Machine Learning & Data Mining
CS/CNS/EE 155

Lecture 11:
Recent Applications
Kaggle Miniproject Closed

This leaderboard is calculated on approximately 50% of the test data. The final results will be based on the other 50%, so the final standings may be different.

<table>
<thead>
<tr>
<th>#</th>
<th>Team Name</th>
<th>Score</th>
<th>Entries</th>
<th>Last Submission UTC (Best - Last Submission)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Black Tornado</td>
<td>0.87278</td>
<td>14</td>
<td>Tue, 09 Feb 2016 21:31:00 (-1.4h)</td>
</tr>
<tr>
<td>2</td>
<td>Yellow Yakuza UnderEducated</td>
<td>0.80178</td>
<td>37</td>
<td>Mon, 08 Feb 2016 01:44:30 (-24.9h)</td>
</tr>
<tr>
<td>3</td>
<td>A.D.D.</td>
<td>0.67160</td>
<td>38</td>
<td>Sat, 06 Feb 2016 00:48:42</td>
</tr>
<tr>
<td>4</td>
<td>Human Learners</td>
<td>0.67160</td>
<td>52</td>
<td>Tue, 09 Feb 2016 19:51:39 (-0.2h)</td>
</tr>
<tr>
<td>5</td>
<td>Meng Meng Da</td>
<td>0.67012</td>
<td>38</td>
<td>Tue, 09 Feb 2016 10:05:40 (-10.5d)</td>
</tr>
<tr>
<td>6</td>
<td>The Riders of Rohan</td>
<td>0.67012</td>
<td>38</td>
<td>Tue, 09 Feb 2016 19:32:23 (-4.8d)</td>
</tr>
<tr>
<td>7</td>
<td>adhd</td>
<td>0.67012</td>
<td>26</td>
<td>Sat, 06 Feb 2016 00:25:57 (-0.1h)</td>
</tr>
<tr>
<td>8</td>
<td>Monday</td>
<td>0.67012</td>
<td>25</td>
<td>Sat, 06 Feb 2016 00:42:36 (-0.1h)</td>
</tr>
<tr>
<td>9</td>
<td>VoraciousKinkyZebras</td>
<td>0.66716</td>
<td>63</td>
<td>Tue, 09 Feb 2016 05:43:10 (-9.8d)</td>
</tr>
<tr>
<td>10</td>
<td>StickerParty</td>
<td>0.66716</td>
<td>48</td>
<td>Tue, 09 Feb 2016 04:16:34 (-5.1d)</td>
</tr>
<tr>
<td>11</td>
<td>Miss.GreenBean</td>
<td>0.66568</td>
<td>13</td>
<td>Tue, 09 Feb 2016 17:42:49 (-36.3h)</td>
</tr>
<tr>
<td>12</td>
<td>10 Points to Hufflepuff</td>
<td>0.66568</td>
<td>36</td>
<td>Tue, 09 Feb 2016 12:42:22 (-16.6h)</td>
</tr>
<tr>
<td>13</td>
<td>Victorious Secret</td>
<td>0.66568</td>
<td>13</td>
<td>Tue, 09 Feb 2016 20:04:58</td>
</tr>
<tr>
<td>14</td>
<td>NorthSide StrongSide</td>
<td>0.66420</td>
<td>41</td>
<td>Tue, 09 Feb 2016 19:56:26 (-2.8d)</td>
</tr>
<tr>
<td>15</td>
<td>Prachi</td>
<td>0.66420</td>
<td>55</td>
<td>Tue, 09 Feb 2016 21:56:13 (-2.6d)</td>
</tr>
</tbody>
</table>

This competition has completed. This leaderboard reflects the final standings.

