Online Markov Decoding:

Lower Bounds and Near-Optimal Approximation Algorithms

Vikas K. Garg, Tamar Pichkhadze

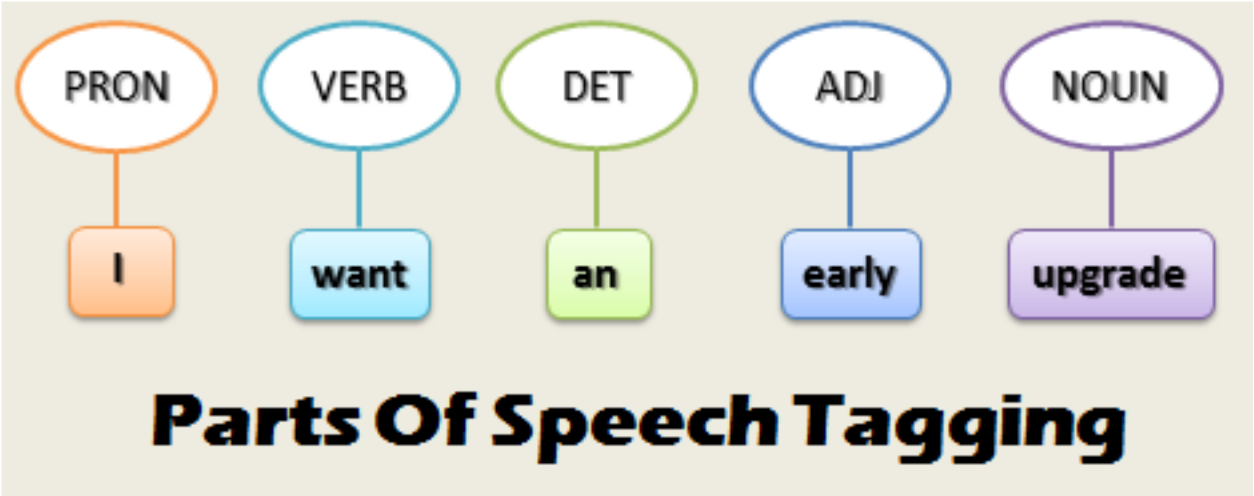
MIT

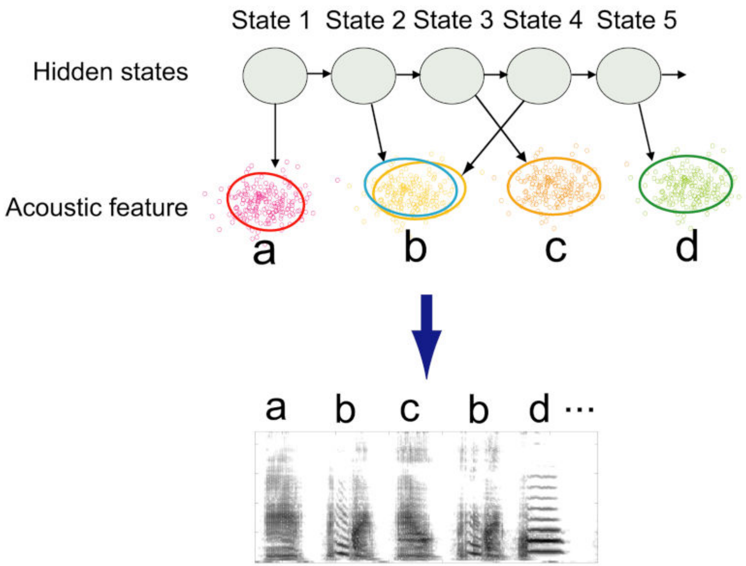
What is Markov chain decoding?

- Given a sequence of observations $x = (x_1, x_2, \ldots, x_T)$
- Assume x generated by state sequence $y = (y_1, y_2, \ldots, y_T)$
- Each state y_i takes value in a discrete set and *emits* x_i
- We do not know y but would like to infer it from x
- Markov chain of order k:

  y_i depends on only previous states $y_{i-1}, \ldots, y_{i-k}$

Markov chain models are ubiquitous!





- Bioinformatics (e.g. gene sequencing)
- Computer Vision (e.g. gesture recognition)
- Telecommunication (e.g. convolutional codes)
- Language processing (e.g. named entity recognition)
- Computer Networks (e.g. intrusion detection)
- Speech recognition …

… … …. … …..

Images sourced from (a) ThinkInfi blog, (b) Devopedia.org (adapted from Katahira et al.)

