Hidden Markov Models and Monophonic Music Transcription

Brendt Gerics

University of Michigan, Flint

Advisor
Cameron Mcleman

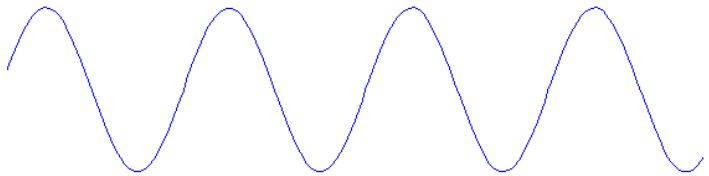
May 2, 2014

Musical Notation

Note/pitch

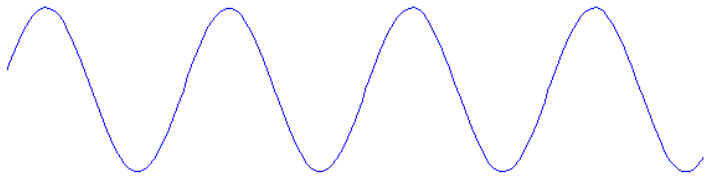
Musical Notation

Note/pitch



Musical Notation

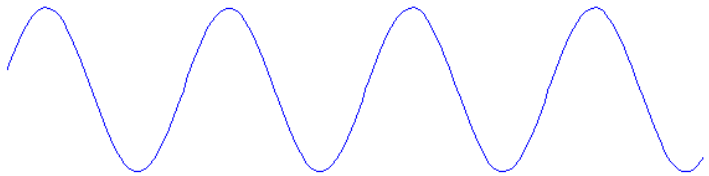
Note/pitch



- frequency $\mapsto$ note

Musical Notation

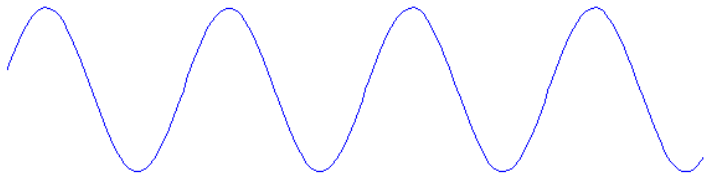
Note/pitch



- frequency $\mapsto$ note
- Best visualized by a keyboard

Musical Notation

Note/pitch



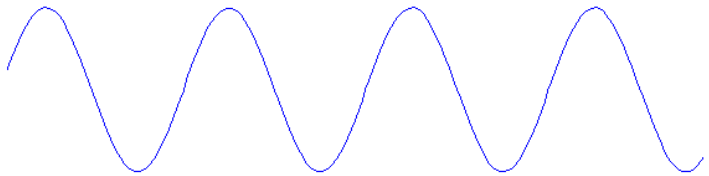
- frequency $\mapsto$ note
- Best visualized by a keyboard

Length of note

- quarternote $\quad = $ one beat

Musical Notation

Note/pitch



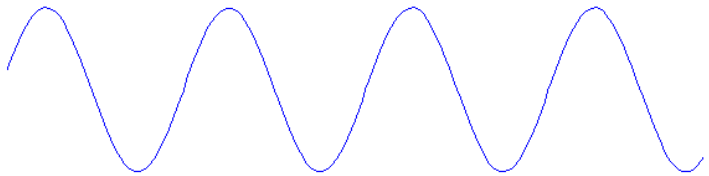
- frequency $\mapsto$ note
- Best visualized by a keyboard