<table>
<thead>
<tr>
<th>#</th>
<th>Team Name</th>
<th>Score</th>
<th>Entries</th>
<th>Last Submission UTC (Best - Last Submission)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Black Tornado</td>
<td>0.90869</td>
<td>14</td>
<td>Tue, 09 Feb 2016 21:31:00 (-1.4h)</td>
</tr>
<tr>
<td>2</td>
<td>Yellow Yakuza UnderEducated</td>
<td>0.83652</td>
<td>37</td>
<td>Mon, 08 Feb 2016 01:44:30 (-24.9h)</td>
</tr>
<tr>
<td>3</td>
<td>Andrew &quot;Uchiha&quot; Chico</td>
<td>0.71723</td>
<td>5</td>
<td>Tue, 09 Feb 2016 19:13:54</td>
</tr>
<tr>
<td>4</td>
<td>Human Learners</td>
<td>0.70545</td>
<td>52</td>
<td>Tue, 09 Feb 2016 19:51:39 (-4.8d)</td>
</tr>
<tr>
<td>5</td>
<td>Prachi</td>
<td>0.70250</td>
<td>55</td>
<td>Tue, 09 Feb 2016 21:56:13 (-5d)</td>
</tr>
<tr>
<td>6</td>
<td>Meng Meng Da</td>
<td>0.69809</td>
<td>38</td>
<td>Tue, 09 Feb 2016 10:05:40 (-8.2d)</td>
</tr>
<tr>
<td>7</td>
<td>VoraciousKinkyZebras</td>
<td>0.69809</td>
<td>63</td>
<td>Tue, 09 Feb 2016 05:43:10 (-0.2h)</td>
</tr>
<tr>
<td>8</td>
<td>Do you even train, bro?</td>
<td>0.69661</td>
<td>11</td>
<td>Tue, 09 Feb 2016 12:19:59 (-4.7d)</td>
</tr>
<tr>
<td>9</td>
<td>10 Points to Hufflepuff</td>
<td>0.69514</td>
<td>36</td>
<td>Tue, 09 Feb 2016 12:42:22 (-2.6d)</td>
</tr>
<tr>
<td>10</td>
<td>Nico--Nico--Ni~~~☆</td>
<td>0.69219</td>
<td>38</td>
<td>Tue, 09 Feb 2016 20:35:30 (-4.4d)</td>
</tr>
<tr>
<td>11</td>
<td>StickerParty</td>
<td>0.69072</td>
<td>48</td>
<td>Tue, 09 Feb 2016 04:16:34 (-5.1d)</td>
</tr>
<tr>
<td>12</td>
<td>D</td>
<td>0.69072</td>
<td>27</td>
<td>Tue, 09 Feb 2016 16:54:51 (3.7d)</td>
</tr>
<tr>
<td>13</td>
<td>Walker Mills</td>
<td>0.69072</td>
<td>25</td>
<td>Mon, 08 Feb 2016 23:41:48 (-24.9h)</td>
</tr>
<tr>
<td>14</td>
<td>g8</td>
<td>0.68925</td>
<td>3</td>
<td>Wed, 27 Jan 2016 04:18:49 (-24.1h)</td>
</tr>
<tr>
<td>15</td>
<td>AbysML</td>
<td>0.68925</td>
<td>8</td>
<td>Tue, 09 Feb 2016 07:37:17 (-2h)</td>
</tr>
</tbody>
</table>
Today

• Recent Applications:
  - Edge Detection
    ![Edge Detection Image]
    - ODS = 0.74, 12.5Hz
  - Speech Animation
    ![Speech Animation Image]

• Introduction to Learning Reductions
Recall: Sequence Prediction

• $X = \text{“The Dog Jumped Over the Fence”}$
• $Y = D N V P D N$

$\begin{align*}
y^1 & \quad y^2 & \quad y^3 & \quad \ldots & \quad y^M \\
x^1 & \quad x^2 & \quad x^3 & \quad \ldots & \quad x^M
\end{align*}$
Recall: Conditional Random Field

\[ P(y \mid x) = \frac{1}{Z(x)} \exp \{ F(x, y) \} \]

\[ F(y, x) \equiv \sum_{j=1}^{M} \left[ w^T \varphi^j(y^j, y^{j-1} \mid x) \right] \]

\[ \varphi^j(a, b \mid x) = \begin{bmatrix} \varphi_1(a \mid x^j) \\ \varphi_2(a, b) \end{bmatrix} \]
Limitations of CRFs

• Linear model
  – Requires good feature representation

• Only first-order effects
  – Cannot model higher-order dependencies

\[
\begin{align*}
\varphi_2(y^1, y^0) & \quad \varphi_2(y^2, y^1) & \quad \varphi_2(y^3, y^2) \\
\varphi_1(y^1 | x^1) & \quad \varphi_1(y^2 | x^2) & \quad \varphi_1(y^3 | x^3) & \quad \varphi_1(y^M | x^M)
\end{align*}
\]
Fig. 3. Illustration of edge detection results on the BSDS500 dataset on five sample images. The first two rows show the original image and ground truth. The next three rows contain results for gPb-owt-ucm [1], Sketch Tokens [3], and SCG [4]. The final four rows show our results for variants of SE. Use viewer zoom functionality to see fine details.
2D Conditional Random Field

• Each $y_{ij}$ is binary label
  – Edge or Not Edge

• What features?
  – Defined over pixels?
Today: Learning Reductions

• Convert complicated problem into simpler ones
  – Use complex models for simpler problems
  – E.g., decision trees, neural nets

• Recompose predictions for complicated problem
Strong Local Properties

• Local patterns matter
  – E.g., image patches

• Complex relationship
  – Non-linear

Fig. 3. Illustration of edge detection results on the BSDS500 dataset on five sample images. The first two rows show the original image and ground truth. The next three rows contain results for gPb-owt-ucm, Sketch Tokens, and SCG. The final four rows show our results for variants of SE. Use viewer zoom functionality to see fine details.
Weak Global Properties

• Edge detections local

• No need to fully connect model
Sliding Window Approach

• Train model to predict patches
  – E.g., 16x16

• Slide across image

• What model?
**Recall: Binary Decision Tree**

Every **internal node** has a **binary query function** \( q(x) \).

Every **leaf node** has a prediction, e.g., 0 or 1.

