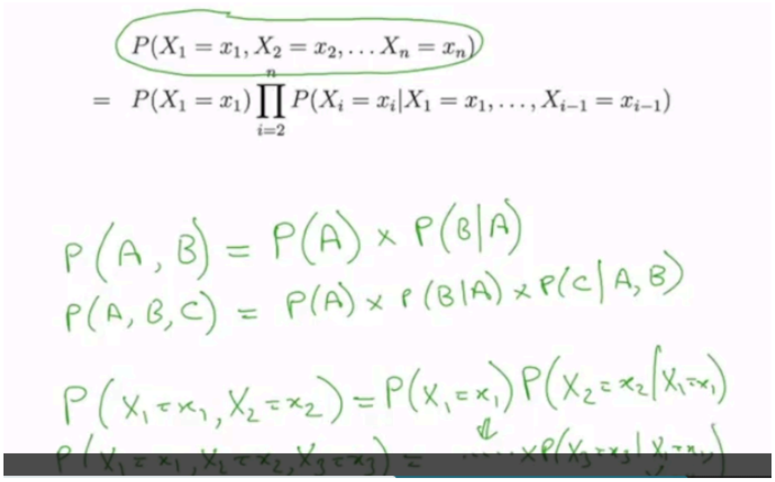
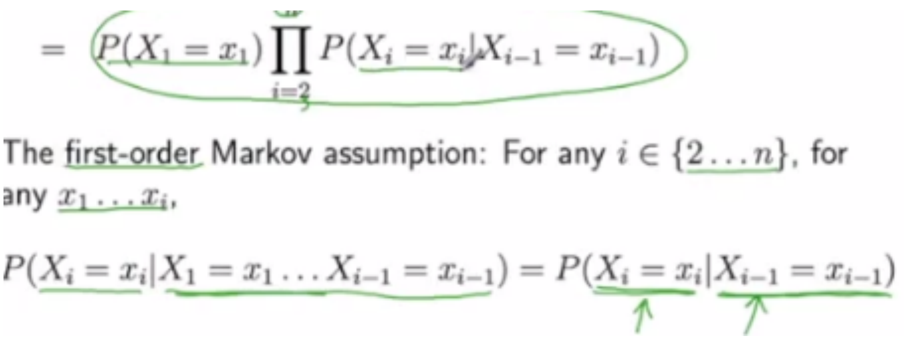
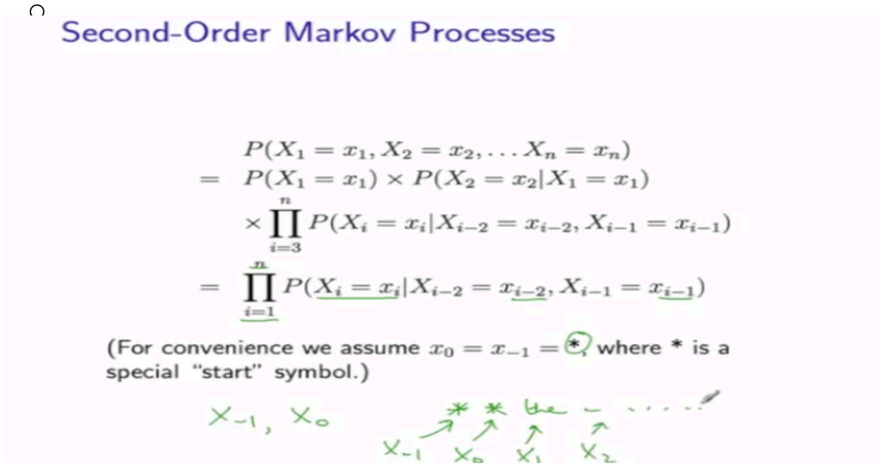
- Natural Language Processing
- The Language Modeling Problem
  - We have some (finite) vocabulary, say V = {the, a, man, telescope …}
  - We have an (infinite) set of strings, V^t
    - The STOP
    - The fan STOP
  - We have a training sample of example sentences in English
  - We need to "learn" a probability distribution p over the sentences in our language
    - Summation x element sentences of language
    - P(x) = 1, p(x) >= 0 for all x element of sentences in language
  - Assign a probability to every sentence in the language
- Why on earth would we want to do this?
  - Speech recognition was the original motivation (related problems are optical character recognition, handwriting recognition.)
  - The estimation techniques developed for this problem will be very useful for other problems in NLP
- A Naïve Method
  - We have N training sentences
  - For any sentence x1…xn c(x1…xn) is the number of times the sentence is seen in our training data
  - A naïve estimate
    - P(x1…xn) = c(x1…xn) / N
- Trigram models
- Markov Processes
  - Consider a sequence of random variables X1, X2, … Xn each random variable can take any value in a finite set V. For now we assume the length n is fixed (e.g., n = 100)
  - Markov process with states V = {0,1,2} and length n = 10
    - Then 3^10 sequences can be generated
- Our goal: model
  - P(X1 = x1, X2 = x2 … , Xn = xn)
- First-Order Markov Processes

- 

- First order Markov assumption: For any I element {2,….n} for any x1…xi
  - P(Xi = xi|X1 = x1 … Xi-1 = xi-1) = P(Xi = xi | Xi-1 = xi-1)
  - 

- Second Order Markov Processes
  - 

- Modeling Variable Length Sequences
  - We would like the length of the sequence, n, to also be a random variable
  - A simple solution: always define X_n = STOP where STOP is a special symbol
  - Then use a Markov process as before

- Generating the value of I’th random variable on the two previous conditions

$X_n = STOP$

$$P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n)$$
$$= \prod_{i=1}^{n} P(X_i = x_i | X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1})$$

(For convenience we assume $x_0 = x_{-1} = *$, where $*$ is a special “start” symbol.)

- Trigram language Models
  - A trigram language model consists of
    - A finite set V vocabulary in the language model
    - A parameter q(w|u,v) for each trigram u,v,w such that w element V U {STOP}, and u,v element V U {*}