Length of note

- quarternote ♩ = one beat
- half note 𝅗𝅥=two beats whole note 𝅝

Musical Notation

Note/pitch



- frequency $\mapsto$ note
- Best visualized by a keyboard

Length of note

- quarternote  = one beat
- half note  =two beats whole note
- eighth note 

Musical Notation

Note/pitch

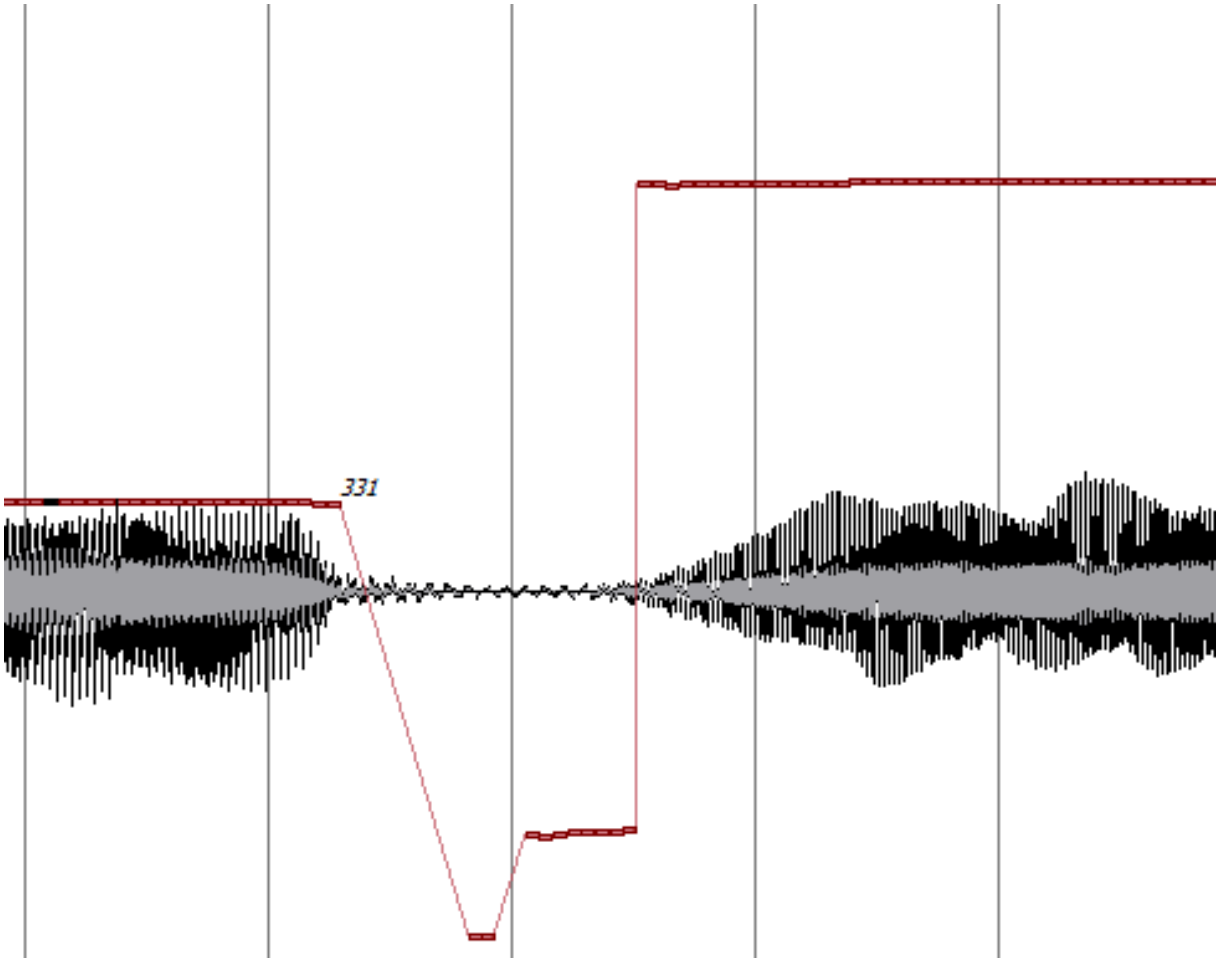
- frequency $\mapsto$ note
- Best visualized by a keyboard

Length of note

- quarternote ♩ = one beat
- half note 𝅗𝅥=two beats whole note 𝅝
- eighth note ♪
- Rests are identical

The problem arises

The YIN makes a guess at every interval, regardless of the input!



The question

How do we determine which of these notes to keep?

Why hidden?

Why hidden?

- Traditional Markov Processes are useful

Why hidden?

- Traditional Markov Processes are useful

Why hidden?

- Traditional Markov Processes are useful but...
- they assume we can determine states by direct observation.

Why hidden?