First Order Markov chain models

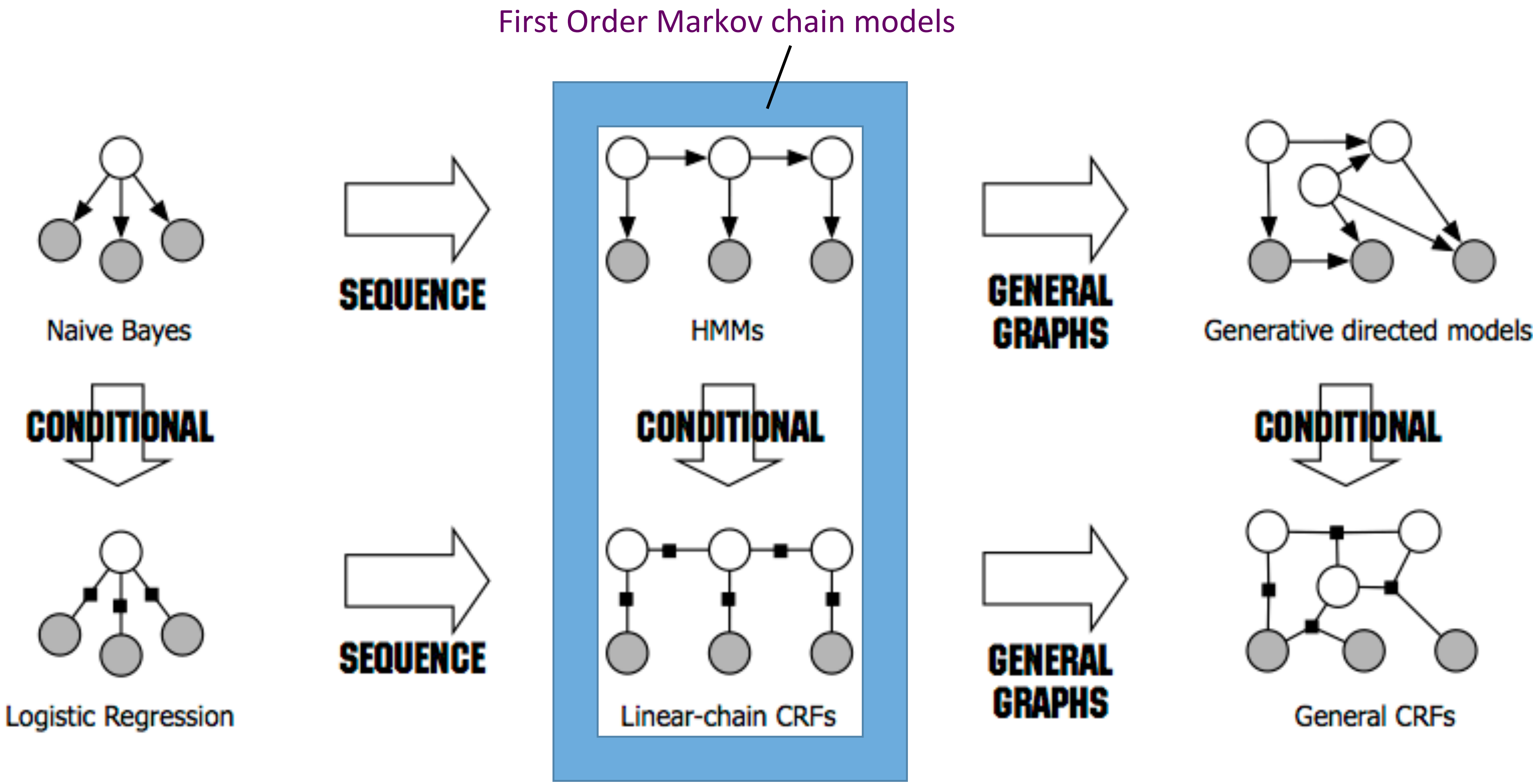
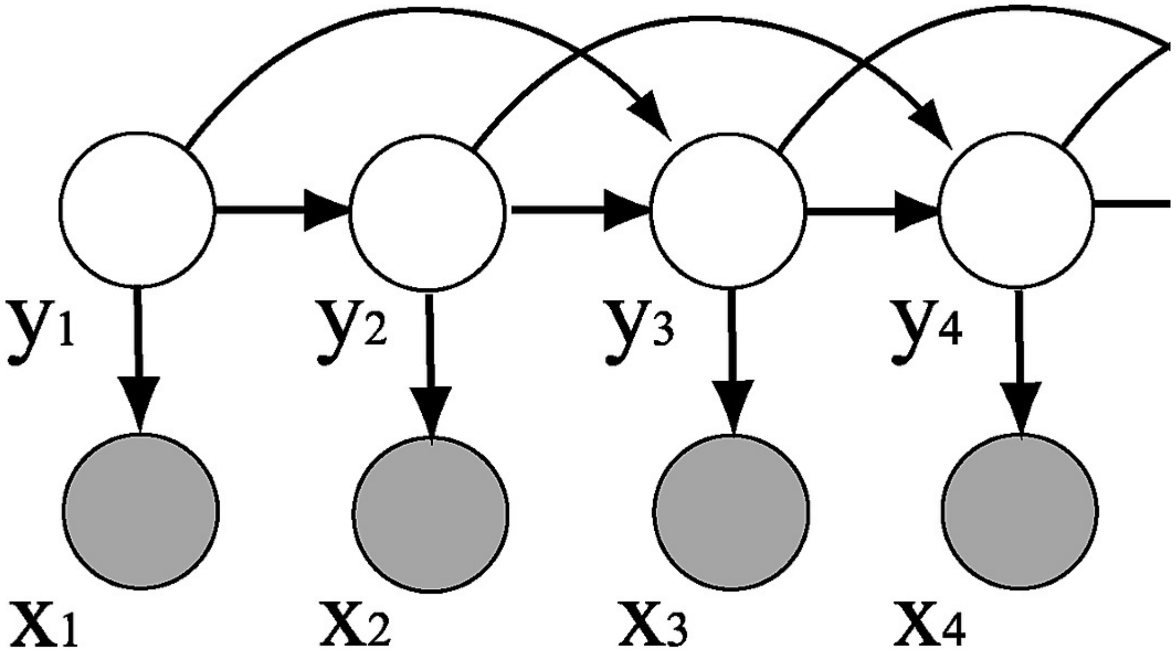


