Hidden Markov Music

Daniel Wysocki

May 7, 2015

OSWEGO
STATE UNIVERSITY OF NEW YORK

Algorithmic Composition

Knowledge-based Systems

- follow a set of rules defined by the programmer
- depends on knowledge of the programmer

Machine Learning

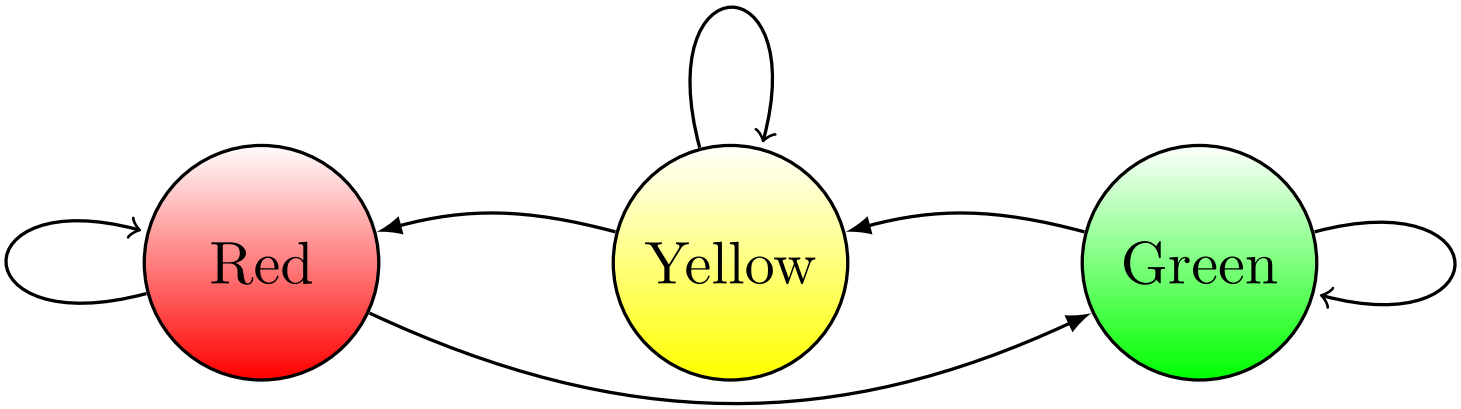
- existing compositions are used to create a model
- new compositions are produced based on the model
  - deterministic
  - probabilistic
- challenging to find a model which captures the essence of music

Markov Processes

Definition

- the future depends only on the present
- nondeterministic
- may not perfectly represent the system being modeled
  - often serves as a good approximation

Markov Chain



Training a Markov Chain

- to train a Markov chain, simply count the occurences of each transition
- divide each element by its row's total

	G	Y	R
G	45	5	0
Y	0	25	25
R	30	0	20

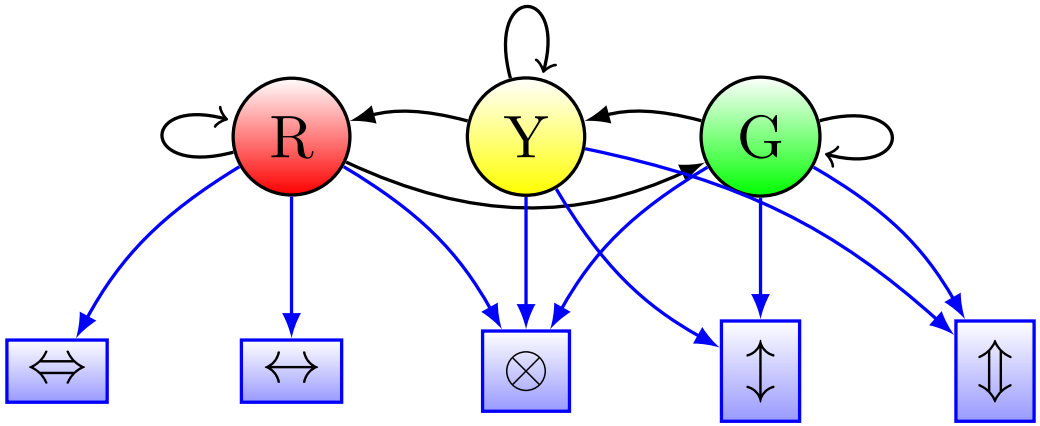
$\Rightarrow$

	G	Y	R
G	0.9	0.1	0
Y	0	0.5	0.5
R	0.6	0	0.4

Hidden Markov Model

- Marvin the Martian is looking down at a traffic light from space
- he cannot see the actual lights, but instead he sees the speed and direction of the cars
- he can still model the traffic light, using an HMM

Hidden Markov Model Example

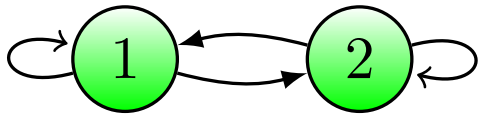


Hidden Markov Music

Overview

- we model songs as Markov processes
- notes are observed
- some underlying states of the song are hidden from us
  - we choose the number of states, and everything else is automatic
- we train the model on a song
  - allows us to generate new songs (algorithmic composition)

Model



C D E F G A B END

Hello Hello Little World

- trained a model on Twinkle, Twinkle, Little Star Play
- produced the following song Play

Für Elise (melody only)

- trained a model on Beethoven's Für Elise Play
- using 5 states Play
- using 15 states Play

Für Elise (melody and duration)

- 50 states Play
- 100 states Play

Korobeiniki (the Tetris song)

