Multisensory Perception

Alex Teichman

August 26, 2009
The Object Recognition Problem

How much does this look like a hand?
The Object Recognition Problem

Where is the coke can?
The Object Recognition Problem

What is the object in the box?
Some Difficulties Among Many

- Labels are painful to acquire.
Some Difficulties Among Many

- Labels are painful to acquire.
- Generalization to data not in your training set can be arbitrarily hard.
Some Difficulties Among Many

- Labels are painful to acquire.
- Generalization to data not in your training set can be arbitrarily hard.
- Descriptors are generally suited to a particular task.
Boosting - Intuition

- Person +0.3
- Image Descriptors +0.8
- Applications -0.5
- Moving Forward · · ·
Boosting - Intuition

<table>
<thead>
<tr>
<th></th>
<th>Person</th>
<th>Hand</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>+0.3</td>
<td>+0.3</td>
</tr>
<tr>
<td></td>
<td>+0.8</td>
<td>-0.7</td>
</tr>
<tr>
<td></td>
<td>-0.5</td>
<td>-0.5</td>
</tr>
</tbody>
</table>
Training examples \((y_m, \vec{x}_m)\) for all \(m = 1..M\), \(y_m \in \{-1, +1\}\).
Boosting - Math

Training examples \((y_m, \bar{x}_m)\) for all \(m = 1..M\), \(y_m \in \{-1, +1\}\).

\[
\min_H \sum_m \exp(-y_m H(\bar{x}_m))
\]  \hspace{1cm} (1)
Training examples \((y_m, \vec{x}_m)\) for all \(m = 1..M\), \(y_m \in \{-1, +1\}\).

\[
\min_H \sum_m \exp(-y_m H(\vec{x}_m)) \tag{1}
\]

\[
H(\vec{x}) = \sum_t h_t(\vec{x}) \tag{2}
\]
Training examples \((y_m, \vec{x}_m)\) for all \(m = 1..M\), \(y_m \in \{-1, +1\}\).

\[
\min_H \sum_m \exp(-y_m H(\vec{x}_m)) \tag{1}
\]

\[
H(\vec{x}) = \sum_t h_t(\vec{x}) \tag{2}
\]

\[
h_t(\vec{x}) = \begin{cases} 
a_t & \text{if dist}(\vec{x}, \vec{x}_t) < \theta_t \\
0 & \text{otherwise}
\end{cases} \tag{3}
\]

Stagewise optimization:
\[
\min_{h_{t+1}} \sum_m \exp(-y_m (H(\vec{x}_m) + h_{t+1}(\vec{x}_m)))
\]
Goal: Plug-and-play image descriptors.
Multisensory Perception
Alex Teichman
The Object Recognition Problem
Boosting
Image Descriptors Library
Applications
Moving Forward

See descriptors_2d package
Hand Detection

Multisensory Perception
Alex Teichman

The Object Recognition Problem
Boosting
Image Descriptors Library
Applications
Moving Forward

Hand Detection

- Histogram of Oriented Gradient descriptors (3 different parameter settings)
- Superpixel Color Histogram descriptors (4 different superpixel sizes)

1 hour to train.
Collaboration with Romain and Alex S.
Hand Detection

- Histogram of Oriented Gradient descriptors (3 different parameter settings)
Hand Detection

- Histogram of Oriented Gradient descriptors (3 different parameter settings)
- Superpixel Color Histogram descriptors (4 different superpixel sizes)
Hand Detection

- Histogram of Oriented Gradient descriptors (3 different parameter settings)
- Superpixel Color Histogram descriptors (4 different superpixel sizes)
- 1 hour to train.

Collaboration with Romain and Alex S.
Label a video given just a few frame labels.
Implementing the feedback loop of the hands training methodology on mechanical turk with a bottles dataset.
Online Learning

\[ \min_a \sum_m \exp(-y_m H(x_m)) \]

- Can solve this for all weak classifier responses simultaneously.
- Constant space for any number of training examples.

Maintain the gradient and Hessian as a stream of labels comes into the classifier:

\[ g = \frac{1}{M} \sum_m -y_m I_m \]

\[ H = \frac{1}{M} \sum_m I_m I_m^T \]
Labels are painful to acquire.
Moving Forward

- Labels are painful to acquire.
  - Put classifier predictions on Mechanical Turk, get corrections, feed back into boosting.
Moving Forward

- Labels are painful to acquire.
  - Put classifier predictions on Mechanical Turk, get corrections, feed back into boosting.

- Generalization to data not in your training set can be arbitrarily hard.
  - Put classifier predictions on Mechanical Turk, get corrections, feed back into boosting.
Moving Forward

- Labels are painful to acquire.
  - Put classifier predictions on Mechanical Turk, get corrections, feed back into boosting.

- Generalization to data not in your training set can be arbitrarily hard.
  - Put classifier predictions on Mechanical Turk, get corrections, feed back into boosting.

- Descriptors are generally suited to a particular task.
Moving Forward

- Labels are painful to acquire.
  - Put classifier predictions on Mechanical Turk, get corrections, feed back into boosting.

- Generalization to data not in your training set can be arbitrarily hard.
  - Put classifier predictions on Mechanical Turk, get corrections, feed back into boosting.

- Descriptors are generally suited to a particular task.
  - Use a large library of descriptors and allow boosting to choose which are most useful through the optimization.
Thanks!