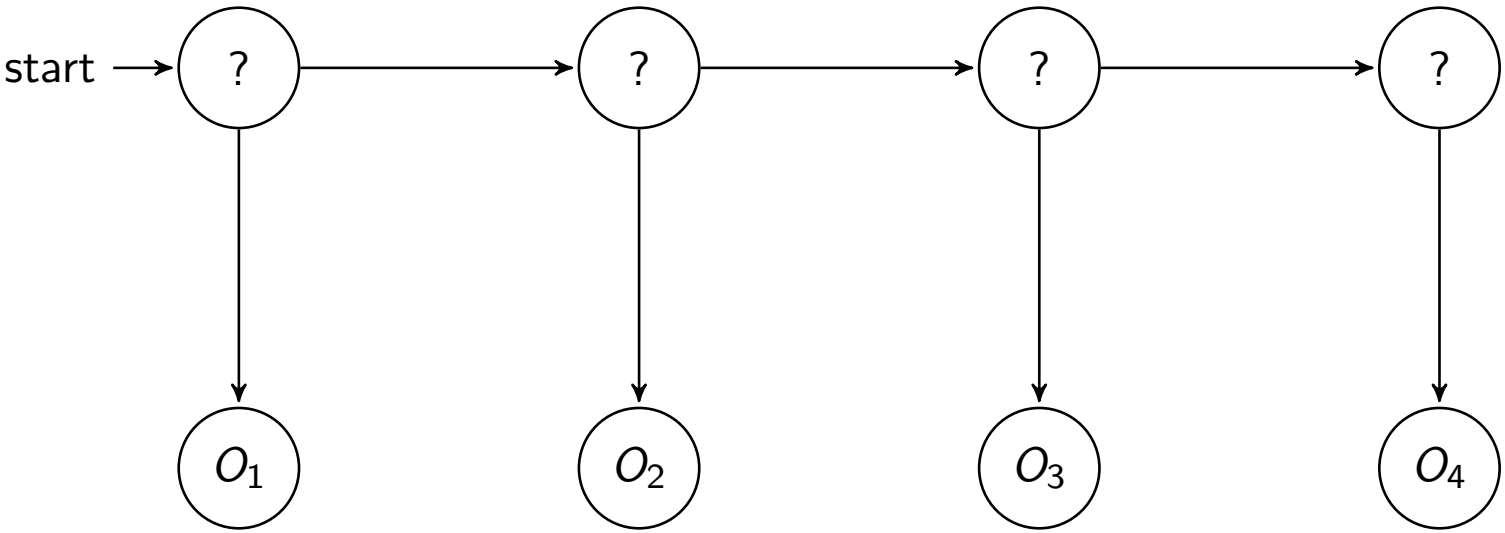
- Traditional Markov Processes are useful but...
- they assume we can determine states by direct observation.
- When primary fails, look to secondary!

Why hidden?

- Traditional Markov Processes are useful but...
- they assume we can determine states by direct observation.
- When primary fails, look to secondary!

Why hidden?

- Traditional Markov Processes are useful but...
- they assume we can determine states by direct observation.
- When primary fails, look to secondary!