**Prediction starts at root node.**
Recursively calls query function.
Positive response ➔ Left Child.
Negative response ➔ Right Child.
Repeat until Leaf Node.

**Input:** Alice
Gender: Female
Age: 14

**Prediction:** Height > 55”
Structured Decision Tree

• Each leaf node predicts a 16x16 edge matrix
  – Average of all training patch labels

• Prediction is very fast!
  – Slide predictor across image, average results
  – No need for Viterbi-type algorithms

• What is splitting criterion?
• What is query set?
Finally, we define how to combine a set of structured labels into two classes (d). Given the class labels, a standard splitting criterion, such as Gini impurity, may be used (e). We begin by augmenting each image patch with multiple input features: 

- Domain information, resulting in a feature vector \( z \).
- Input features, but not vice versa. We utilize both.
- Edge map for each image patch independently and merge overlapping contours.

The biggest limitation is that any prediction (either for training a leaf node or merging the output of multiple trees), hence using a coarse distance metric suffices.

We assume we are given a set of segmented training images, \( \{ y_1, y_2, \ldots, y_k \} \), such that labels \( y \) are binary (structured labels into two classes). Given the class labels, a standard splitting criterion, such as Gini impurity, may be used.

Both approaches perform similarly: 

1. **Structured Information Gain**
   - The discrete labels may be binary (contrast with \( k = 2 \) vs \( k = 256 \)).
   - We begin by augmenting each image patch with multiple input features.
   - With approximations, we can quantize the distribution of labels at a given node (contrast with \( k = 2 \) vs \( k = 256 \)).
   - Given the mapping functions \( \pi : Y \rightarrow Z \) proposed such an approach, but due to its complexity, \( k = 256 \) is necessary to accurately capture similarities of information, resulting in a feature vector \( z \).
2. **Random Forests for Fast Edge Detection**
   - The biggest limitation is that any prediction (either for training a leaf node or merging the output of multiple trees), hence using a coarse distance metric suffices.
   - We begin by augmenting each image patch with multiple input features.
   - With approximations, we can quantize the distribution of labels at a given node (contrast with \( k = 2 \) vs \( k = 256 \)).
   - Given the mapping functions \( \pi : Y \rightarrow Z \) proposed such an approach, but due to its complexity, \( k = 256 \) is necessary to accurately capture similarities of information, resulting in a feature vector \( z \).

- **Random Forests for Fast Edge Detection**
  - The biggest limitation is that any prediction (either for training a leaf node or merging the output of multiple trees), hence using a coarse distance metric suffices.
  - We begin by augmenting each image patch with multiple input features.
  - With approximations, we can quantize the distribution of labels at a given node (contrast with \( k = 2 \) vs \( k = 256 \)).
  - Given the mapping functions \( \pi : Y \rightarrow Z \) proposed such an approach, but due to its complexity, \( k = 256 \) is necessary to accurately capture similarities of information, resulting in a feature vector \( z \).

- **Structured Information Gain**
  - The discrete labels may be binary (contrast with \( k = 2 \) vs \( k = 256 \)).
  - We begin by augmenting each image patch with multiple input features.
  - With approximations, we can quantize the distribution of labels at a given node (contrast with \( k = 2 \) vs \( k = 256 \)).
  - Given the mapping functions \( \pi : Y \rightarrow Z \) proposed such an approach, but due to its complexity, \( k = 256 \) is necessary to accurately capture similarities of information, resulting in a feature vector \( z \).

---

“Structured Random Forests for Fast Edge Detection”
Dollár & Zitnick, ICCV 2013
Structured Information Gain

1. First map labels to coordinate system
   A. For each coordinate, choose pair of pixels
   B. Set coordinate to 1 if in same segment, 0 o.w.
      • Coordinate 1 = 0
      • Coordinate 2 = 1
      • Etc...

2. Cluster training labels

(Actual approach more complicated.)

“Structured Random Forests for Fast Edge Detection”
Dollár & Zitnick, ICCV 2013
Multiclass Entropy

- Reduced training labels to K clusters
  - Can treat as multiclass classification
- Impurity measure = multiclass entropy

![Diagram showing decision tree with multiclass labels]

- Impurity measure based on Shannon entropy or Gini impurity as gain criteria.
- The discrete labels may be binary (2 clusters) or more, e.g., 4 clusters (Fig. 2).
- The discrete labels may be binary (2 clusters) or more, e.g., 4 clusters (Fig. 2).
- The discrete labels may be binary (2 clusters) or more, e.g., 4 clusters (Fig. 2).
- The discrete labels may be binary (2 clusters) or more, e.g., 4 clusters (Fig. 2).
• Features about color gradients
  – Image gets darker from column 1 to column 5
  – Image gets more blue from row 7 to row 3
  – Etc...
  – 7228 features total

(Actual approach more complicated.)

“Structured Random Forests for Fast Edge Detection”
Dollár & Zitnick, ICCV 2013
Putting it Together

• Create new training set $\hat{S} = \{(x,\hat{y})\}$
  – $x = 16\times16$ image patch
  – $\hat{y} = 16\times16$ ground truth edges

• Train structured DT on $\hat{S}$

• Predict by sliding DT over input image
  – Average predictions

(Actual approach more complicated.)

