Building And Visualizing State Transition Matrices from Images and using Markov Chains to Make New Ones

MAT 201A Final Project/Presentation
Aaron McLeran
Presentation Outline

• Brief Introduction (~1 min)
• Examples (~1 min)
• Math and Block Diagrams (~2 min)
• Motivations (~1 min)
• Project Demonstration (~9 min)
• Retrospective (~1 min)
Brief Intro

• Markov refers to Andrey Markov.

• Markov Property:
  - Given present state, future states are independent of past states

• Markov Chain:
  - Stochastic Process which exhibits Markov Property.
Markov Chains and Transition Matrices

• Future States are reached through a probabilistic process.
• Transition: moving from current state to next state
• Transition Probabilities: the probability of moving from current state to given next state
• Transition Matrix: the transition probability distribution for a finite state space describing all possible transition probabilities
Markov Chain Examples

• Random Walks
• Shannon’s “A mathematical theory of communication”
• Google’s PageRank
• Gambling and Games of Chance
• And of course, Music
Some Maths

- Formally, a Markov chain is a collection of random variables having the property that, given the present, the future is conditionally independent of the past:

\[ P(X_t = j \mid X_0 = i_0, X_1 = i_1, \ldots, X_{t-1} = i_{t-1}) = P(X_t = j \mid X_{t-1} = i_{t-1}). \]

- Here, \( t \) is the “order”.

- \( x_1, x_2, x_3, x_n \) is the markov chain

- If finite state space, the \( ij \)-th value of the transition matrix is:

\[ p_{ij} = \Pr(X_{n+1} = j \mid X_n = i). \]

- There is a lot more to it, but my project focused on implementing a simple markov chain
Motivations

• Markov Chains are used a lot in Generative Music

• Typically, melodies are encoded into transition probability matrices and their realization as a Markov Chain represents a “generated” melody.

• These melodies are usually bad, or lack a perceivable connection to original context. However, I believe the method isn’t hopeless as a tool to produce interesting art.

• I am fundamentally interested in generative/algorithmic media.
So why not do Music?

- It is a laborious chore entering note values or writing a complex midi parser.
- To get decent data, one needs a lot of examples of a particular style or composer.
- Images are a readily available data set with which there is ample state transition information.
Project Demonstration
Intro

• Used low-bit pictures to keep the number of color states to a minimum.

• Written using Objective-C, and Apple’s Cocoa framework.

• Intended to be an aesthetic visualization and interface for a real-time generative media performance application. It’s not quite there yet. (version 0.1)
The interface
Building First-Order Transition Matrix

- color = state
- line-weight = transition probability
- dot-size = non-transition probability
Building First Order Transition Matrix (ct’d)

4 ways to traverse image:

- right-left
- left-right
- top-bottom
- bottom-top

not too different....... for this image
Building Second-Order Transition Matrix

The number of states explodes, so draw differently:

With Column In center

With Row In center
Generating Images

Not too successful, not too surprising

But not too bad either
First Order Vs. Second

first-order

top-bottom traversal
not too different
both demo structures

second-order

black spots are dead
transitions (no states from transition row)
The Algorithm To Construct The Transition Matrix

- Clear previous transition matrices (if any)
- Find All Colors In the Image (the columns in the transition matrix)
- Setup Transition Matrices (one for each raster direction)
- Setup State Transition Rows (initialize an object for each row)
- Find Transitions For Image Using Desired Raster Direction
  - Find for every pixel color, the previous several pixels, depending on the order of the desired Markov chain. This represents the row of the transition matrix, i.
  - Find the index of the state column, j;
  - Increment the value in the i,j element in the transition matrix.
- Set all Transition Matrix Rows to Sum to One
The Generation Algorithm

- Prepare drawing space for desired resolution
- Setup pixel widths and heights
- Choose raster direction
- Randomly pick a starting state (from transition row)
- Iteratively choose next state based on transition matrix
- Draw each new chosen state into drawing space
Future Extensions

• Nth-Order chains should be easy to implement. The code was written as general as possible.

• For generation step, other methods of generation could be explored such as random-walk, etc.

• Generation step could be “performed” live as a visualization to generative music.
Retrospective

- Markov Transition Matrices don’t work very well in reproducing or generating normal images, at least at low orders.

- This is not surprising as normal color structures in images arise over many pixels.

- Good exercise to learn the types of algorithms used in Markov analysis.

- Good chance to learn Apple’s Objective-C graphics library or used their interface builder.
The End