Machine Learning & Data Mining
CS/CNS/EE 155

Lecture 9:
Recent Applications
Today

• Recent Applications:

  Edge Detection

  Speech Animation

• Introduction to Learning Reductions
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.
Challenges

• Output Space?

• 400x300 Image
  – 120000 Pixels
  – $2^{120000}$ Labels!
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

- Can ignore most of image
Sliding Window Approach  
(Decomposition)

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

• Slide across image

• What model?
Recall: Binary Decision Tree

Input: Alice
Gender: Female
Age: 14

Prediction: Height > 55”

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

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

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?
Structured Information Gain

“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

(Actual approach more complicated.)

“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...

(Actual approach more complicated.)

“For 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
• 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 $\mathcal{S} = \{(x,\hat{y})\}$
  
  – $x = 16x16$ image patch
  
  – $\hat{y} = 16x16$ ground truth edges

• Train structured DT on $\mathcal{S}$

• 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 $$Y = \langle y_1, y_2, \ldots, y_{|y|} \rangle$$, $$y_t \in \mathbb{R}^D$$

Goal: learn predictor $$h : X \rightarrow Y$$

<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>i</td>
<td>h</td>
<td>i</td>
<td>h</td>
<td>d</td>
<td>d</td>
<td>i</td>
<td>h</td>
<td>i</td>
<td>h</td>
<td>i</td>
<td>h</td>
<td>k</td>
<td>k</td>
<td>sh</td>
<td>sh</td>
<td>sh</td>
<td>sh</td>
</tr>
</tbody>
</table>

Sequence of face configurations

Phoneme sequence
Temporal curvature can vary smoothly or sharply (depends on context – this is the co-articulation problem)

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

 FRAME | 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
 Token | p p r ih ih d d ih ih ih ih k k sh sh sh sh uh uh n

X

Y

Dimension 1

Frame number

2 4 6 8 10 12 14 16 18 20 22 24
Co-Articulation is Hard to Get Right

(Strong Local Properties)

/k/

/t/
Weak Global Properties

• No need to model entire chain directly

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

• Motivates sliding window approach!
Input speech: “P R E D I C T I O N”

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

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

Full tree has 5K+ leaf nodes
Multivariate Regression Tree

• Prediction:

• Training loss: multivariate squared loss:

\[ \sum_{\text{Leaf}} \sum_{\hat{y} \in \text{Leaf}} \left\| \hat{y}_{\text{Leaf}} - \hat{y} \right\|^2 \]
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) x Token - l l l er er er n n n iy iy ng ng ng g g g g -

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

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

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

(e) y

Dimension 1

Frame number

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) x Token - l l l er er er n n n iy iy ng ng ng g g g g -

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

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

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

(e) y

Dimension 1

Frame number
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 s ih ih ih g g g r r ae ae ae ae f f f f -

(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) $\mathbf{y}$

![Graph showing the relationship between input speech and predicted outputs]
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
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
Is the phoneme in the 8th frame a diphthong? Y
Is the phoneme in the 8th frame a semivowel? Y Y N N
Is the phoneme in the 3rd frame articulated at the back of the mouth?

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 ➔ Binary
• Cost-weighted ➔ Unweighted
• Ranking ➔ Binary
• Sequential ➔ Multiclass
• And many more...

http://hunch.net/~jl/projects/reductions/reductions.html
Other Learning Reductions

- Multiclass ➔ Binary
- Cost-weighted ➔ Unweighted
- Ranking ➔ Binary
- Sequential ➔ 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

\[
r = L_P(w) - L_P(w^*)
\]
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

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

First Layer

Each base model is a binary classifier
The Learning Reduction

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

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

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
Next Week

• Probabilistic Models

• Hidden Markov Models

• **Tonight:** Recitation on Probability & Statistics