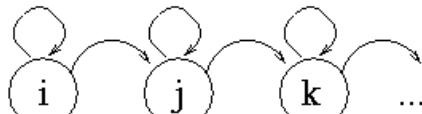


CS440/ECE448 Lecture 6: Hidden Markov Models

Mark Hasegawa-Johnson

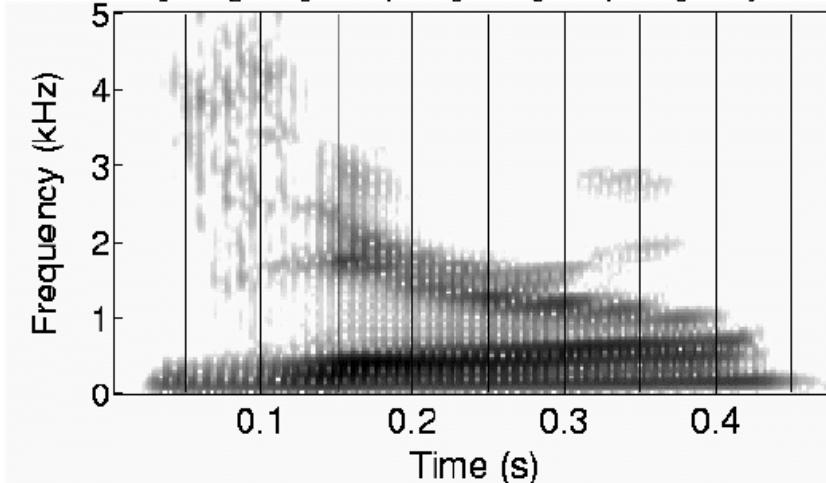
CC0 Public Domain

Re-use, Re-mix, Re-distribute at will

Hidden Markov Model = [ ...]

State Sequence Q = [i i i j j k k k ...]

Observations O = [o₁ o₂ o₃ o₄ o₅ o₆ o₇ o₈ o₉ ...]



Outline

- HMM: Probabilistic reasoning over time
- Viterbi algorithm

Review: Bayesian Classifier

- Class label $Y = y$, drawn from some set of labels
- Observation $X = x$, drawn from some set of features
- Bayesian classifier: choose the class label, y , that minimizes your probability of making a mistake:

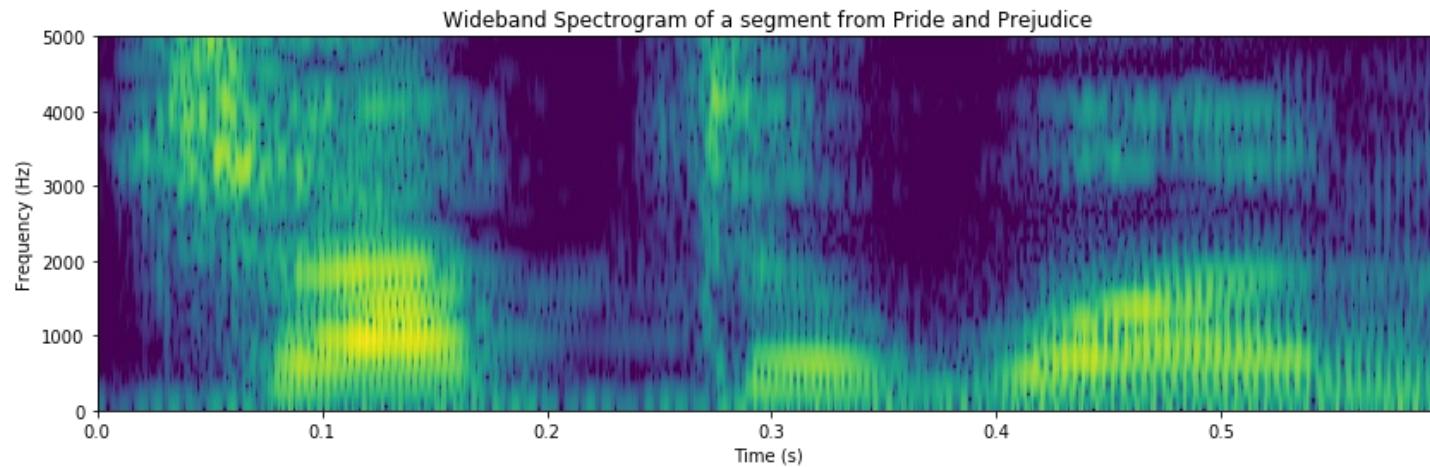
$$f(x) = \operatorname{argmax}_y P(Y = y | X = x)$$

Hidden Markov model: X and Y are sequences

- Class label sequence $Y = [Y_1, \dots, Y_T]$
- Observation sequence $X = [X_1, \dots, X_T]$
- Bayesian classifier: choose the class label sequence, $[y_1, \dots, y_T]$, that minimizes your probability of making a mistake:

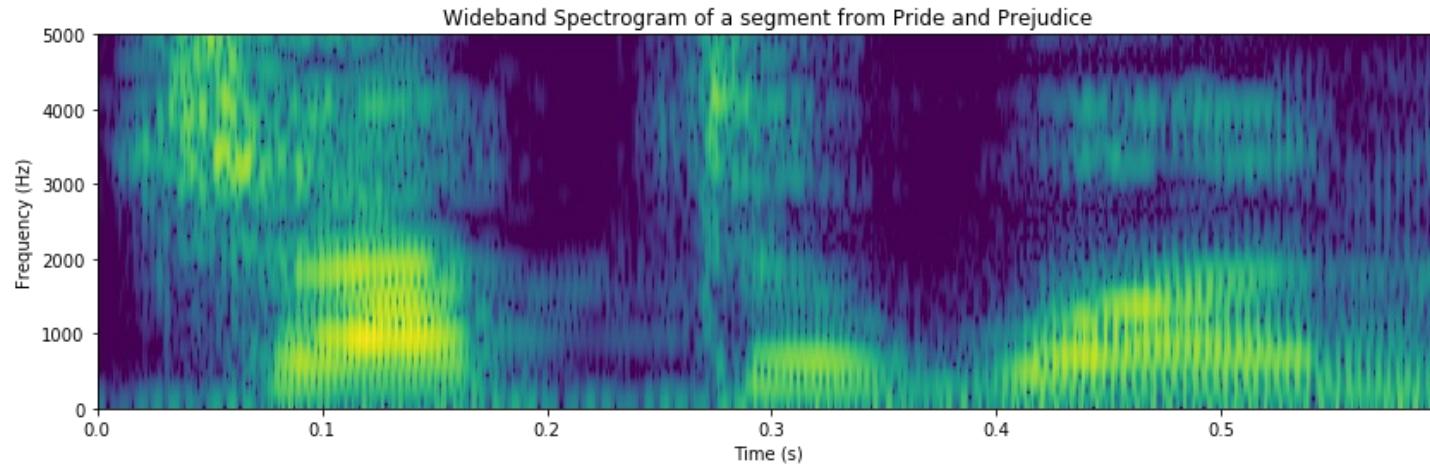
$$f(x) = \operatorname{argmax}_{y_1, \dots, y_T} P(Y = [y_1, \dots, y_T] | X = [x_1, \dots, x_T])$$

Example: Speech Recognition



- Here's a spectrogram of the utterance "chapter one."
- Each column is the Fourier transform of 0.02s of audio, spaced 0.01s apart. Let's call the spectral vector X_t , where t is time in centiseconds
- The speech sounds follow a sequence: silence for a while, then /sh/ for a while, then /ae/ for a while, then.... Let's denote the speech sound at time t as Y_t