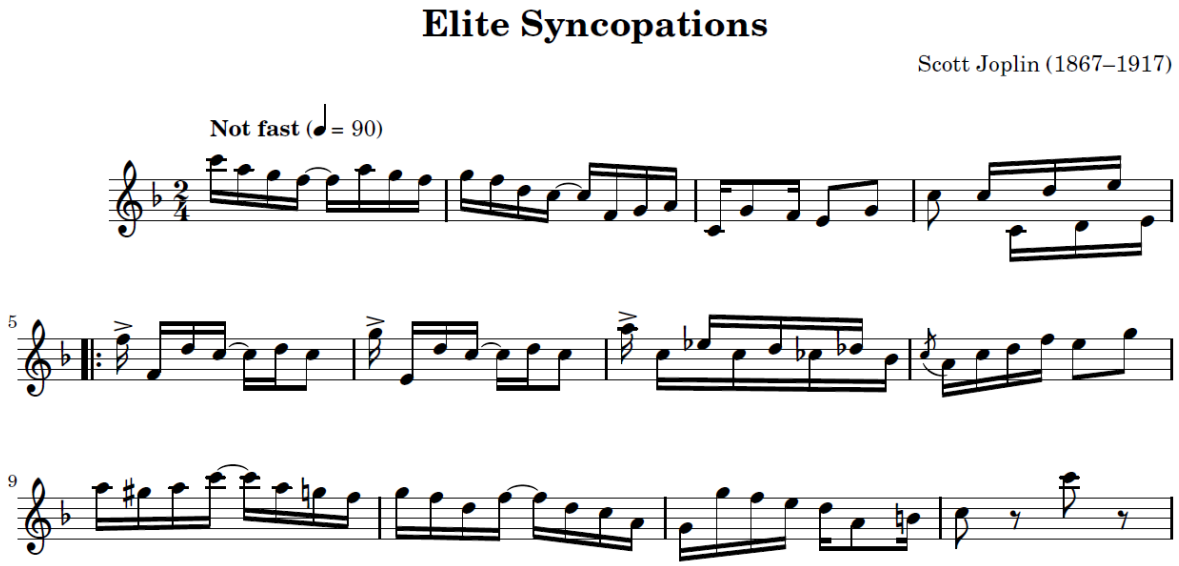
Notation

$H = \{s_1, s_2, \ldots, s_N\}$ Possible states (notes)
$P = \{o_1, o_2, \ldots, o_M\}$ Possible observations (note guesses)
$N =$ Number of states (How many notes we consider)
$M =$ Number of observations
$T =$ Length of observation sequence
$A = \{a_{ij}\}$ Transition matrix
$B = \{b_j(k)\}, b_j(k) := P(o_k|s_j)$ emissions matrix
$\pi =$ Initial distribution
$S = S_1, S_2, \ldots, S_T$ State sequence (Possible song)
$O = O_1, O_2, \ldots, O_T$ Observation sequence (possible data)
$\lambda = \{A, B, \pi\},$ the model

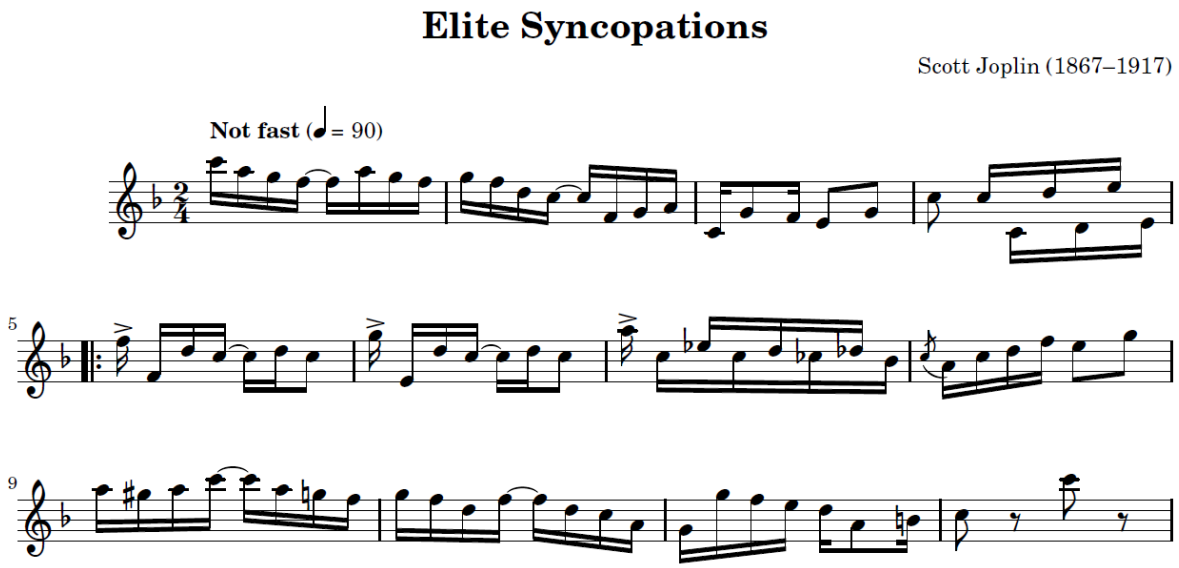
Set-up

1. creating a transition matrix

Set-up

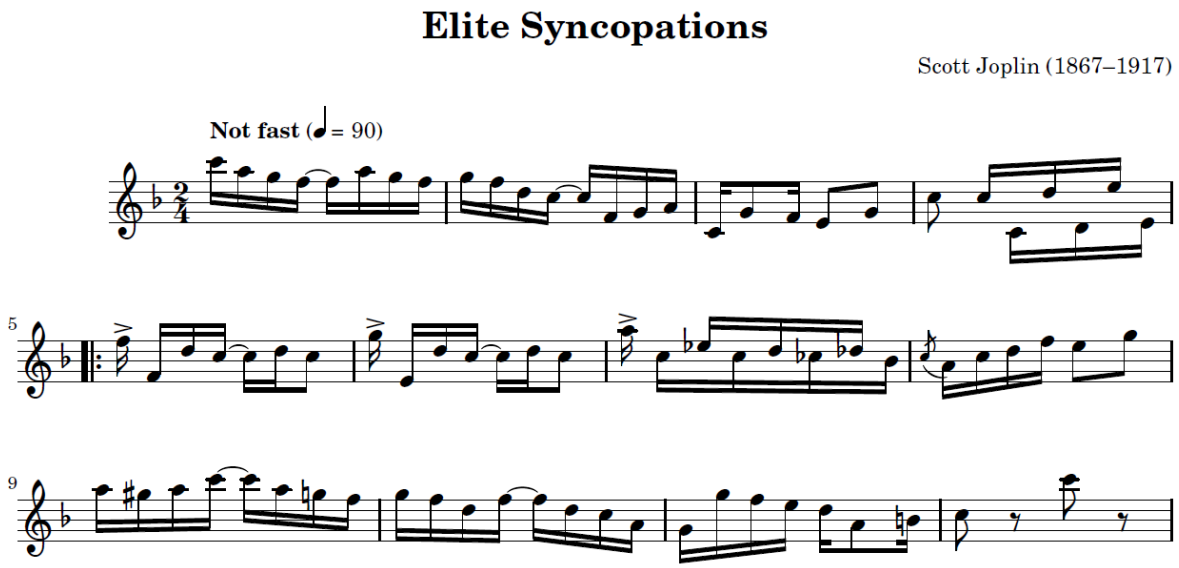
1. creating a transition matrix
2. 

