Boosting Nearest-Neighbor Classifier for Character Recognition

Li Zhang and Lin Liao
May 17, 2004

Outline
- Data set
- Nearest-Neighbor classifier
- Boosting algorithm
- Results

Data set -- MNIST database
- grayscale handwritten digits
- All digits are normalized and centered in a 28x28 image
- Training set: 60,000 examples
- Test set: 10,000 examples

Nearest-Neighbor Classifier

<table>
<thead>
<tr>
<th>Training Set</th>
<th>Test Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 4 5 7 8 9 6 4 1</td>
<td>9 2 3 6 5 3 1 9 7</td>
</tr>
<tr>
<td>6 7 5 8 6 3 4 8 5</td>
<td>2 1 7 9 0 8 3 7 6</td>
</tr>
<tr>
<td>4 8 1 0 3 8 9 4</td>
<td>5 6 1 8 6 1 5 6 0</td>
</tr>
<tr>
<td>7 5 9 2 6 5 3 1 9 7</td>
<td>2 2 3 2 3 4 8 0</td>
</tr>
<tr>
<td>4 2 8 0 7 8 3 5 7</td>
<td>0 1 4 6 4 6 0 2 8 3</td>
</tr>
<tr>
<td>7 1 2 8 0 6 3 8 6 1</td>
<td>5</td>
</tr>
</tbody>
</table>

Motivation of Boosting
- Nearest-Neighbor Classifier (NN)
  - Need large training set to get good accuracy
  - Classification is slow
- Boosted NN:
  - Speed up classification by reducing the examples used during classification
  - Do not increase the error rate

Boosting
- Maintain vector of weights for training samples
- At each iteration
  - Learn a weak classifier using weighted samples
  - Increase the weights of misclassified examples
- Combine weak classifiers by weighted voting
Boosting (cont’d)

- Boost 2-class classifier
  - AdaBoost.M1
  - Boost multi-class classifier
  - AdaBoost.M2

(image, label) → [0,1]

AdaBoost.M1

Input: sequence of m examples \( \langle (x_1,y_1), \ldots, (x_m,y_m) \rangle \) with labels \( y_i \in Y = \{1, \ldots, k\} \) weak learning algorithm \( \text{WeakLearn} \) integer \( T \) specifying number of iterations

Initialize \( D_1(i) = 1/m \) for all \( i \).

Do for \( t = 1, 2, \ldots, T \):
1. Call \( \text{WeakLearn} \), providing it with the distribution \( D_t \).
2. Get back a hypothesis \( h_t : X \rightarrow Y \).
3. Calculate the error of \( h_t \):
   \[
   \epsilon_t = \sum_{i:D_t(i)\neq y_i} D_t(i).
   \]
   If \( \epsilon_t > \frac{1}{2} \), then set \( T = t - 1 \) and abort loop.

AdaBoost.M2

Input: sequence of m examples \( \langle (x_1,y_1), \ldots, (x_m,y_m) \rangle \) with labels \( y_i \in Y = \{1, \ldots, k\} \) weak learning algorithm \( \text{WeakLearn} \) integer \( T \) specifying number of iterations

Let \( B = \{ (i,y) : i = 1, \ldots, m, y \neq y_i \} \).

Initialize \( D_t(i,y) = \frac{1}{|B|} \) for \( (i,y) \in B \).

Do for \( t = 1, 2, \ldots, T \):
1. Call \( \text{WeakLearn} \), providing it with mislabel distribution \( D_t \).
2. Get back a hypothesis \( h_t : X \rightarrow Y \rightarrow [0,1] \).
3. Calculate the pseudo-loss of \( h_t \):
   \[
   \epsilon_t = \sum_{(i,y)\in B} D_t(i,y) \left( 1 - h_t(x_i,y) + h_t(x_i,y_i) \right).
   \]

4. Set \( \beta_t = \epsilon_t / (1 - \epsilon_t) \).
5. Update \( D_{t+1} \):
   \[
   D_{t+1}(i,y) = \frac{D_t(i,y)}{\beta_t} \cdot g(z_t(x_i,y)) \cdot h_t(x_i,y_i) \cdot \text{norm}(z_t).
   \]
   where \( z_t \) is a normalization constant (chosen so that \( D_{t+1} \) will be a distribution).

Output the final hypothesis:
\[
\hat{h}_T(x) = \arg\max_{y \in Y} \sum_{t=1}^{T} \log \frac{1}{\beta_t} h_t(x,y).
\]

Weak Classifier of NN

- Given the distribution on pairs \((X,Y)\), find a subset \( P \) of the training examples and a mapping \( n : (P,Y) \rightarrow [0,1] \) that achieve small "pseudo-loss." (\(|P|=100\))
  1. Initially \( P \) is empty
  2. Sample 10 candidates from the distribution \((X,Y)\)
  3. Pick 1 candidate from the 10 that minimize the pseudo-loss if it is added to \( P \)
  4. Repeat 2, 3 until \(|P|=100\)
  5. Find the best mapping \( n \) given \( P \)
  6. Return the \( P \), \( n \) and the final pseudo-loss
## Conclusion

- **Advantages of boosting**
  - The *training* and *test* error rates are both theoretically bounded
  - Less “overfitting” in practice
  - Many algorithms can be boosted
  - Easy to implement

- **Disadvantages of boosting**
  - Learning is slow