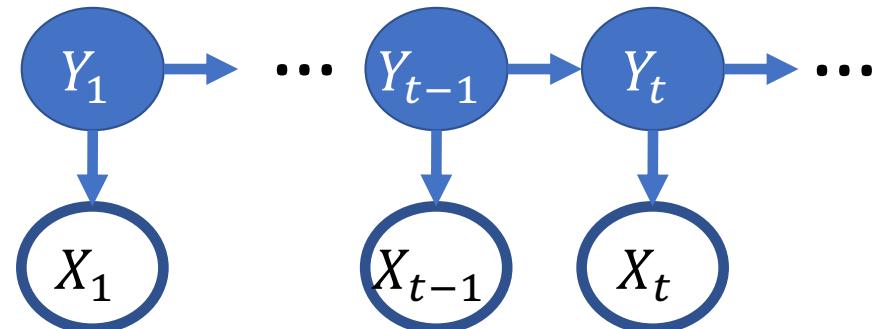
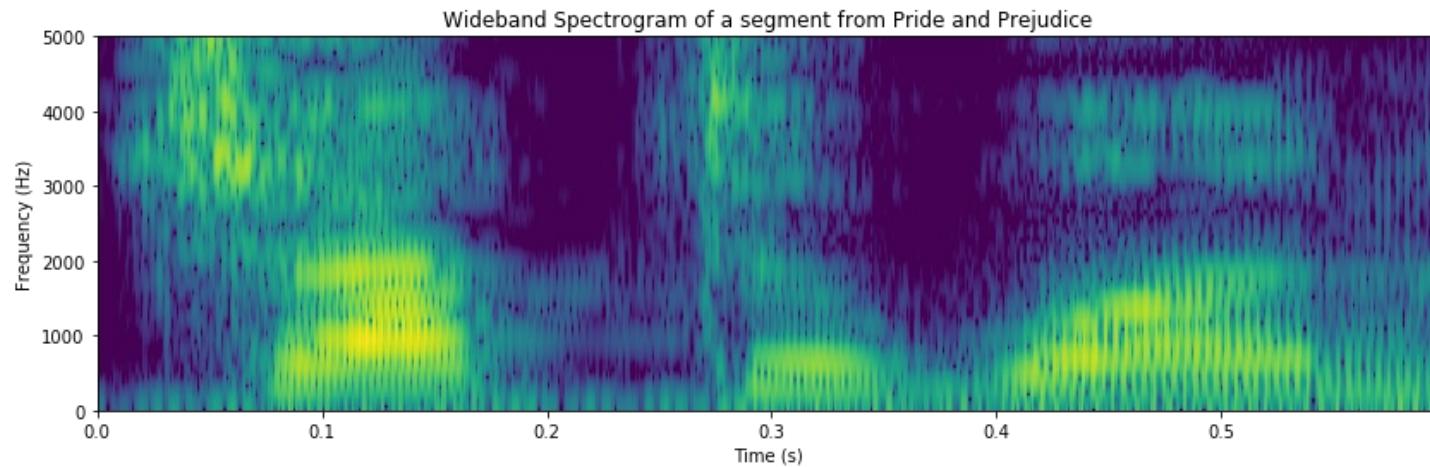
Hidden Markov Model



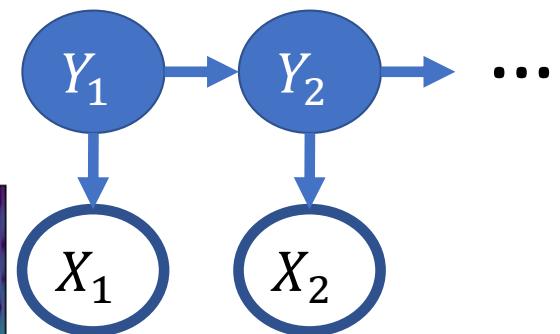
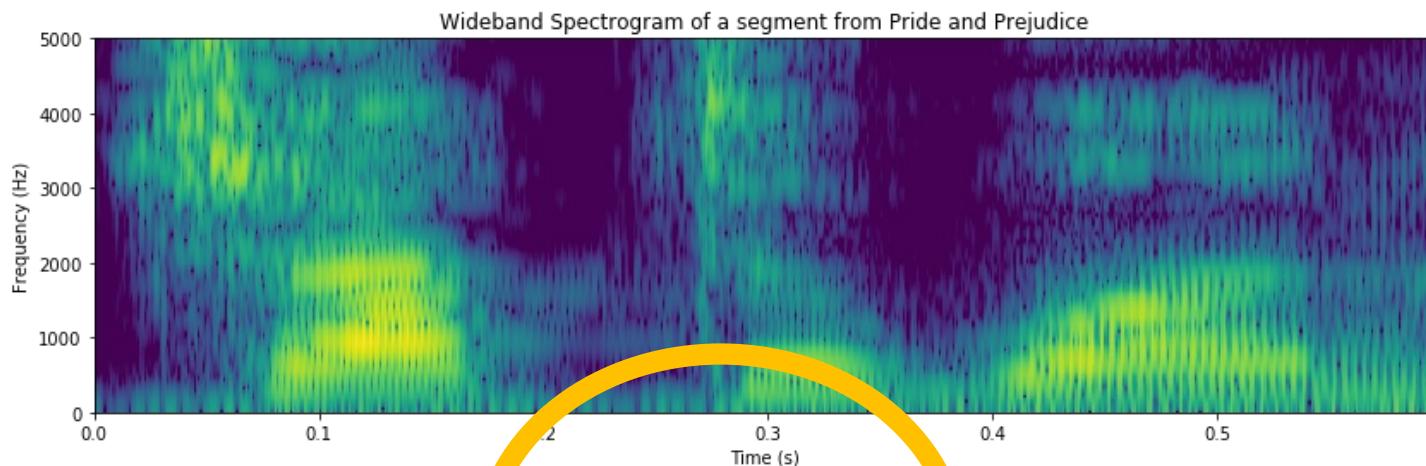
“Hidden Markov” Model:

- **Hidden:** You don't know the label Y_t , instead, you only know the observation X_t , and the probabilities $P(X_t|Y_t)$
- **Markov:** Y_t depends only on Y_{t-1} , and you know $P(Y_t|Y_{t-1})$

Hidden Markov Model is a Bayes Network



Hidden Markov Model



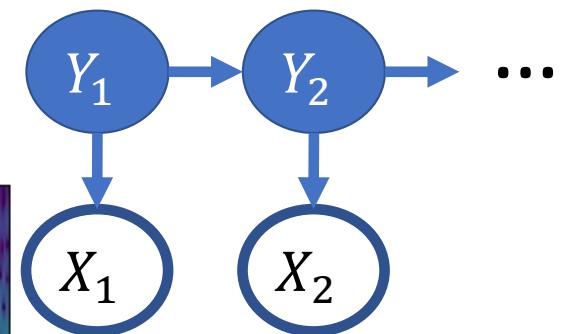
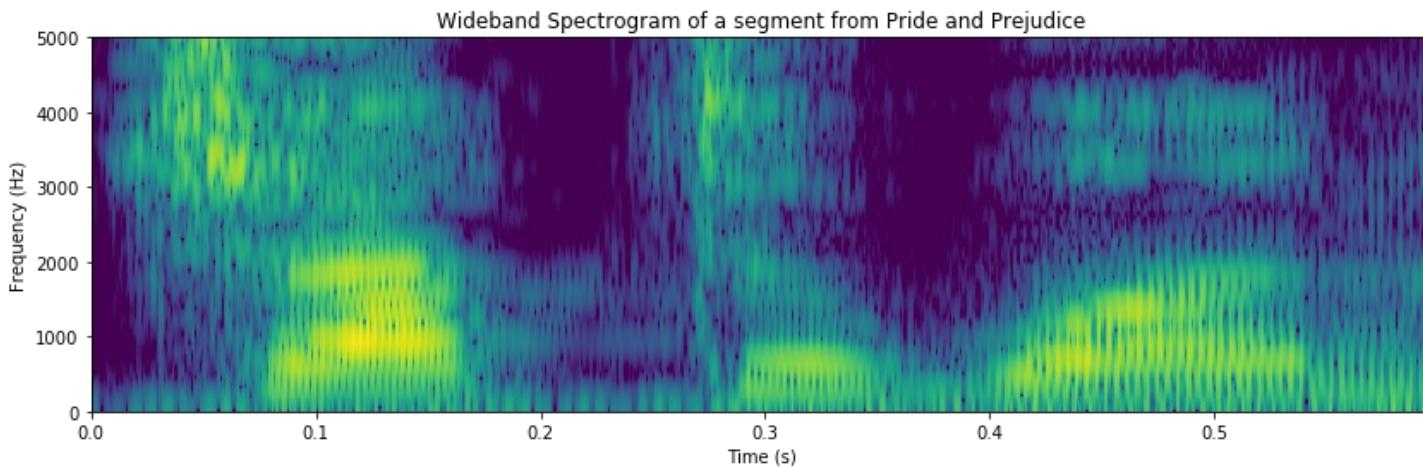
$$P(x_1, \dots, x_d) = \sum_{y_1} \sum_{y_2} \dots \sum_{y_d} P(y_1) P(x_1|y_1) P(y_2|y_1) P(x_2|y_2) P(y_3|y_2) \dots$$

$\mathcal{O}(|Y|^d)$ terms in this summation. Does this mean time-complexity is $\mathcal{O}(|Y|^d)$?

