)$$

Recall that $y_0 = y_{-1} = *$, and $y_{n+1} =$STOP.

Brute Force Search

Problem: for an input $x_1 \dots x_n$, find

$$\arg \max_{y_1 \dots y_{n+1}} p(x_1, \dots, x_n, y_1, \dots, y_{n+1})$$

where the arg max is taken over all sequences $y_1 \dots y_{n+1}$ such that $y_i \in \mathcal{S}$ (e.g D,N,V)for $i = 1 \dots n$, and $y_{n+1} = \text{STOP}$

$$\begin{array}{lcl} \text{the dog laughs} & \rightarrow & \text{D D D STOP}{=}0.3 \\ & \rightarrow & \text{D D N STOP}{=}0.01 \\ & \rightarrow & \text{D D V STOP}{=}0.0001 \\ & \dots. & \end{array}$$

General Case $|\mathcal{S}|^n$

The Viterbi Algorithm

- Define n to be the length of the sentence
- Define $\mathcal{S}_k$ for $k = -1 \ldots n$ to be the set of possible tags at position k:

$$\mathcal{S}_{-1} = \mathcal{S}_0 = \{*\}$$
$$\mathcal{S}_k = \mathcal{S} \text{ for } k \in \{1 \ldots n\}$$

- Define

$$r(y_{-1}, y_0, y_1, \ldots, y_k) = \prod_{i=1}^{k} q(y_i | y_{i-2}, y_{i-1}) \prod_{i=1}^{k} e(x_i | y_i)$$

- Define a dynamic programming table

$$\pi(k, u, v) = \text{max probability of a tag sequence ending in tags } u, v \text{ at position } k$$

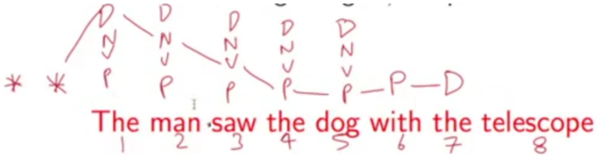
That is,

$$\pi(k, u, v) = \max_{\langle y_{-1}, y_0, y_1, \ldots, y_k \rangle : y_{k-1} = u, y_k = v} r(y_{-1}, y_0, y_1, \ldots, y_k)$$

An Example

$$\pi(k, u, v) = \text{max probability of a tag sequence ending in tags } u, v \text{ at position } k$$

$$\mathcal{S} = \{D, N, V, P\}$$



$\pi(7, P, D)$

A Recursive Definition

- Base case: $\pi(0, *, *) = 1$
- Rec. definition: for any $k \in \{1 \ldots n\}$, for any $u \in \mathcal{S}_{k-1}$ and $v \in \mathcal{S}_k$:

$$\pi(k, u, v) = \max_{w \in \mathcal{S}_{k-2}} \left(\pi(k-1, w, u) \times q(v|w, u) \times e(x_k|v) \right)$$

Justification for Recursive Definition

For any $k \in \{1 \ldots n\}$, for any $u \in \mathcal{S}_{k-1}$ and $v \in \mathcal{S}_k$:

$$\pi(k, u, v) = \max_{w \in \mathcal{S}_{k-2}} \left(\pi(k-1, w, u) \times q(v|w, u) \times e(x_k|v)\right)$$

$\pi(6, N, P)^*$
$\times q(D|N, P)$
$\times e(\text{the}|D)$

D
$\boxed{N}$
V
P   P   D

The man saw the dog with the telescope
1 2 3 4 5 6 7 8

$$\pi(7, P, D) = \max_{w \in \{D,V,N,P\}} \left(\pi(6, w, P) \times q(D|w, P) \times e(\text{the}|D)\right)$$

The Viterbi Algorithm

Input: a sentence $x_1 \ldots x_n$, parameters $q(s|u,v)$ and $e(x|s)$
Initialization: Set $\pi(0,*,*) = 1$
Definition: $\mathcal{S}_{-1} = \mathcal{S}_0 = \{*\}, \mathcal{S}_k = \mathcal{S}$ (e.g. D,N,V,P) for $k \in \{1 \ldots n\}$
Algorithm:

- For $k = 1 \ldots n$
  - For $u \in \mathcal{S}_{k-1}, v \in \mathcal{S}_k$,

$$\pi(k,u,v) = \max_{w \in \mathcal{S}_{k-2}} \left(\pi(k-1,w,u) \times q(v|w,u) \times e(x_k|v)\right)$$

- Return $\max_{u \in \mathcal{S}_{n-1}, v \in \mathcal{S}_n} \left(\pi(n,u,v) \times q(STOP|u,v)\right)$

The Viterbi Algorithm with Backpointers ($O(n|S|^3)$)

Input: a sentence $x_1 \dots x_n$, parameters $q(s|u,v)$ and $e(x|s)$
Initialization: Set $\pi(0,*,*) = 1$
Definition: $\mathcal{S}_{-1} = \mathcal{S}_0 = 0$, $\mathcal{S}_k = \mathcal{S}$ (e.g. {D,N,V,P}) for $k \in 1 \dots n$
Algorithm:

- For $k = 1 \dots n$
  - For $u \in \mathcal{S}_{k-1}, v \in \mathcal{S}_k$,

$$\pi(k,u,v) = \max_{w \in \mathcal{S}_{k-2}} (\pi(k-1,w,u) \times q(v|w,u) \times e(x_k|v))$$

$$bp(k,u,v) = \arg\max_{w \in \mathcal{S}_{k-2}} (\pi(k-1,w,u) \times q(v|w,u) \times e(x_k|v))$$

- Set $(y_{n-1}, y_n) = \arg\max_{(u,v)} (\pi(n,u,v) \times q(STOP|u,v))$
- For $k = (n-2) \dots 1$, $y_k = bp(k+2, y_{k+1}, y_{k+2})$
- Return the tag sequence $y_1 \dots y_n$

Tarea

- Crear un Jupyter notebook e implementar HMM para realizar Named Entity Recognition en Español
- Utilizar el dataset CoNLL-2002 (https://www.clips.uantwerpen.be/conll2002/ner/)