Optical Character Recognition:
Machine learning of handwritten characters

Bryant Mairs
Dec. 8, 2009
Premise

- Explore various methods for performing optical character recognition using machine learning algorithms

- Baseline: Pixel features with linear regression

- Current: Adaboost-chosen Haar features with linear regression

- Idea: Probability-chosen Haar features with linear regression
Dataset

- Handwritten numbers 0 through 9 as 32x32 binary images
  - From Bogazici University
- Handwritten lowercase English letters as 15x7 binary images
  - From MIT's Spoken Languages Systems Group
Data formatting

- Heterogenous prescaling (zoom-to-fit)
- Nearest-neighbor interpolation
Pixel Features

0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 ...
Haar Features

- Demonstrated by Viola & Jones 2001 for facial recognition
- Feature is exhibited if intensity under black area is less than under grey area
- Too many to use all (>1M at 32x32), also over-complete set
Adaboost-chosen features

- Use Haar features as learners
- Run Adaboost to select most effective features

Given example images \((x_1, y_1), \ldots, (x_n, y_n)\) where \(y_i = 0, 1\) for negative and positive examples respectively.

Initialize weights \(w_{1,i} = \frac{1}{2m}, \frac{1}{2l}\) for \(y_i = 0, 1\) respectively, where \(m\) and \(l\) are the number of negatives and positives respectively.

For \(t = 1, \ldots, T:\)

1. Normalize the weights,
   \[w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{m} w_{t,j}}\]
   so that \(w_t\) is a probability distribution.

2. For each feature, \(j\), train a classifier \(h_j\) which is restricted to using a single feature. The error is evaluated with respect to \(w_t\), \(e_j = \sum_i w_i |h_j(x_i) - y_i|\).

3. Choose the classifier, \(h_t\), with the lowest error \(e_t\).

4. Update the weights:
   \[w_{t+1,i} = w_{t,i} \beta_t^{1 - e_i}\]
   where \(e_i = 0\) if example \(x_i\) is classified correctly, \(e_i = 1\) otherwise, and \(\beta_t = \frac{s}{1 - e_t}\).

The final strong classifier is:

\[
h(x) = \begin{cases} 
1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\
0 & \text{otherwise}
\end{cases}
\]

where \(\alpha_t = \log \frac{1}{\beta_t}\)
Probability-chosen Features

- Simplify the feature filtering process by using pixel probabilities
- Select discriminating features that are only exhibited in 2/3 image classes
- Weight classes inverse to features already affecting class
Multiclass Problems

- Both Logistic Regression and Adaboost models solve multi-class problems
- Slight variation on 2-class problems, reduces to original algorithm when classes=2
Logistic Regression

- Eta = .001, lambda = .01, fit to .01
Adaboost

- Used SAMME algorithm by Zhu et al.
- Adds constant to weight update $\alpha$ dependent on number of added classes
Results

Percentage identified correctly by class

- Pixels
- Haar
Future

- Adaboost-selected features couldn't complete in a way to make a relevant comparison
- Pixel-features worked best even without negated features
- Plenty of room for improvement
  - Improve probability-selection algorithm to displace weaker features once class-feature cap hit
  - Remove features by comparison to character-set probability map in order to reduce Adaboost-computation time