The problem HMMs solve: Exponential complexity

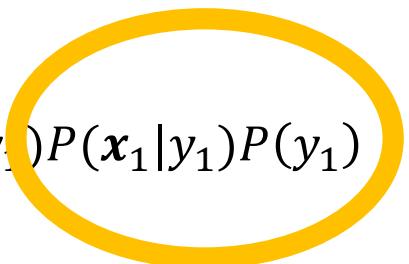
- Suppose there are $|\mathcal{Y}|$ different speech sounds in English ($|\mathcal{Y}| \approx 50, d \approx 100$)
- The length of the utterance is d centiseconds ($d \approx 100$)
- Without the HMM assumptions, to compute $f(\mathbf{x}) = \text{argmax} P(y_1, \dots, y_d | \mathbf{x}_1, \dots, \mathbf{x}_d)$ requires a time complexity of $\mathcal{O}\{|\mathcal{Y}|^d\} \approx 50^{100}$

Hidden Markov Model



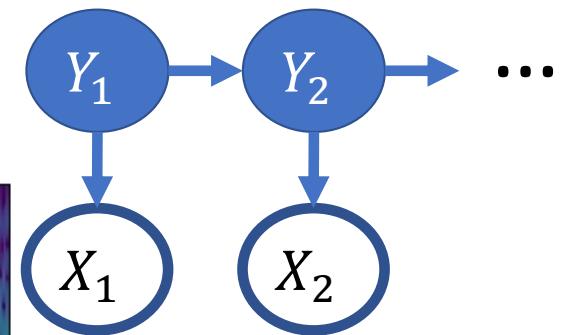
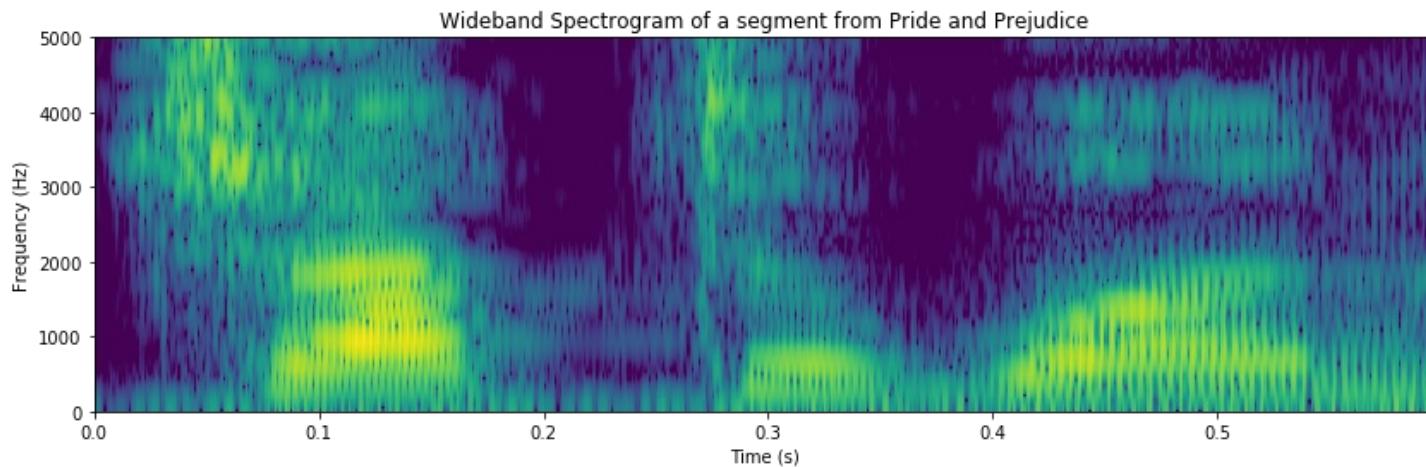
$$P(x_1, \dots, x_d) = \sum_{y_1} \sum_{y_2} \dots \sum_{y_d} P(y_1) P(x_1|y_1) P(y_2|y_1) P(x_2|y_2) P(y_3|y_2) \dots$$

$$= \sum_{y_d} \dots \sum_{y_2} P(y_3|y_2) P(x_2|y_2) \sum_{y_1} P(y_2|y_1) P(x_1|y_1) P(y_1)$$



$\mathcal{O}\{|\mathcal{Y}|\}$

Hidden Markov Model

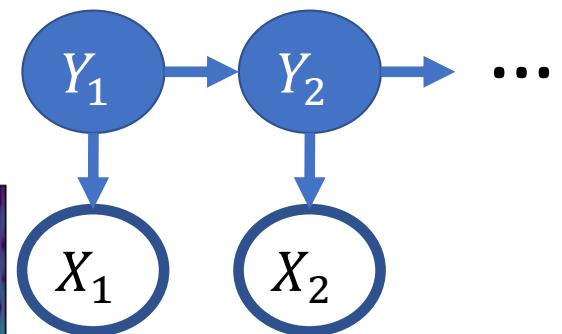
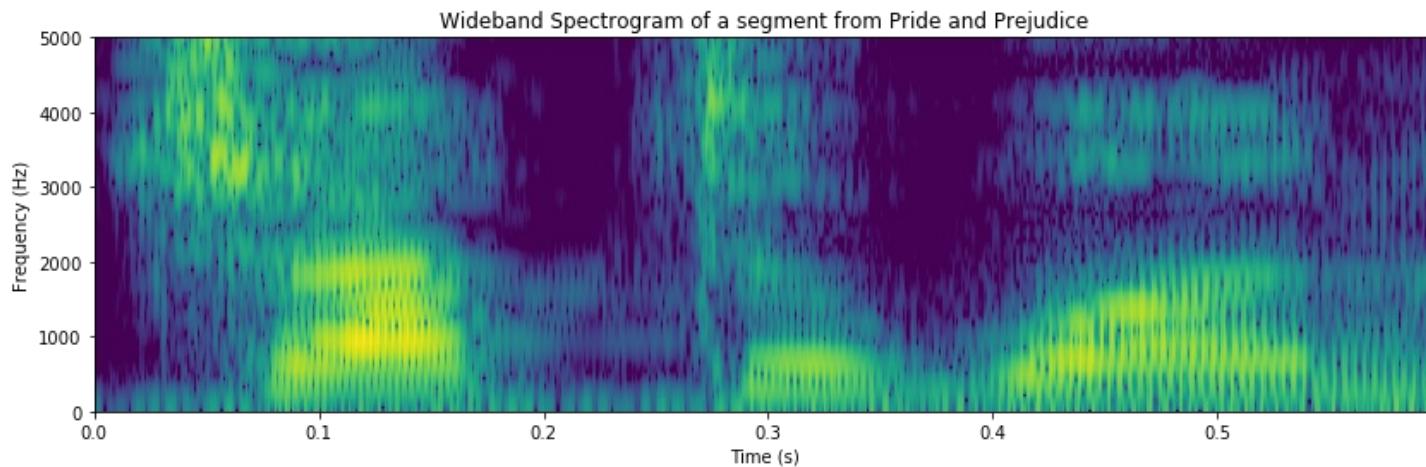


$$P(x_1, \dots, x_d) = \sum_{y_1} \sum_{y_2} \dots \sum_{y_d} P(y_1) P(x_1|y_1) P(y_2|y_1) P(x_2|y_2) P(y_3|y_2) \dots$$

$$= \sum_{y_d} \dots \sum_{y_2} P(y_3|y_2) P(x_2|y_2) \sum_{y_1} P(y_2|y_1) P(x_1|y_1) P(y_1)$$