“Structured Random Forests for Fast Edge Detection”
Dollár & Zitnick, ICCV 2013
Fig. 3. Illustration of edge detection results on the BSDS500 dataset on five sample images. The first two rows show the original image and ground truth. The next three rows contain results for gPb-owt-ucm, Sketch Tokens, and SCG. The final four rows show our results for variants of SE. Use viewer zoom functionality to see fine details.
Comparable accuracy vs state-of-the-art

Much faster!

<table>
<thead>
<tr>
<th></th>
<th>ODS</th>
<th>OIS</th>
<th>AP</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>.80</td>
<td>.80</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Canny</td>
<td>.60</td>
<td>.64</td>
<td>.58</td>
<td>15</td>
</tr>
<tr>
<td>Felz-Hutt [11]</td>
<td>.61</td>
<td>.64</td>
<td>.56</td>
<td>10</td>
</tr>
<tr>
<td>Hidayat-Green [16]</td>
<td>.62†</td>
<td>-</td>
<td>-</td>
<td>20</td>
</tr>
<tr>
<td>BEL [9]</td>
<td>.66†</td>
<td>-</td>
<td>-</td>
<td>1/10</td>
</tr>
<tr>
<td>gPb + GPU [6]</td>
<td>.70†</td>
<td>-</td>
<td>-</td>
<td>1/2†</td>
</tr>
<tr>
<td>gPb [1]</td>
<td>.71</td>
<td>.74</td>
<td>.65</td>
<td>1/240</td>
</tr>
<tr>
<td>gPb-owt-ucm [1]</td>
<td>.73</td>
<td>.76</td>
<td>.73</td>
<td>1/240</td>
</tr>
<tr>
<td>Sketch tokens [21]</td>
<td>.73</td>
<td>.75</td>
<td>.78</td>
<td>1</td>
</tr>
<tr>
<td>SCG [31]</td>
<td>.74</td>
<td>.76</td>
<td>.77</td>
<td>1/280</td>
</tr>
<tr>
<td>SE-SS, T=1</td>
<td>.72</td>
<td>.74</td>
<td>.77</td>
<td>60</td>
</tr>
<tr>
<td>SE-SS, T=4</td>
<td>.73</td>
<td>.75</td>
<td>.77</td>
<td>30</td>
</tr>
<tr>
<td>SE-MS, T=4</td>
<td>.74</td>
<td>.76</td>
<td>.78</td>
<td>6</td>
</tr>
</tbody>
</table>

“Structured Random Forests for Fast Edge Detection”
Dollár & Zitnick, ICCV 2013
Speech Animation
Automatically Animate to Input Audio?
(Given Training Data)

“A Decision Tree Framework for Spatiotemporal Sequence Prediction”
Kim, Yue, Taylor, Matthews, KDD 2015, http://projects.yisongyue.com/visual_speech
Training Data

• ~2500 Sentences
  – Recorded at 30 Hz
  – ~10 hours of recorded speech

• Active Appearance Model
  – Actor’s lower face
  – 30 degrees of freedom (also 100+)

Data from [Taylor et al., 2012]
Prediction Task

Input sequence

\[ X = \langle x_1, x_2, \ldots, x_{|x|} \rangle \]

Output sequence

\[ Y = \langle y_1, y_2, \ldots, y_{|y|} \rangle, \quad y_t \in R^D \]

**Goal:** learn predictor

\[ h : X \rightarrow Y \]

**X**

<table>
<thead>
<tr>
<th>Frame</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token</td>
<td>-</td>
<td>p</td>
<td>p</td>
<td>r</td>
<td>ih</td>
<td>ih</td>
<td>d</td>
<td>d</td>
<td>ih</td>
<td>ih</td>
<td>ih</td>
<td>ih</td>
<td>ih</td>
<td>k</td>
<td>k</td>
<td>sh</td>
<td>sh</td>
<td>sh</td>
<td>sh</td>
<td>sh</td>
<td>uh</td>
<td>uh</td>
</tr>
</tbody>
</table>

**Y**

- Dimension 1

<table>
<thead>
<tr>
<th>Frame number</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
<th>22</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values</td>
<td>-50</td>
<td>-20</td>
<td>0</td>
<td>20</td>
<td>50</td>
<td>80</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Phoneme sequence**

**Sequence of face configurations**
Temporal curvature can vary smoothly or sharply
(Changes are context-dependent – this is the co-articulation problem)

\( X \)

Frame
Token

\( Y \)

Dimension 1

Minimal long-range dependencies
(prediction = construction = election...)