Image adapted from: Sutton and McCallum (An Introduction to Conditional Random Fields)

Higher order Markov chain models



Second order model

Image source: Katahira et al. (Complex sequencing rules of birdsong can be explained by simple Hidden Markov Processes)

Ergodic Markov chains

- Might have additional constraints on y_i, y_{i+1}
- e.g., part-of-speech tagging on input document
  - label each word with tag, e.g., noun, adjective, or punctuation mark
  - unlikely that y_i and y_{i+1} are both punctuation marks
- Markov chain ergodic if any state can be *reached* from any other state in finite (at most Δ) steps
  - Δ is the diameter
  - we define effective diameter $\tilde{\Delta} = \Delta + n - 1$
    - here, n is the order of Markov chain
    - $\tilde{\Delta} = 1$ for the fully connected ($\Delta = 1$) first order ($n = 1$) setting

How do we decode Markov chains?

- May view decoding as maximizing a sum of scores or rewards
  - e.g. in first order hidden Markov Model, reward pertaining to (x_i, y_i) is simply
    $\log P(y_i | y_{i-1}) + \log P(x_i | y_i)$
  - find a sequence of states y that maximizes the sum
    - break ties arbitrarily
- Exact solution by dynamic programming
  - method commonly known as the Viterbi algorithm

Why online Markov decoding?

- Viterbi has high *latency*
  - Processes entire input x before producing any state labels
  - not suitable for several scenarios (see Narasimhan et al.)
    - network intrusion detection
    - critical patient health monitoring
    - low resource devices that cannot store long input x
- We would like to have latency at most *L*
  - i.e. decode any y_i using only $x_i, x_{i+1}, \dots x_{i+L}$
  - also ensure quality of decoding does not suffer much

M. Narasimhan, P. Viola, and M. Shilman (Online decoding of Markov models under latency constraints)

How do we evaluate quality of decoding?

- Assume each reward is non-negative
  - can always add same positive quantity to each possible reward
    - does not change the maximizing sequence
    - therefore, without loss of generality
- Competitive ratio (C.R.)
  - OPT = total reward fetched by optimal algorithm (Viterbi)
  - ON = total reward by online algorithm
  - C.R. = OPT/ON is our measure
    - since each reward > 0, C.R. is at least 1
    - we would like to minimize C.R.
    - plug in expected value of ON instead for randomized online algorithms

Our results on C.R.