$\mathcal{O}\{|\mathcal{Y}|^2\}$

Hidden Markov Model

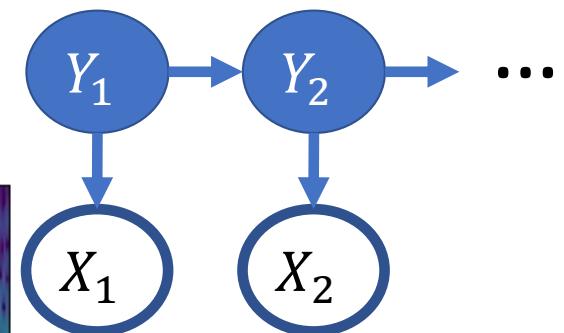
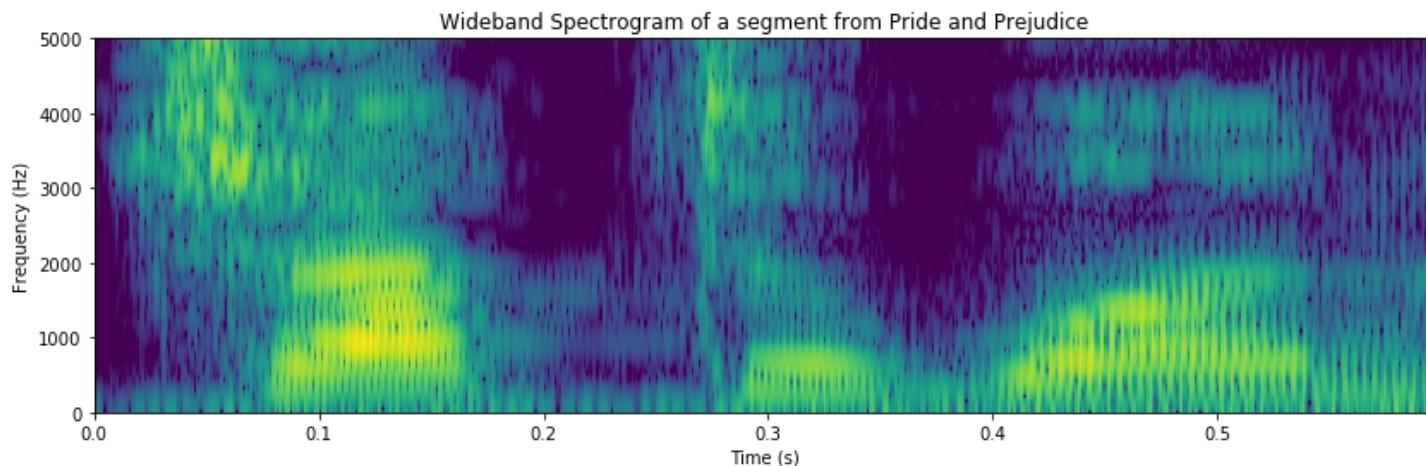


$$P(\mathbf{x}_1, \dots, \mathbf{x}_d) = \sum_{y_1} \sum_{y_2} \dots \sum_{y_d} P(y_1) P(\mathbf{x}_1|y_1) P(y_2|y_1) P(\mathbf{x}_2|y_2) P(y_3|y_2) \dots$$

$$= \sum_{y_d} \dots \sum_{y_2} P(y_3|y_2) P(\mathbf{x}_2|y_2) \sum_{y_1} P(y_2|y_1) P(\mathbf{x}_1|y_1) P(y_1)$$

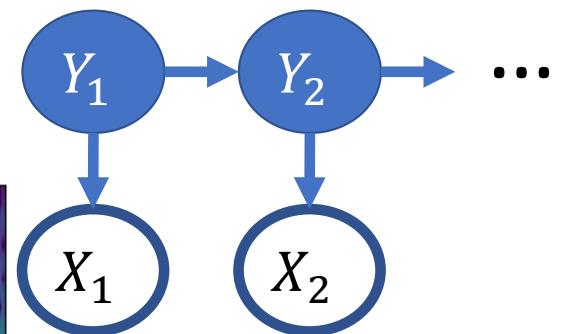
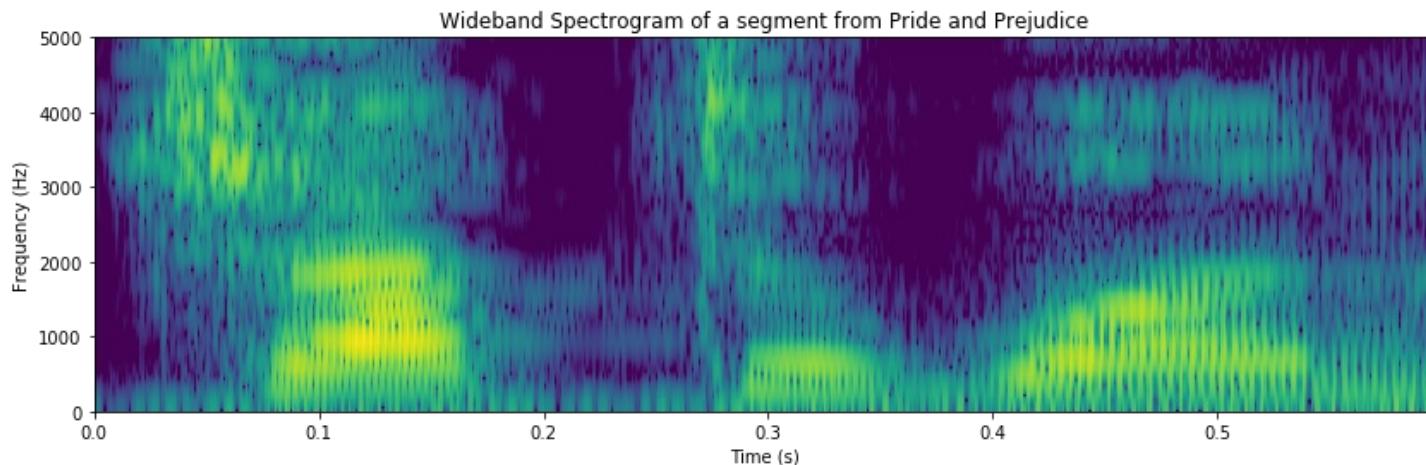
$\mathcal{O}\{|\mathcal{Y}|\}$

Hidden Markov Model



$$\begin{aligned} P(\mathbf{x}_1, \dots, \mathbf{x}_d) &= \sum_{y_1} \sum_{y_2} \dots \sum_{y_d} P(y_1) P(\mathbf{x}_1|y_1) P(y_2|y_1) P(\mathbf{x}_2|y_2) P(y_3|y_2) \dots \\ &= \sum_{y_d} \dots \sum_{y_2} P(y_3|y_2) P(\mathbf{x}_2|y_2) \sum_{y_1} P(y_2|y_1) P(\mathbf{x}_1|y_1) P(y_1) \end{aligned} \quad \mathcal{O}\{|\mathbf{y}|^2\}$$

Hidden Markov Model



$$\begin{aligned} P(\mathbf{x}_1, \dots, \mathbf{x}_d) &= \sum_{y_1} \sum_{y_2} \dots \sum_{y_d} P(y_1) P(\mathbf{x}_1|y_1) P(y_2|y_1) P(\mathbf{x}_2|y_2) P(y_3|y_2) \dots \\ &= \sum_{y_d} \dots \sum_{y_2} P(y_3|y_2) P(\mathbf{x}_2|y_2) \sum_{y_1} P(y_2|y_1) P(\mathbf{x}_1|y_1) P(y_1) \end{aligned}$$

$\mathcal{O}\{|\mathbf{y}|\}$

Key advantage of a hidden Markov model: Polynomial-time complexity

- A hidden Markov model makes the computation local, in the sense that each Y_t depends only on Y_{t-1}
- As a result, the computational complexity never gets larger than $\mathcal{O}\{|Y|^2\}$

...and it works much better than naïve Bayes.

Claude Shannon (1948) gave these examples:

- Text generated by unigram naïve Bayes:

Representing and speedily is an good apt or come can different natural
here he the a in came the to of to expert gray come to furnishes the
line message had be these...

- Text generated by an HMM with bigram transitions:

The head and in frontal attack on an English writer that the character of
this point is therefore another for the letters that the time of who ever
told the problem for an unexpected...