<table>
<thead>
<tr>
<th>Frame</th>
<th>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token</td>
<td>p p r ih ih d d ih ih ih ih k k sh sh sh sh uh uh n</td>
</tr>
</tbody>
</table>

(Due to the assumption that there are salient dependencies such as observed tracking correspondences, effective frame-by-frame warping misalignments can be handled efficiently, where input/output pairs are drawn from a standard animation sequence. For speech synthesis, speech can be maintained at the start and end of the tracked sequence and audio such as spelling, which leads to missing values in the resulting output sequence. Despite this, different people may form somewhat different lip shapes while speaking the same sentence. In that sense, one can consider all such observed trackings or imperfections in performing certain actions. For example, in visual speech animation, or in any case where natural temporal variability in the phenomenon being studied can be tolerated due to the assumption that there are salient dependencies such as observed tracking correspondences, effective frame-by-frame warping misalignments can be handled efficiently, where input/output pairs are drawn from a standard animation sequence. For speech synthesis, speech can be maintained at the start and end of the tracked sequence and audio such as spelling, which leads to missing values in the resulting output sequence. Despite this, different people may form somewhat different lip shapes while speaking the same sentence. In that sense, one can consider all such observed trackings or imperfections in performing certain actions. For example, in visual speech animation, or in any case where natural temporal variability in the phenomenon being studied can be tolerated due to the assumption that there are salient dependencies such as observed tracking correspondences, effective frame-by-frame warping misalignments can be handled efficiently, where input/output pairs are drawn from a standard animation sequence. For speech synthesis, speech can be maintained at the start and end of the tracked sequence and audio such as spelling, which leads to missing values in the resulting output sequence. Despite this, different people may form somewhat different lip shapes while speaking the same sentence. In that sense, one can consider all such observed trackings or imperfections in performing certain actions. For example, in visual speech animation, or in any case where natural temporal variability in the phenomenon being studied can be tolerated due to the assumption that there are salient dependencies such as observed tracking correspondences, effective frame-by-frame warping misalignments can be handled efficiently, where input/output pairs are drawn from a standard animation sequence. For speech synthesis, speech can be maintained at the start and end of the tracked sequence and audio such as spelling, which leads to missing values in the resulting output sequence. Despite this, different people may form somewhat different lip shapes while speaking the same sentence. In that sense, one can consider all such observed trackings or imperfections in performing certain actions. For example, in visual speech animation, or in any case where natural temporal variability in the phenomenon being studied can be tolerated due to the assumption that there are salient dependencies such as observed tracking correspondences, effective frame-by-frame warping misalignments can be handled efficiently, where input/output pairs are drawn from a standard animation sequence. For speech synthesis, speech can be maintained at the start and end of the tracked sequence and audio such as spelling, which leads to missing values in the resulting output sequence. Despite this, different people may form somewhat different lip shapes while speaking the same sentence. In that sense, one can consider all such observed trackings or imperfections in performing certain actions. For example, in visual speech animation, or in any case where natural temporal variability in the phenomenon being studied can be tolerated due to the assumption that there are salient dependencies such as observed tracking correspondences, effective frame-by-frame warping misalignments can be handled efficiently, where input/output pairs are drawn from a standard animation sequence. For speech synthesis, speech can be maintained at the start and end of the tracked sequence and audio such as spelling, which leads to missing values in the resulting output sequence. Despite this, different people may form somewhat different lip shapes while speaking the same sentence. In that sense, one can consider all such observed trackings or imperfections in performing certain actions. For example, in visual speech animation, or in any case where natural temporal variability in the phenomenon being studied can be tolerated due to the assumption that there are salient dependencies such as observed tracking correspondences, effective frame-by-frame warping misalignments can be handled efficiently, where input/output pairs are drawn from a standard animation sequence. For speech synthesis, speech can be maintained at the start and end of the tracked sequence and audio such as spelling, which leads to missing values in the resulting output sequence. Despite this, different people may form somewhat different lip shapes while speaking the same sentence. In that sense, one can consider all such observed trackings or imperfections in performing certain actions. For example, in visual speech animation, or in any case where natural temporal variability in the phenomenon being studied can be tolerated due to the assumption that there are salient dependencies such as observed tracking correspondences, effective frame-by-frame warping misalignments can be handled efficiently, where input/output pairs are drawn from a standard animation sequence. For speech synthesis, speech can be maintained at the start and end of the tracked sequence and audio such as spelling, which leads to missing values in the resulting output sequence. Despite this, different people may form somewhat different lip shapes while speaking the same sentence. In that sense, one can consider all such observed trackings or imperfections in performing certain actions. For example, in visual speech animation, or in any case where natural temporal variability in the phenomenon being studied can be tolerated due to the assumption that there are salient dependencies such as observed tracking correspondences, effective frame-by-frame warping misalignments can be handled efficiently, where input/output pairs are drawn from a standard animation sequence. For speech synthesis, speech can be maintained at the start and end of the tracked sequence and audio such as spelling, which leads to missing values in the resulting output sequence. Despite this, different people may form somewhat different lip shapes while speaking the same sentence. In that sense, one can consider all such observed trackings or imperfections in performing certain actions. For example, in visual speech animation, or in any case where natural temporal variability in the phenomenon being studied can be tolerated due to the assumption that there are salient dependencies such as observed tracking correspondences, effective frame-by-frame warping misalignments can be handled efficiently, where input/output pairs are drawn from a standard animation sequence. For speech synthesis, speech can be maintained at the start and end of the tracked sequence and audio such as spelling, which leads to missing values in the resulting output sequence. Despite this, different people may form somewhat different lip shapes while speaking the same sentence. In that sense, one can consider all such observed trackings or imperfections in performing certain actions. For example, in visual speech animation, or in any case where natural temporal variability in the phenomenon being studied can be tolerated due to the assumption that there are salient dependencies such as observed tracking correspondences, effective frame-by-frame warping misalignments can be handled efficiently, where input/output pairs are drawn from a standard animation sequence. For speech synthesis, speech can be maintained at the start and end of the tracked sequence and audio such as spelling, which leads to missing values in the resulting output sequence. Despite this, different people may form somewhat different lip shapes while speaking the same sentence. In that sense, one can consider all such observed trackings or imperfections in performing certain actions. For example, in visual speech animation, or in any case where natural temporal variability in the phenomenon being studied can be tolerated due to the assumption that there are salient dependencies such as observed tracking correspondences, effective frame-by-frame warping misalignments can be handled efficiently, where input/output pairs are drawn from a standard animation sequence. For speech synthesis, speech can be maintained at the start and end of the tracked sequence and audio such as spelling, which leads to missing values in the resulting output sequence. Despite this, different people may form somewhat different lip shapes while speaking the same sentence. In that sense, one can consider all such observed trackings or imperfections in performing certain actions. For example, in visual speech animation, or in any case where natural temporal variability in the phenomenon being studied can be tolerated due to the assumption that there are salient dependencies such as observed tracking correspondences, effective frame-by-frame warping misalignments can be handled efficiently, where input/output pairs are drawn from a standard animation sequence. For speech synthesis, speech can be maintained at the start and end of the tracked sequence and audio such as spelling, which leads to missing values in the resulting output sequence. Despite this, different people may form somewhat different lip shapes while speaking the same sentence. In that sense, one can consider all such observed trackings or imperfections in performing certain actions. For example, in visual speech animation, or in any case where natural temporal variability in the phenomenon being studied can be tolerated due to the assumption that there are salient dependencies such as observed tracking correspondences, effective frame-by-frame warping misalignments can be handled efficiently, where input/output pairs are drawn from a standard animation sequence. For speech synthesis, speech can be maintained at the start and end of the tracked sequence and audio such as spelling, which leads to missing values in the resulting output sequence. Despite this, different people may form somewhat different lip shapes while speaking the same sentence. In that sense, one can consider all such observed trackings or imperfections in performing certain actions. For example, in visual speech animation, or in any case where natural temporal variability in the phenomenon being studied can be tolerate...
Strong Local Properties