	Lower Bound	Upper Bound (Our Algorithms)
Deterministic ($\Delta = 1, n = 1$)	$1 + \frac{1}{L} + \frac{1}{L^2+1}$	$\min\left\{\left(1+\frac{1}{L}\right)\sqrt[L]{L+1}, 1+\frac{4}{L-7}\right\}$
Randomized ($\Delta = 1, n = 1, \epsilon > 0$)	$1 + \frac{(1-\epsilon)}{L+\epsilon}$	$1 + \frac{1}{L}$
Deterministic	$1 + \frac{\tilde{\Delta}}{L}\left(1 + \frac{\tilde{\Delta}+L-1}{(L-\tilde{\Delta}-1)^2 + 4\tilde{\Delta}L - 3\tilde{\Delta}}\right)$	$1 + \min\left\{\Theta\left(\frac{\log L}{L-\tilde{\Delta}+1}\right), \Theta\left(\frac{1}{L-8\tilde{\Delta}+1}\right)\right\}$
Randomized ($\epsilon > 0$)	$1 + \frac{(2^{\Delta-1}\lceil 1/\epsilon \rceil - 1)\, n}{2^{\Delta-1}\lceil 1/\epsilon \rceil L + n}$	$1 + \Theta\left(\frac{1}{L-\tilde{\Delta}+1}\right)$

Some Intuition: First Order Setting

If fully connected: (recovers the single server setting in Jayram et al.)

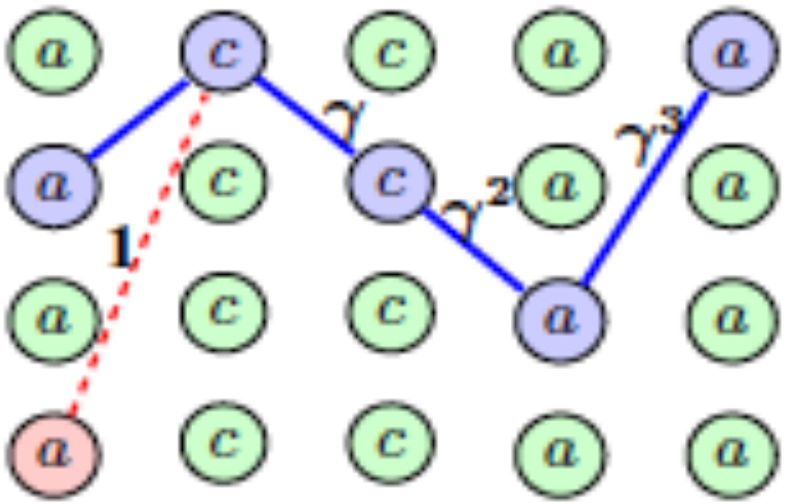
jump to a state on Viterbi path (blue nodes) in one step, and stay for next *L* steps

If not fully connected:

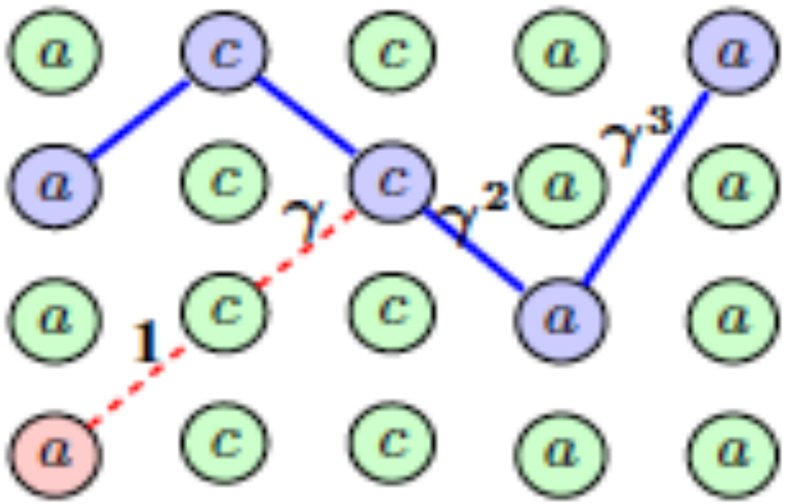
may have to waste additional steps before tracing Viterbi path

γ is an "explore-exploit" parameter (max value 1) that can be optimized based on *L*

Fully connected (diameter = 1)



diameter = 2



Jayram et al. (Online server allocation in a server farm via benefit task systems)

Key ideas: Algorithms and Analyses

- Understand the role of diameter and order for fixed latency
  - greater the diameter, worse the performance of online algorithm
  - likewise for the order
- Toolkit
  - adaptive optimization perspective for algorithm design
    - approximate Viterbi by a sequence of smaller problems, each over latency L
    - formulate optimization objectives that ensure each smaller problem is “good”
      - good if Viterbi only marginally better than the online algorithm on the smaller problem
  - prismatic polytope constructions for lower bounds
    - conjure scenarios such that the effects of diameter and order add up