Data-driven methods: Video Textures

Many slides from Alexei Efros
Let’s predict weather:

• Given today’s weather only, we want to know tomorrow’s
• Suppose weather can only be \{Sunny, Cloudy, Raining\}

The “Weather Channel” algorithm:

• Over a long period of time, record:
  – How often S followed by R
  – How often S followed by S
  – Etc.
• Compute percentages for each state:
  – P(R|S), P(S|S), etc.
• Predict the state with highest probability!
• It’s a Markov Chain
Markov Chain

What if we know today and yesterday’s weather?
## Second Order Markov Chain

<table>
<thead>
<tr>
<th>Observation</th>
<th>Tomorrow</th>
</tr>
</thead>
<tbody>
<tr>
<td>R,R</td>
<td>0.3</td>
</tr>
<tr>
<td>R,C</td>
<td>0.4</td>
</tr>
<tr>
<td>R,S</td>
<td>0.2</td>
</tr>
<tr>
<td>C,R</td>
<td>0.4</td>
</tr>
<tr>
<td>C,C</td>
<td>0.2</td>
</tr>
<tr>
<td>C,S</td>
<td>0.4</td>
</tr>
<tr>
<td>S,R</td>
<td>0.3</td>
</tr>
<tr>
<td>S,C</td>
<td>0.4</td>
</tr>
<tr>
<td>S,S</td>
<td>0.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>C</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>R,R</td>
<td>0.3</td>
<td>0.6</td>
<td>0.1</td>
</tr>
<tr>
<td>R,C</td>
<td>0.4</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>R,S</td>
<td>0.2</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>C,R</td>
<td>0.4</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>C,C</td>
<td>0.2</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>C,S</td>
<td>0.4</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>S,R</td>
<td>0.3</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>S,S</td>
<td>0.2</td>
<td>0.4</td>
<td>0.4</td>
</tr>
</tbody>
</table>

**Observation**: Today's weather conditions.

**Tomorrow**: Weather conditions to be predicted.

- **R**: Rainy
- **C**: Cloudy
- **S**: Sunny
Text Synthesis

[Shannon,’48] proposed a way to generate English-looking text using N-grams:

• Assume a generalized Markov model
• Use a large text to compute prob. distributions of each letter given N-1 previous letters
• Starting from a seed repeatedly sample this Markov chain to generate new letters
• Also works for whole words

WE NEED TO EAT CAKE
Mark V. Shaney (Bell Labs)

Results (using alt.singles corpus):

• “As I've commented before, really relating to someone involves standing next to impossible.”

• “One morning I shot an elephant in my arms and kissed him.”

• “I spent an interesting evening recently with a grain of salt”
Data Driven Philosophy

The answer to most non-trivial questions is “Let’s see what the data says”, rather than “Let’s build a parametric model” or “Let’s simulate it from the ground up”.

This can seem unappealing in its simplicity (see Chinese Room thought experiment), but there are still critical issues of representation, generalization, and data choices.

In the end, data driven methods work very well. “Every time I fire a linguist, the performance of our speech recognition system goes up.” “Fred Jelinek (Head of IBM's Speech Recognition Group), 1988
Still photos
Video clips

- American flag
- Candle flame
- Child on a swing
- Decorated tree
Video textures

- American flag waving
- Flames in the dark
- Child on a swing in a garden
- Tree decorated with balloons
Problem statement

video clip

→

video texture
Our approach

• How do we find good transitions?
Finding good transitions

Compute $L_2$ distance $D_{i,j}$ between all frames vs. frame $i$

Similar frames make good transitions
Markov chain representation

Similar frames make good transitions
Transition costs

- Transition from $i$ to $j$ if successor of $i$ is similar to $j$

  Cost function: $C_{i \rightarrow j} = D_{i+1, j}$
Transition probabilities

• Probability for transition $P_{i \rightarrow j}$ inversely related to cost:

  $$P_{i \rightarrow j} \sim \exp \left( - \frac{C_{i \rightarrow j}}{\sigma^2} \right)$$
Preserving dynamics
Preserving dynamics
Preserving dynamics

• Cost for transition $i \rightarrow j$

\[ C_{i \rightarrow j} = \sum_{k = -N}^{N - 1} w_k D_{i+k+1, j+k} \]
Preserving dynamics – effect

- Cost for transition $i \rightarrow j$

\[ C_{i \rightarrow j} = \sum_{k = -N}^{N-1} w_k D_{i+k+1, j+k} \]
Dead ends

• No good transition at the end of sequence
Future cost

- Propagate future transition costs backward
- Iteratively compute new cost

\[ F_{i \rightarrow j} = C_{i \rightarrow j} + \alpha \min_k F_{j \rightarrow k} \]
Future cost

• Propagate future transition costs backward
• Iteratively compute new cost

\[ F_{i \rightarrow j} = C_{i \rightarrow j} + \alpha \min_k F_{j \rightarrow k} \]
Future cost

- Propagate future transition costs backward
- Iteratively compute new cost

\[ F_{i \rightarrow j} = C_{i \rightarrow j} + \alpha \min_k F_{j \rightarrow k} \]
Future cost

- Propagate future transition costs backward
- Iteratively compute new cost

\[ F_{i \rightarrow j} = C_{i \rightarrow j} + \alpha \min_k F_{j \rightarrow k} \]
Future cost

- Propagate future transition costs backward
- Iteratively compute new cost
  \[ F_{i \rightarrow j} = C_{i \rightarrow j} + \alpha \min_k F_{j \rightarrow k} \]
- Q-learning
Future cost – effect
Finding good loops

- Alternative to random transitions
- Precompute set of loops up front
Video portrait

- Useful for web pages
Region-based analysis

• Divide video up into regions

• Generate a video texture for each region
Automatic region analysis
Discussion

• Some things are relatively easy
Discussion

• Some are hard
Michel Gondry train video

http://www.youtube.com/watch?v=0S43IwBF0uM