• Need to model arbitrary local curvature

• Not well suited by linear chain models!

![Diagram of X and Y sequences with token examples and frame numbers, illustrating the need for models to handle arbitrary local curvature and the inability of linear chain models to capture long-range dependencies.]
Weak Global Properties

- No need to model entire chain directly

Minimal long-range dependencies
(prediction = construction = election...)

- Motivates sliding window approach!
Input speech: “PREDICTION”

Frame 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

x Token - p p r ih ih d d ih ih ih ih k k sh sh sh sh uh uh n -

\( \hat{x}_1, \hat{x}_2, \ldots \)

\( h(\hat{x}) \)

\( \hat{y}_1, \hat{y}_2, \ldots \)

Overlapping Sliding Window of Inputs

Decision Tree Model
150-variate regression

This is the only thing that requires machine learning!

Aggregate Outputs
Very fast!
Training

Original Training Data
(Variable-Length Trajectory Prediction)

Modified Training Data
(Fixed-Length Multivariate Regression)

Train Decision Tree
(Or some other regression model)
Query Set for Speech Animation

- Frame 8 is a vowel that contains /a/?
  - yes
  - no
    - Frame 8 is a long vowel?
      - yes
      - no
        - Frame 6 is a vowel that contains /a/?
          - yes
          - no
        - Frame 8 is a vowel that contains /o/?
          - yes
          - no
        - Frame 8 is a sibilant consonant?
          - yes
          - no
            - Frame 8 is a post-alveolar consonant?
              - yes
              - no
            - Frame 8 is a front consonant?
              - yes
              - no

Frames indexed by 1-11 (center is frame 6)

Full tree has 5K+ leaf nodes
Multivariate Regression Tree

• **Prediction:**

  ![Training Data in Leaf Node: 150](image)

  Prediction: $\hat{y}_{Leaf} = $ Mean

• **Training loss:** multivariate squared loss:

$$\sum_{Leaf} \sum_{\hat{y} \in Leaf} \left\| \hat{y}_{Leaf} - \hat{y} \right\|^2$$
Prediction on New Speaker

“A Decision Tree Framework for Spatiotemporal Sequence Prediction”
Kim, Yue, Taylor, Matthews, KDD 2015, http://projects.yisongyue.com/visual_speech
Prediction on New Speaker