Applications

...as listed at https://en.wikipedia.org/wiki/Hidden_Markov_model

- Computational finance
- Single-molecule kinetic analysis
- Neuroscience
- Cryptanalysis
- Speech recognition
- Speech synthesis
- Part-of-speech tagging
- Document scanning
- Machine translation
- Dielectric breakdown
- Gene prediction
- Handwriting recognition
- Alignment of bio-sequences
- Time series analysis
- Activity recognition
- Protein folding
- Sequence classification
- Metamorphic virus detection
- DNA/protein motif discovery
- DNA hybridization kinetics
- Chromatin state discovery
- Transportation forecasting
- Solar irradiance variability

Start with a multiple sequence alignment

↓

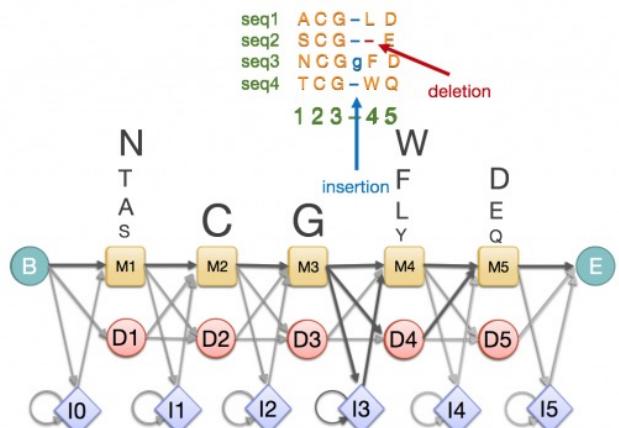
Insertions / deletions can be modelled

↓

Occupancy and amino acid frequency at each position in the alignment are encoded

↓

Profile created



CC-SA 4.0,
https://commons.wikimedia.org/wiki/File:A_profile_HMM_modelling_a_multiple_sequence_alignment.png

Outline

- HMM: Probabilistic reasoning over time
- Viterbi algorithm

Viterbi Algorithm

The Viterbi algorithm is a computationally efficient algorithm for computing the maximum *a posteriori* (MAP) state sequence,

$$f(\mathbf{x}) = \operatorname{argmax}_{y_1, \dots, y_d} P(y_1, \dots, y_d | \mathbf{x}_1, \dots, \mathbf{x}_d)$$

Bayes Network view, Finite State Machine view

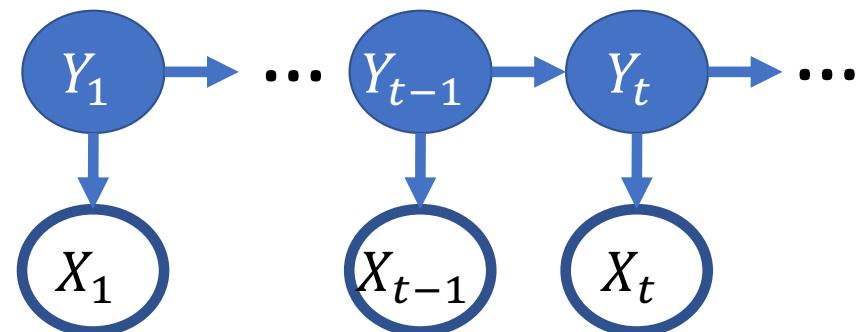
Bayes Network = view across time

- Node = one of the variables ($Y_1, X_1, Y_2, X_2, \dots$)
- Edge = dependence

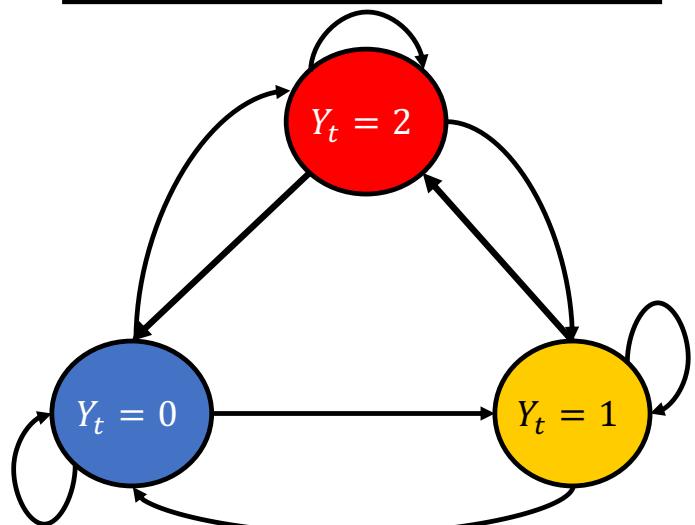
FSM = close-up of one particular time

- “State” = one of the values that Y_t can take
- “Edge” = a possible transition

HMM as a Bayes Network



HMM as an FSM



The parameters that define an HMM

- Initial State Probability:

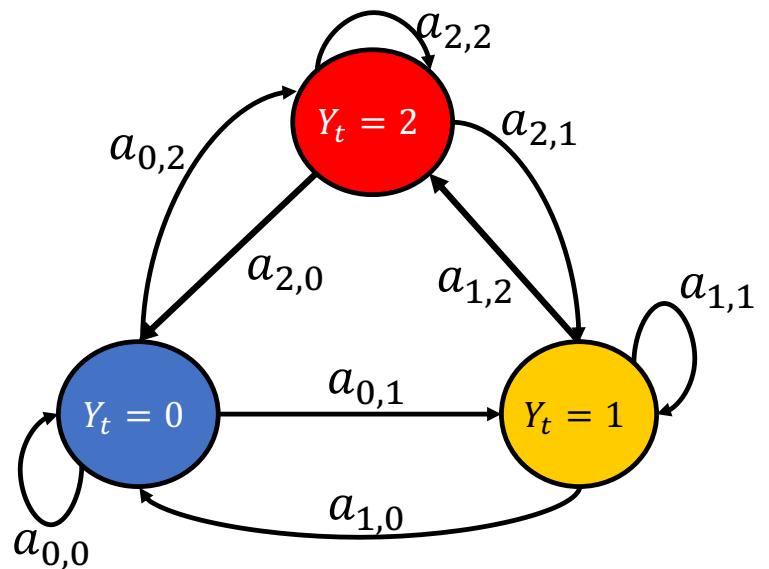
$$\pi_i = P(Y_1 = i)$$

- Transition Probabilities:

$$a_{i,j} = P(Y_t = j | Y_{t-1} = i)$$

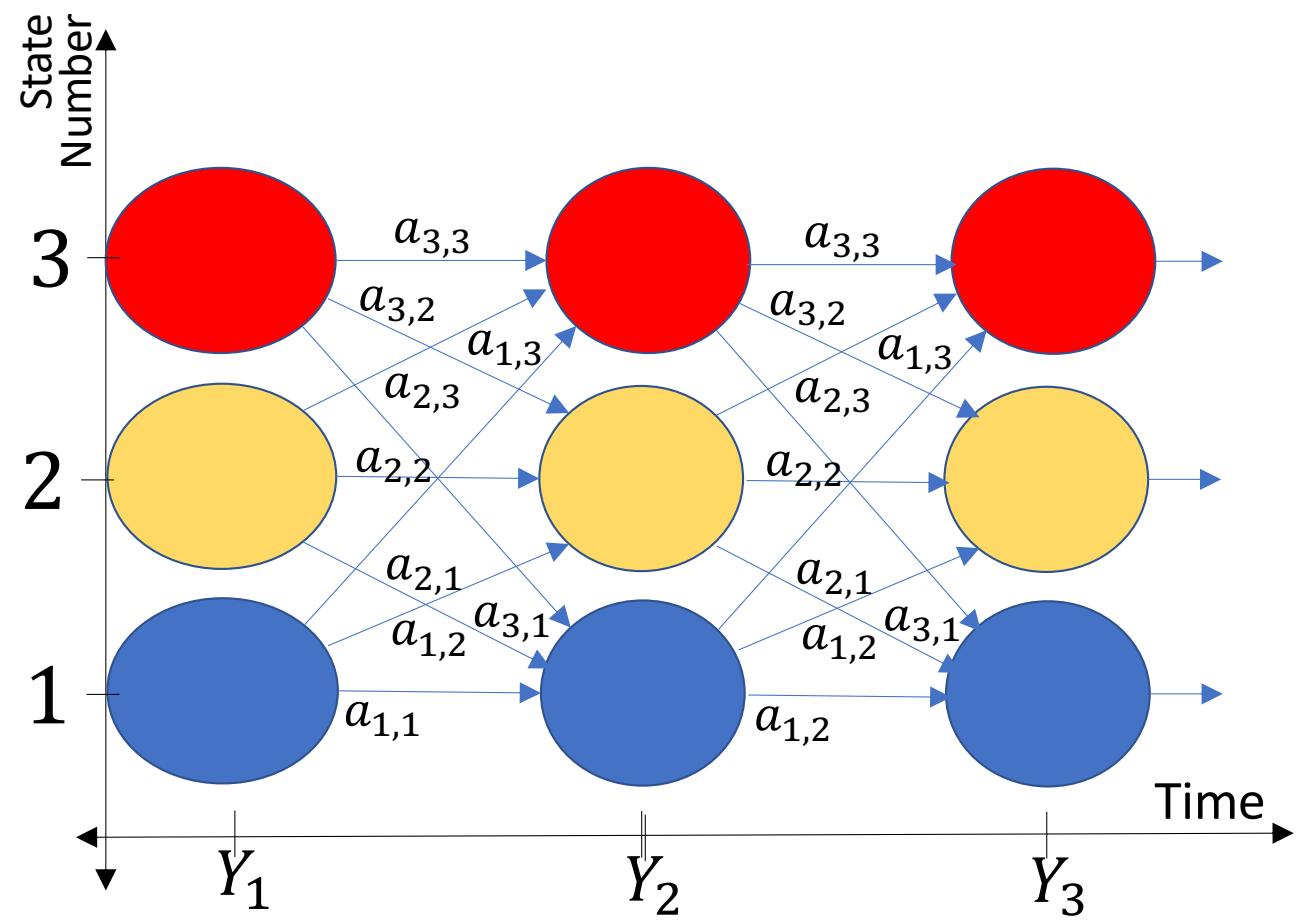
- Observation Probabilities:

$$b_j(x_t) = P(X_t = x_t | Y_t = j)$$



The Trellis =
BN view \times
FSM view

- Time is on the horizontal axis
- State number on the vertical axis
- Edges show state transitions: $a_{i,j}$