Set-up

1. creating a transition matrix
2. 

   - Genre

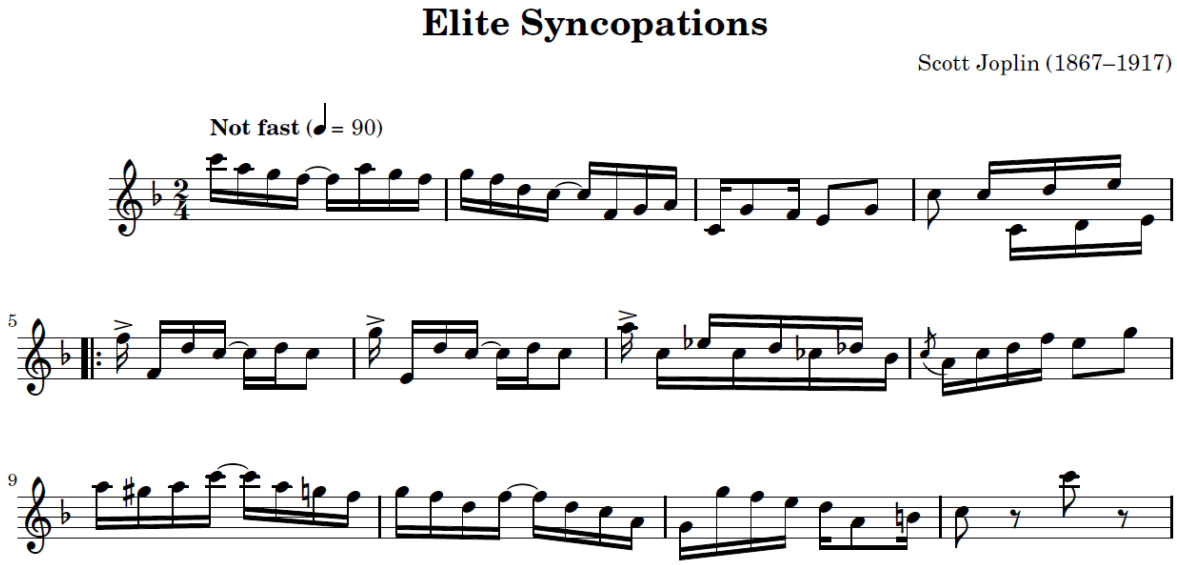
Set-up

1. creating a transition matrix
2. 

   - Genre
   - Scott Joplin/Ragtime

Set-up

1. creating a transition matrix



2.
   - Genre
   - Scott Joplin/Ragtime
3. Emissions matrix – Thankfully easy

Calculation Problems

+ $P(O|\lambda) = \sum_S P(O, S|\lambda)$

Calculation Problems

+ $P(O|\lambda) = \sum_S P(O, S|\lambda)$
  * Consider every possible note combination, i.e. every song

Calculation Problems

+ $P(O|\lambda) = \sum_S P(O, S|\lambda)$
  * Consider every possible note combination, i.e. every song
+ With N states at T observations, this yields $|N|^{|T|}$ possible paths.

Calculation Problems

+ $P(O|\lambda) = \sum_S P(O, S|\lambda)$
  * Consider every possible note combination, i.e. every song
+ With N states at T observations, this yields $|N|^{|T|}$ possible paths.
  * This is more than the number of atoms in the universe.