“A Decision Tree Framework for Spatiotemporal Sequence Prediction”
Kim, Yue, Taylor, Matthews, KDD 2015, http://projects.yisongyue.com/visual_speech
**Input speech:** “LEARNING”

Frame | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22
(a) \( \mathbf{x} \) | - | l | l | l | l | er | er | er | n | n | n | iy | iy | ng | ng | ng | ng | g | g | g | g | -

(b) \( \hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots \)

(c) \( h(\hat{\mathbf{x}}) \)

(d) \( \hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots \)

(e) \( \mathbf{y} \)
Input speech: “S I G G R A P H”

(a) \( \mathbf{x} \)  

| Frame | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Label | - | s | s | s | ih | ih | ih | g | g | g | r | r | ae | ae | ae | ae | f | f | f | f | - |

(b) \( \hat{x}_1, \hat{x}_2, \ldots \)

(c) \( h(\hat{x}) \)

(d) \( \hat{y}_1, \hat{y}_2, \ldots \)

(e) \( y \)
Side-by-Side User Study

Comparing our approach versus competitor on 50 held-out test sentences.

“A Decision Tree Framework for Spatiotemporal Sequence Prediction”
Kim, Yue, Taylor, Matthews, KDD 2015, http://projects.yisongyue.com/visual_speech
Side-by-Side User Study

Comparing our approach versus competitor on 50 held-out test sentences.
Comparison with Ground Truth

We under-articulate relative to ground truth!
(Could be solved with more training data...)

“A Decision Tree Framework for Spatiotemporal Sequence Prediction”
Kim, Yue, Taylor, Matthews, KDD 2015, http://projects.yisongyue.com/visual_speech
Comparison with Ground Truth

Figure 4: (Top) Ground truth AAM parameter 1 for sentence 21 (black) and the corresponding parameter trajectories generated using decision tree regression (red), HMM synthesis (green) and dynamic visemes (blue). (Bottom) Frame-by-frame error over sentence 21 for the three synthesis methods. Sentence 21 has a mid-range decision tree MSE.

Figure 5: (Top) Ground truth AAM parameter 1 for sentence 31 (black) and the corresponding parameter trajectories generated using decision tree regression (red), HMM synthesis (green) and dynamic visemes (blue). (Bottom) Frame-by-frame error over sentence 31 for the three synthesis methods. Sentence 31 has a mid-range decision tree MSE.

Figure 6: (Top) Ground truth AAM parameter one for sentence 4 (black) and the corresponding parameter trajectories generated using decision tree regression (red), HMM synthesis (green) and dynamic visemes (blue). (Bottom) Frame-by-frame error over sentence 4 for the three synthesis methods. Sentence 4 has the lowest decision tree MSE.

Figure 7: Max square error for each of the 50 test sentences with the examples that are shown in this document highlighted.

Figure 8: Mid-range square error measured against ground truth (tracked) AAM parameters. The sentence with the highest max square error is the same as that with the highest mean square error and can be seen in Figure 3.

Figure 9: (Top) Ground truth AAM parameter one for sentence 1 (black) and the corresponding parameter trajectories generated using decision tree regression (red), HMM synthesis (green) and dynamic visemes (blue). (Bottom) Frame-by-frame error over sentence 1 for the three synthesis methods. Sentence 1 has the lowest decision tree max SE.

The highlighted sentences represent those corresponding to decision tree predictions with mid-range (Figure 8) and lowest (Figure 9) max square error measured against ground truth (tracked) AAM parameters.

2.2 Ordered on Max SE

The following examples show predicted AAM trajectories and frame-wise square error for the sentences highlighted in Figure 7.
Is the phoneme in the 8th frame a diphthong?
Y
Is the phoneme in the 8th frame a semivowel?
Y
N
N

Is the phoneme in the 3rd frame articulated at the back of the mouth?

"SIGGRAPH"

Realistic Speech Animation

Target Speech

Decision Tree

Input Audio

Speech Recognition

Speech Animation

Retargeting
E.g., [Sumner & Popovic 2004]

(chimp rig courtesy of Hao Li)

Editing
Aside: Retargeting

Reference face ➔ target face

(Semi-)Automatic:
Deformation Transfer [Sumner & Popovic 2004]
Finds linear transform (requires reference pose)

Manual:
Pose basis shapes & linear blending
Prediction for Very Different Language
Prediction for Very Different Language
Overview of Learning Reductions
Motivation

• Know how to solve “standard” ML problems
  – Classification, regression, etc.
  – SVMs, logistic regression, decision trees, neural nets, etc.

• “Reduce” complex problems to simple ones?
  – Variable-length trajectories \( \rightarrow \) multivariate regression

• Similar to other reduction problems
  – E.g., NP-complete reductions
  – Some learning reductions have provable guarantees

Many toolkits available!

Still non-trivial!
Other Learning Reductions

- Multiclass $\Rightarrow$ Binary
- Cost-weighted $\Rightarrow$ Unweighted
- Ranking $\Rightarrow$ Binary
- Sequential $\Rightarrow$ Multiclass
- And many more...

http://hunch.net/~jl/projects/reductions/reductions.html
Why Multiclass ➔ Binary?