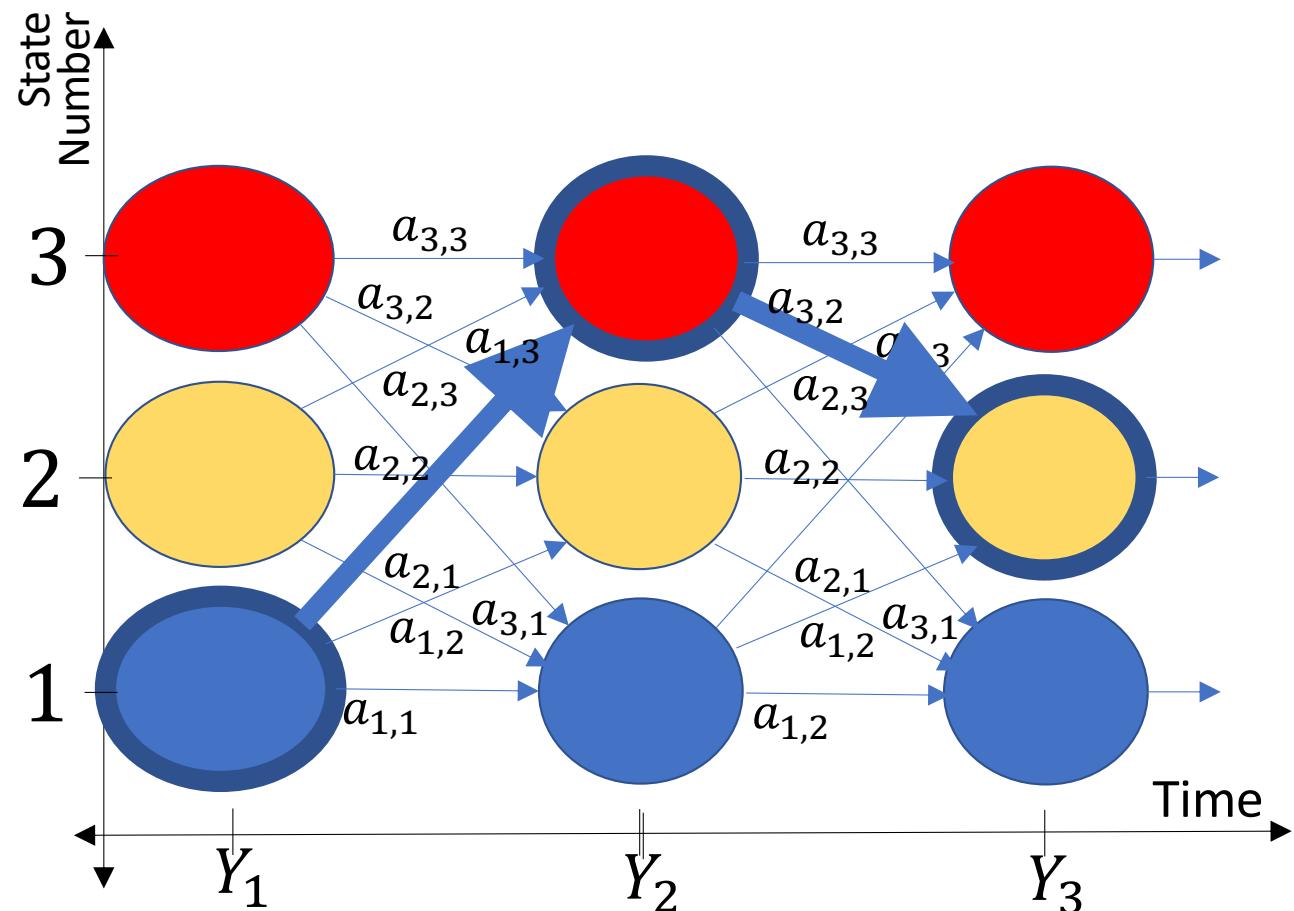


The Trellis

- A sequence of state variables is a path through the trellis.

For example:

$$P(Y_1 = 1, Y_2 = 3, Y_3 = 2, X_1 = x_1, X_2 = x_2, X_3 = x_3) = \pi_1 b_1(x_1) a_{1,3} b_3(x_2) a_{3,2} b_2(x_3)$$



Scores and backpointers

- **Score** = log probability of the best path until node j at time t

$$v_t(j) = \max_{y_1, \dots, y_{t-1}} \log P(Y_1 = y_1, \dots, Y_{t-1} = y_{t-1}, Y_t = j, X_0 = x_0, \dots, X_t = x_t)$$

- **Backpointer** = which node precedes node j on the best path?

$$\psi_t(j) = \operatorname{argmax}_{y_{t-1}} \max_{y_1, \dots, y_{t-2}} \log P(Y_1 = y_1, \dots, Y_{t-1} = y_{t-1}, Y_t = j, X_0 = x_0, \dots, X_t = x_t)$$

Forward tracing and backtracing

- **Forward tracing** = Work from left to right through the trellis, finding the score of every node

$$v_t(j) = \max_{y_1, \dots, y_{t-1}} \log P(Y_1 = y_1, \dots, Y_{t-1} = y_{t-1}, Y_t = j, X_0 = x_0, \dots, X_t = x_t)$$

- **Backtracing** = Work from right to left (backward), finding the best path that ends up in a known desirable endpoint

$$\psi_t(j) = \operatorname{argmax}_{y_{t-1}} \max_{y_1, \dots, y_{t-2}} \log P(Y_1 = y_1, \dots, Y_{t-1} = y_{t-1}, Y_t = j, X_0 = x_0, \dots, X_t = x_t)$$

Viterbi Algorithm

- Initialization: for all states i :

$$v_1(i) = \pi_i b_i(x_1)$$

- Forward-Tracing:

$$\text{SCORE: } v_t(j) = \max_i v_{t-1}(i) + \log a_{i,j} + \log b_j(x_t)$$

$$\text{BACKPOINTER: } \psi_t(j) = \operatorname{argmax}_i v_{t-1}(i) + \log a_{i,j} + \log b_j(x_t)$$

- Termination:

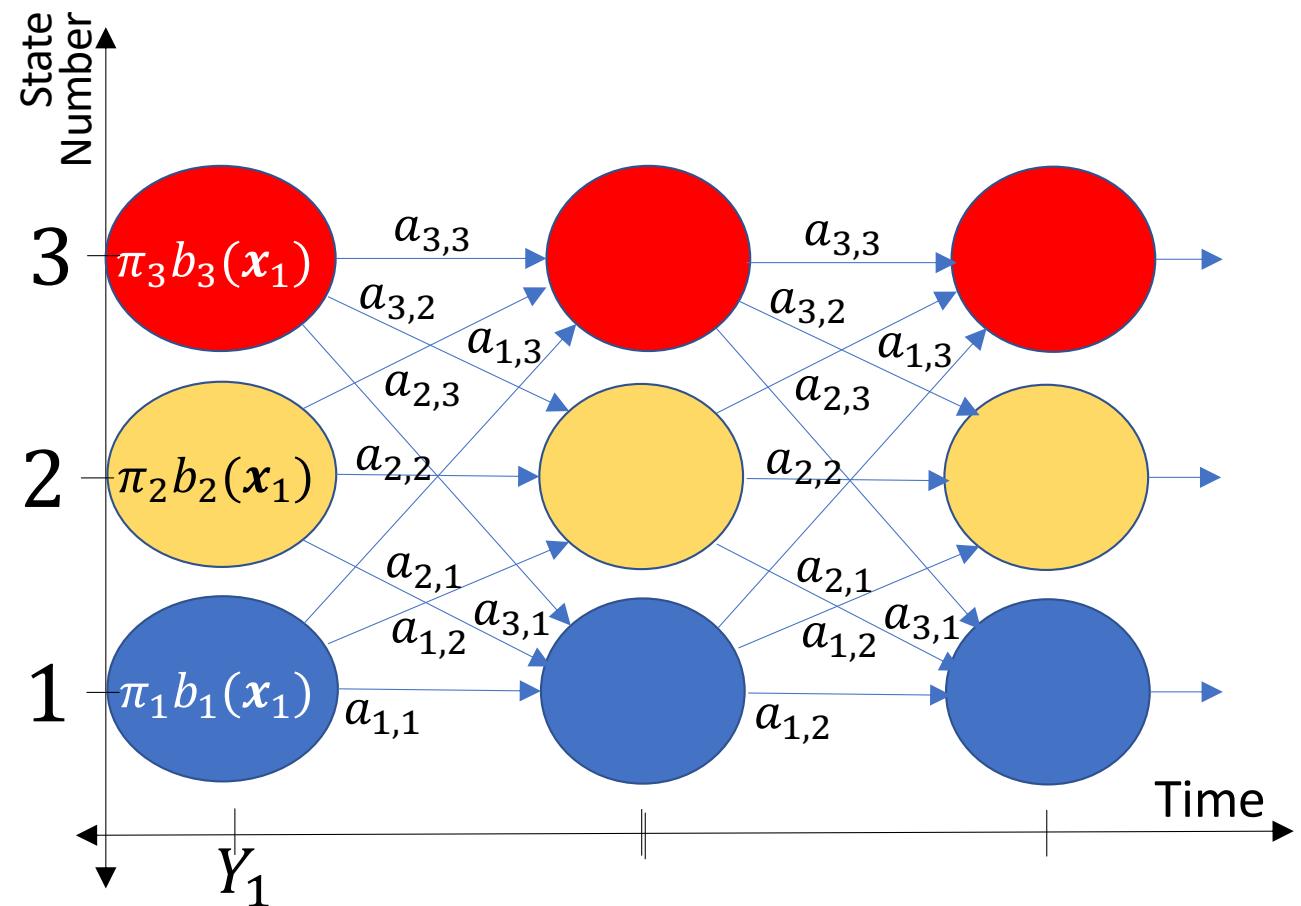
$$y_d = \operatorname{argmax}_i v_d(i)$$

- Back-Tracing:

$$y_t = \psi_{t+1}(y_{t+1}), \quad t = d-1, \dots, 1$$

Initialization

$$v_1(i) = \log \pi_i + \log b_i(\mathbf{x}_1)$$



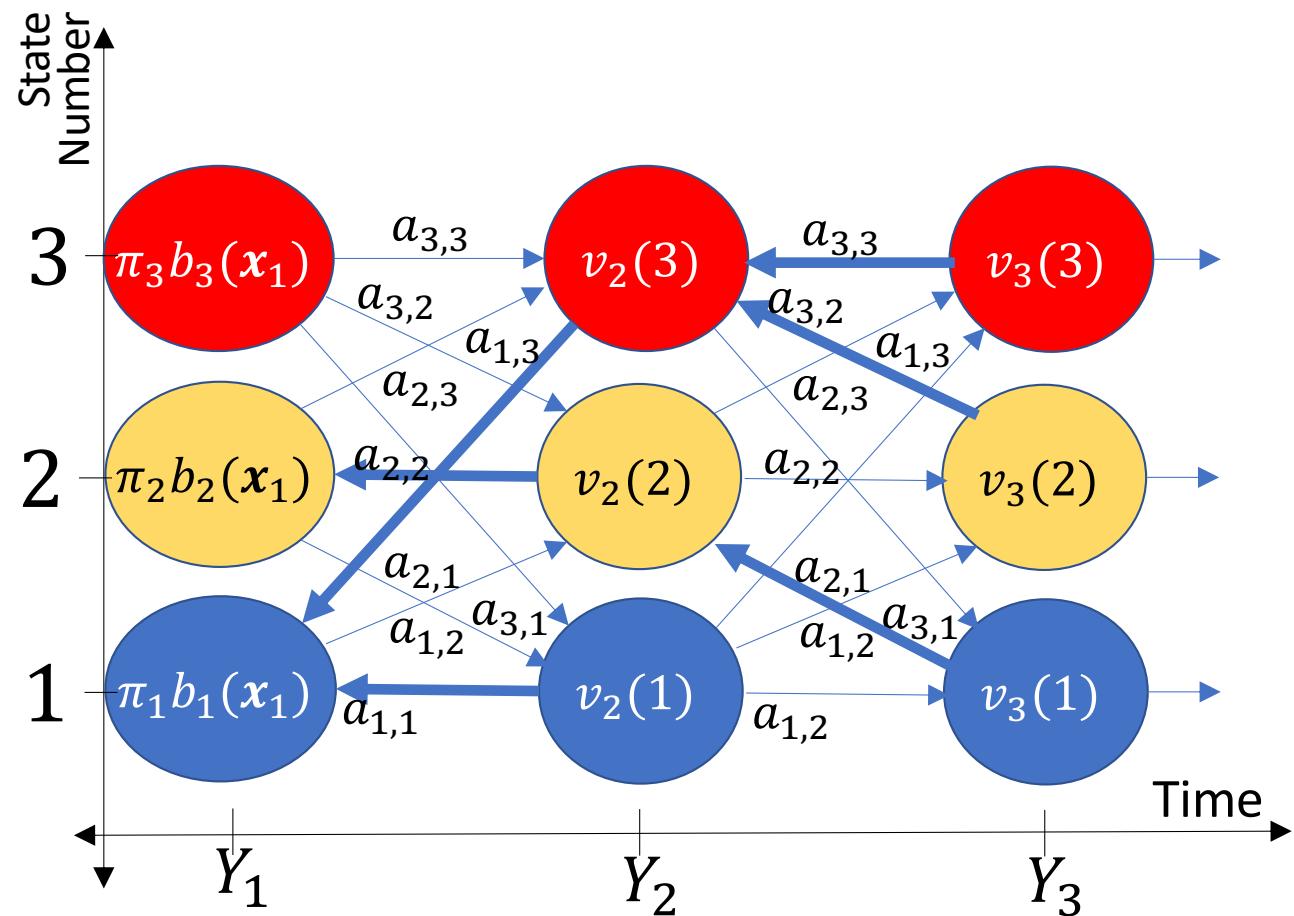
Forward-tracing

Each node now has a SCORE:

$$v_t(j) = \max_i v_{t-1}(i) + \log a_{i,j} + \log b_j(x_t)$$

... and each node has a BACKPOINTER:

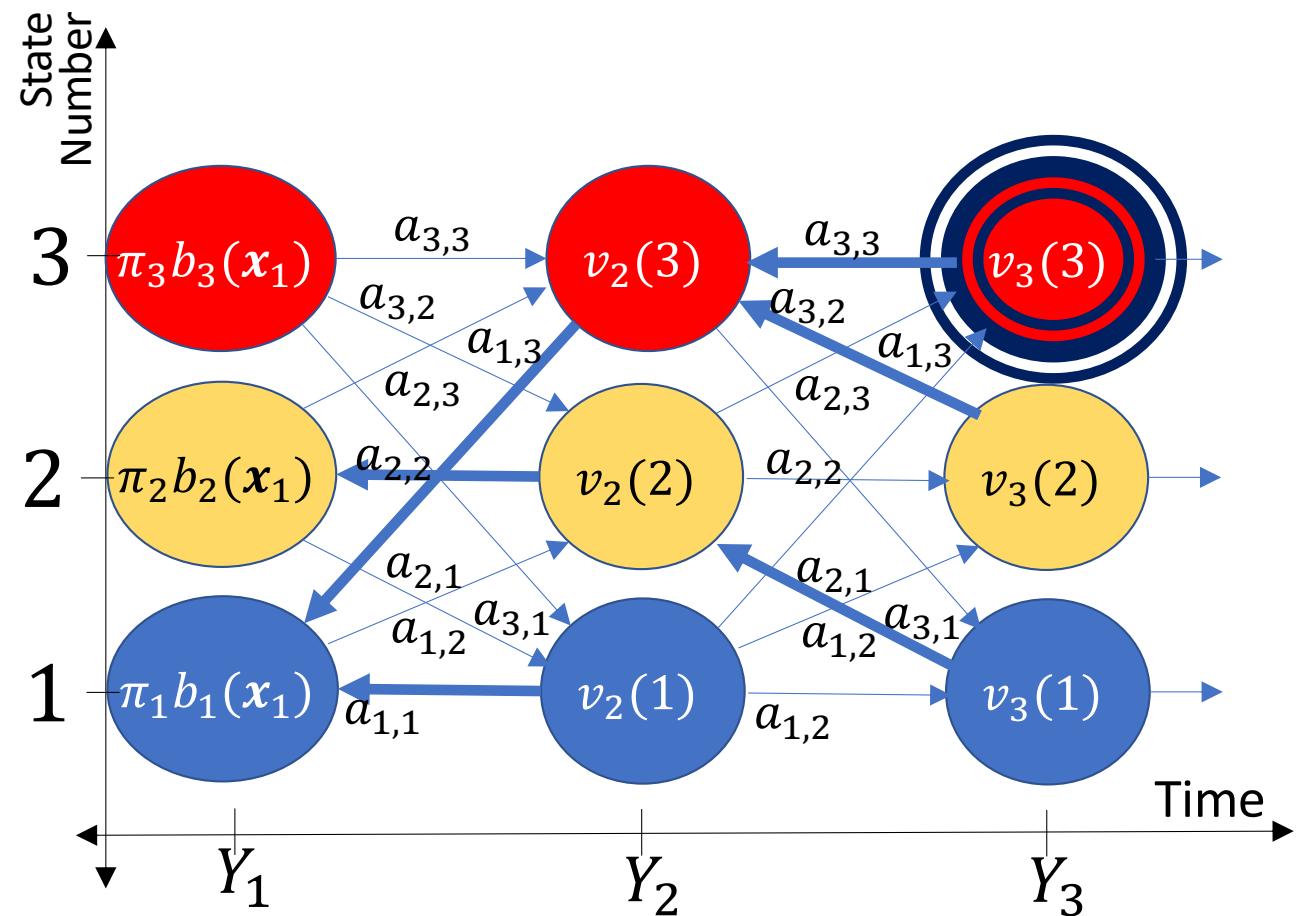
$$\psi_t(j) = \operatorname{argmax}_i v_{t-1}(i) + \log a_{i,j} + \log b_j(x_t)$$



Termination

The best path is the one that ends with the highest-value node:

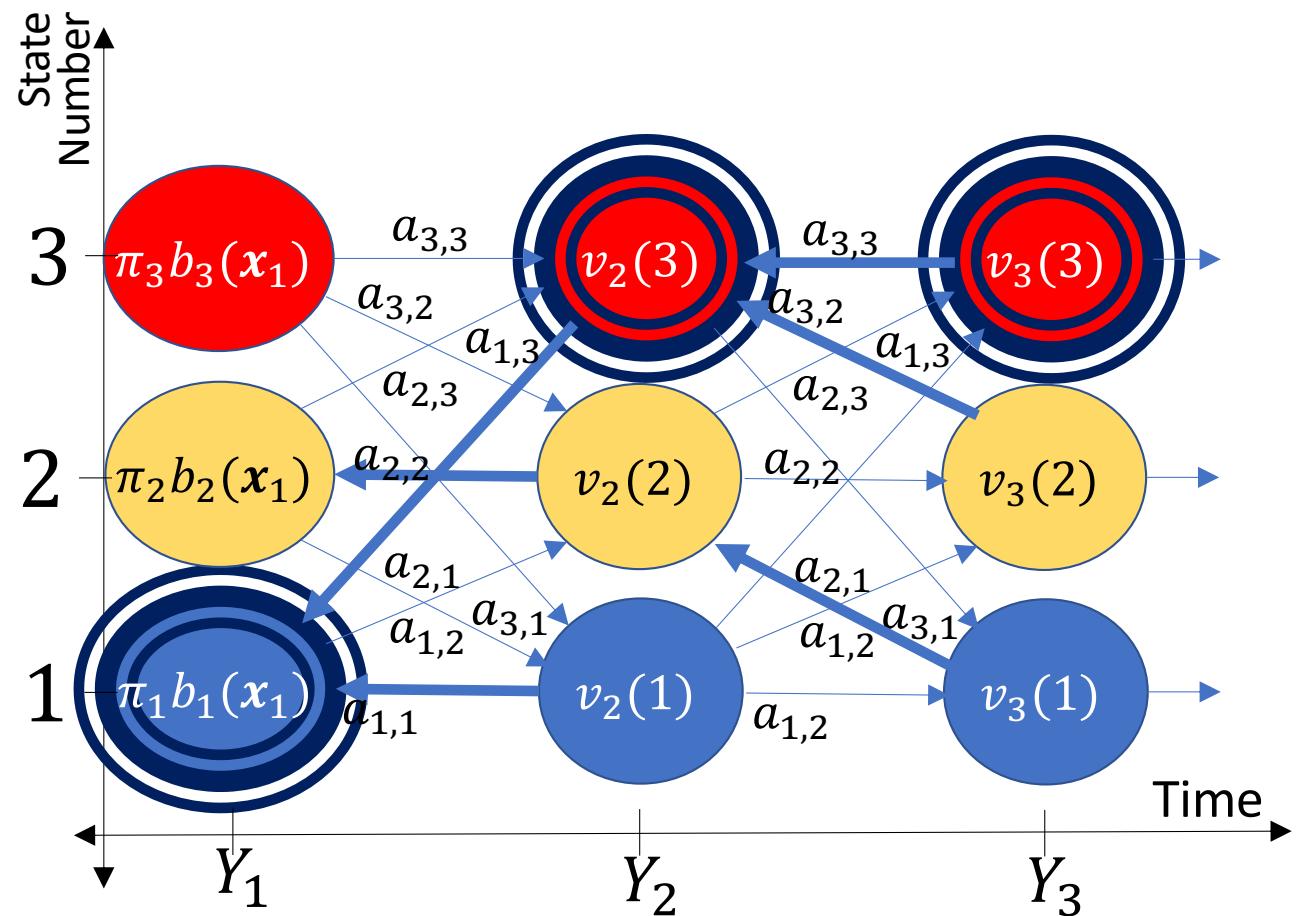
$$y_d = \operatorname{argmax}_i v_d(i)$$



Back-Tracing

The most likely state sequence is the one that ends with the highest-value node:

$$y_t = \psi_{t+1}(y_{t+1})$$



Viterbi Algorithm Computational Complexity

- Initialization: for $i \in \mathcal{Y}$:

$$v_1(i) = \pi_i b_i(x_1)$$

$\mathcal{O}(|\mathcal{Y}|)$

- Iteration: for $2 \leq t \leq d$, for $j \in \mathcal{Y}$:

SCORE: $v_t(j) = \max_i v_{t-1}(i) + \log a_{i,j} + \log b_j(x_t)$

$\mathcal{O}(d|\mathcal{Y}|^2)$

BACKPOINTER: $\psi_t(j) = \operatorname{argmax}_i v_{t-1}(i) + \log a_{i,j} + \log b_j(x_t)$

- Termination:

$$y_d = \operatorname{argmax}_{i \in \mathcal{Y}} v_d(i)$$

$\mathcal{O}(|\mathcal{Y}|)$

- Back-Trace:

$$y_t = \psi_{t+1}(y_{t+1})$$

$\mathcal{O}(d)$

Total: $\mathcal{O}(d|\mathcal{Y}|^2)$

Try the quiz!

- Go to prairielearn, try the quiz!

Outline

- Review: Bayesian classifier, Bayesian networks

$$f(x) = \operatorname{argmax}_y P(Y = y | X = x)$$

- HMM: Probabilistic reasoning over time

$$\begin{aligned}\pi_i &= P(Y_1 = i) \\ a_{i,j} &= P(Y_t = j | Y_{t-1} = i) \\ b_j(\mathbf{x}_t) &= P(X_t = \mathbf{x}_t | Y_t = j)\end{aligned}$$

- Viterbi algorithm

$$\text{SCORE: } v_t(j) = \max_i v_{t-1}(i) + \log a_{i,j} + \log b_j(\mathbf{x}_t)$$

$$\text{BACKPOINTER: } \psi_t(j) = \operatorname{argmax}_i v_{t-1}(i) + \log a_{i,j} + \log b_j(\mathbf{x}_t)$$