Calculation Problems

+ $P(O|\lambda) = \sum_S P(O, S|\lambda)$
  * Consider every possible note combination, i.e. every song
+ With N states at T observations, this yields $|N|^{|T|}$ possible paths.
  * This is more than the number of atoms in the universe.
+ This is unreasonable

Calculation Problems

+ $P(O|\lambda) = \sum_S P(O, S|\lambda)$
* Consider every possible note combination, i.e. every song
+ With N states at T observations, this yields $|N|^{|T|}$ possible paths.
* This is more than the number of atoms in the universe.
+ This is unreasonable
+ Introduce the forward variable:
  $\alpha_t(i) = P(O_1, O_2, \ldots, O_t, S_t = s_i|\lambda)$

Calculation Problems

+ $P(O|\lambda) = \sum_S P(O, S|\lambda)$
* Consider every possible note combination, i.e. every song
+ With N states at T observations, this yields $|N|^{|T|}$ possible paths.
* This is more than the number of atoms in the universe.
+ This is unreasonable
+ Introduce the forward variable:
  $\alpha_t(i) = P(O_1, O_2, \ldots, O_t, S_t = s_i|\lambda)$
* What is the probability of all note combinations up to a certain point?

Calculation Problems

+ $P(O|\lambda) = \sum_S P(O, S|\lambda)$
* Consider every possible note combination, i.e. every song
+ With N states at T observations, this yields $|N|^{|T|}$ possible paths.
* This is more than the number of atoms in the universe.
+ This is unreasonable
+ Introduce the forward variable:
  $\alpha_t(i) = P(O_1, O_2, \ldots, O_t, S_t = s_i|\lambda)$
* What is the probability of all note combinations up to a certain point?
+ Benefit: recursive definition

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \quad 1 \leq t \leq T-1,\ 1 \leq j \leq N$$

Foward and backward algorithms

- $P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$

Foward and backward algorithms

- $P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$
- $\approx |N|^{|T|} \mapsto N^2 T$ computations

Foward and backward algorithms

- $P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$
- $\approx |N|^{|T|} \mapsto N^2 T$ computations
- Backwards algorithm – By symmetry, calculate $\beta_t(i)$, the backwards variable

Viterbi

Define:

$$\delta_t(i) = \max_{s_1 s_2, \ldots s_{t-1}} P(s_1 s_2, \ldots s_t = i, O_1 O_2 \ldots O_t | \lambda)$$

Viterbi

Define:

$$\delta_t(i) = \max_{s_1 s_2, \ldots s_{t-1}} P(s_1 s_2, \ldots s_t = i, O_1 O_2 \ldots O_t | \lambda)$$

* What is the most likely note pattern (song) given the data?

Viterbi

Define:

$$\delta_t(i) = \max_{s_1 s_2, \ldots s_{t-1}} P(s_1 s_2, \ldots s_t = i, O_1 O_2 \ldots O_t | \lambda)$$

* What is the most likely note pattern (song) given the data?
+ Initialization:

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N$$

$$\phi_1(i) = 0$$

Viterbi

Define:

$$\delta_t(i) = \max_{s_1 s_2, \ldots s_{t-1}} P(s_1 s_2, \ldots s_t = i, O_1 O_2 \ldots O_t | \lambda)$$

* What is the most likely note pattern (song) given the data?
+ Initialization:

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N$$

$$\phi_1(i) = 0$$

+ Recursion:

$$\delta_t(j) = \max_{1 \leq i \leq N} (\delta_{t-1}(i) a_{ij}) b_j(O_t)$$

$$\phi_t(j) = \arg \max_{1 \leq i \leq N} (\delta_{t-1}(i) a_{ij})$$

Viterbi Cont.

+ Termination

$$P^* = \max_{1 \leq i \leq N} \delta_T(i)$$

$$s_T^* = \arg \max_{1 \leq i \leq N} \delta_T(i)$$

- Backtracking: Now work backwards from s^* and choose the best states:

$$s_t^* = \phi_{t+1}(s_{t+1}^*)$$

References

L. R. Rabine

A tutorial on hidden Markov models and selected applications in speech recognition

Processeings of the IEEE, Voll. 77, No. 2, Ferbruary 1989

http://www.cs.ucsb.edu/ cs281b/papers/HMMS

Mark Stamp (2012)

A Revealing Introduction to Hidden Markov Models

San Jose State University

M. Mauch and S. Dixon

pYIN: A Fundamental Frequency Estimator Using Probabilistic Threshold Distributions

Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing

Przemyslaw Dymarsk

Hidden Markov Models, Theory and Applications

Book available at intechweb.org