• Conventional approach: one-versus-all
  – Scoring function per class
  – Predict class with highest score

• Limitations:
  – Linear in #classes
  – Hard to prove generalization bounds
  – (Binary SVM analyzes generalization via margin)
Learning Reduction Recipe

• Given original training set: \( S = \{(x_i, y_i)\}_{i=1}^N \)

• Create modified training set(s):
  \[
  \hat{S} = \{(x_i, \hat{y}_i)\}_{i=1}^N
  \]

  – Train \( \hat{h}'s \) on \( \hat{S}'s \)

• Final \( h = \) combining predictions \( \hat{h}'s \)
Two Flavors of Analysis

• Error Reduction:
  – Each $\hat{h}$ achieves 0/1 Loss $\varepsilon$
  – Implication for multiclass 0/1 loss of $h$?
    • Answer: $(K-1)\varepsilon$

• Regret Reduction:
  – Each $\hat{h}$ achieves 0/1 regret $r$
  – Implication of multiclass regret?
    • E.g., $Kr$?
  – More powerful result

$$\varepsilon = L_P(w)$$
Zero 0/1 Test Error typically not possible

$$r = L_P(w) - L_P(w^*)$$

Recall: 'Illustration' Lecture '11: 'Learning Reducions' & 'Recent Applicaions' of 'Decision Trees'

<table>
<thead>
<tr>
<th>Training Set Size</th>
<th>Test Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^4$</td>
<td>0.1</td>
</tr>
<tr>
<td>$10^5$</td>
<td>0.2</td>
</tr>
<tr>
<td>$10^6$</td>
<td>0.3</td>
</tr>
<tr>
<td>$10^7$</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Bayes' Optimal

Consistent Learning Algorithm

Inconsistent Learning Algorithm

"Regret"
Aside: Sliding Window Regression

• If base model $\hat{h}$ has 0 error
  – Then sliding window prediction has 0 error

• What about when $\hat{h}$ has $>0$ error?
  – As regret of $\hat{h}$ decreases...
  – ... decrease in regret of $h$?
  – Open question!
    • Need to formalize lack of global dependencies
Filter Tree for Multiclass ➔ Binary

Each base model is a binary classifier

http://mi.eng.cam.ac.uk/~mjfg/local/Projects/filter_tree.pdf
The Learning Reduction

• First Layer
  – Train each $h_{ij}$ using

$$S_{ij} = \left\{ (x, 1_{[y=i]}) \mid \forall (x, y) \in S : y \in \{i, j\} \right\}$$

Each base model is a binary classifier.
The Learning Reduction

- **Second Layer**
  - Train $h_{\text{Left,Right}}$ using

$$S_{\text{Left,Right}} = \left\{ (x, 1_{[y \in \{L,R\}]} ) \mid \forall (x,y) \in S : y \in \{1, ..., 4\} \land (\text{no mistake by } h_{12}, h_{34}) \right\}$$

Each base model is a binary classifier.

Train Lower Layers only using mistake-free training data.
The Learning Reduction

• Classification problem dependent on classifiers learned in previous layers

• Reduction happens iteratively
  – I.e., adaptively

Each base model is a binary classifier
Recall: Two Flavors of Analysis

• Error Reduction:
  – Each \( \hat{h} \) achieves 0/1 Loss \( \varepsilon \)
  – Implication for multiclass 0/1 loss of \( h \)?
    • Answer: \((K-1)\varepsilon\)

• Regret Reduction:
  – Each \( \hat{h} \) achieves 0/1 regret \( r \)
  – Implication of multiclass regret?
    • E.g., \( Kr \)?
  – More powerful result

\[
\varepsilon = L_p(w)
\]

Zero 0/1 Test Error typically not possible

\[
r = L_p(w) - L_p(w^*)
\]
Filter Tree Regret Guarantee

• If each classifier has regret $r$
• Filter Tree has multiclass regret $\leq (\log_2 K)r$
  – Good dependence on $K$
• Inductive proof
• See details in paper

http://mi.eng.cam.ac.uk/~mjfg/local/Projects/filter_tree.pdf
Runtime Computational Benefits

• Logarithmic test time
  – With respect to #classes

See also: Logarithmic Time Online Multiclass Prediction
Very Briefly: Sequential ➔ Multiclass

- Suppose we want to use decision trees for first-order sequence prediction
Recurrent Multiclass Classifier

- \( h(x, y_{prev}) \)
  - Takes in current \( x \), previous \( y \)
  - Predicts next \( y \)

http://www.umiacs.umd.edu/~hal/searn/
http://arxiv.org/abs/1011.0686
Next Week

• No Lecture Thursday
  – Student Faculty Conference

• Recitation Thursday
  – Conditional Random Fields Review

• Kaggle Miniproject Writeup due Thursday
  – Via Moodle

• Next Week:
  – Unsupervised, Clustering, Dim. Reduction