Play

- 50 states Play
- 100 states Play

Model Analysis

State Probability Distributions

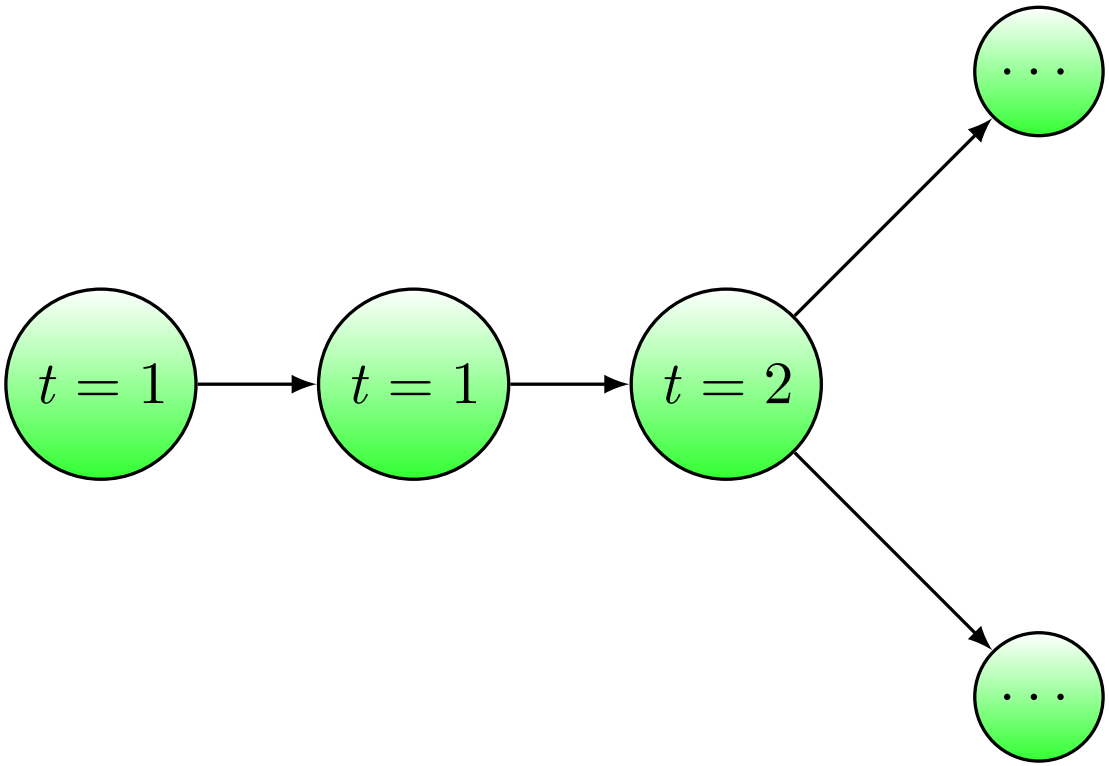
- any given state may *possibly* transition to any other state, and emit any note
- in practice, only a few are probable
- classify states according the number of probable transitions and emissions
    - classify songs according to their states

Types of States

- transition type
  - linear progression
  - branch
- emission type
  - single note
  - multi note

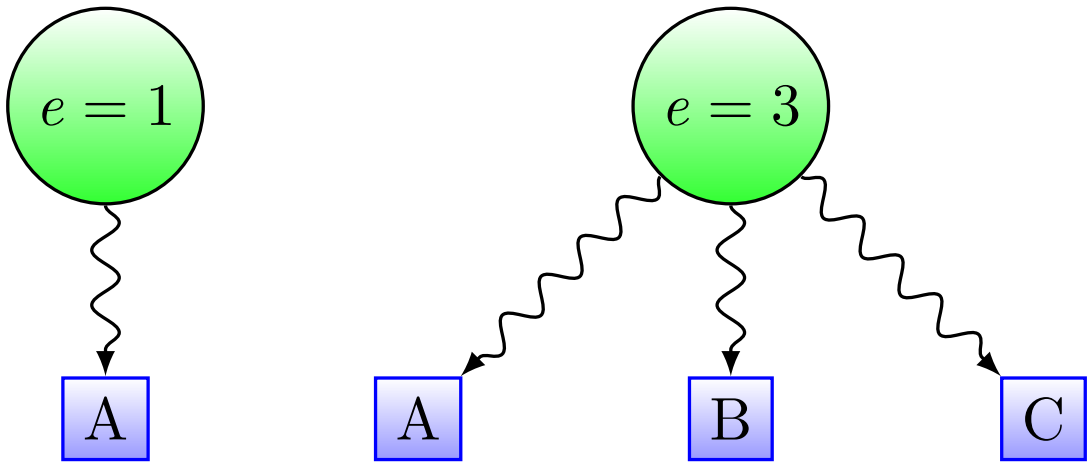
Transition Types

$t = \#$ probable transitions



Emission Types

$e = \#$ probable emissions



Song Signatures

$$\mathbf{e} = \langle e_1, e_2, e_3, \ldots \rangle, \qquad \mathbf{t} = \langle t_1, t_2, t_3, \ldots \rangle$$

$$e_i = \frac{\#\text{ states with } e = i}{\text{total }\#\text{ states}}, \qquad t_i = \frac{\#\text{ states with } t = i}{\text{total }\#\text{ states}}$$

$$\mathcal{S} = (\mathbf{e}, \mathbf{t})$$

Song Comparison

- create a model for each song
  - use some criterion for determining # states
    - Akaike information criterion (AIC)
    - Bayesian information criterion (BIC)
- measure the angles between the vectors in their signature $\mathcal{S}$
  - cosine distance

Acknowlegements

Special thanks to Craig Graci, and Andrey Markov

Questions?

Listen to more here

https://dwysocki.github.io/csc466/music.html

OSWEGO
STATE UNIVERSITY OF NEW YORK