Handwritten Digit Classification

Ahmed Bou-Rabee    Stephen Boyd

EE103
Stanford University

August 24, 2014
Outline

Classification

$k$-means

Binary (two-way) classification

10-way classification

Classification with random features
Handwritten digit classification

- goal is to automatically determine what a handwritten digit image is (i.e., 0, 1, ..., 8, or 9?)

![Handwritten digit images](images/handwritten_digits.png)
images are $16 \times 16$ pixels, represented as 256-vectors
values in $[0, 1]$ (0 is black, 1 is white)
images were first de-slanted and size normalized
our classifier is a function $f : \mathbb{R}^{256} \rightarrow \{0, 1, \ldots, 9\}$
our guess is $\hat{y} = f(x)$ for image $x$
our classifier is wrong when $\hat{y} \neq y$
Data set

- NIST data from US Postal Service
- training set has $N = 7291$ images
  - we’ll use this data set to develop our classifiers
- test set has $N^{\text{test}} = 2007$ images
  - we’ll use this data set to test/judge our classifiers
- we’ll look at error on training set and on test set
Outline

Classification

$k$-means

Binary (two-way) classification

10-way classification

Classification with random features
$k$-means

- start with a collection of image 256-vectors $x_1, \ldots, x_N$
- run $k$-means algorithm to cluster into $k$ groups, 10 times with random initial centroids
- use best of these 10 (in mean-square distance to closest centroid)
- centroids/representatives $z_1, \ldots, z_k$ can be viewed as images
Centroids, $k = 2$
Centroids, $k = 10$
Centroids, $k = 20$
Classification via $k$-means

- label $k = 20$ centroids by hand
- classify new image by label of nearest centroid
- classification error rate (on test set): 24%
Classification via $k$-means

confusion matrix:
true ↓ predicted →

<table>
<thead>
<tr>
<th></th>
<th>0</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>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>338</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>253</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>1</td>
<td>131</td>
<td>10</td>
<td>29</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>143</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>103</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>81</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>8</td>
<td>78</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>154</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>113</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>16</td>
<td>10</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>107</td>
<td>13</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>1</td>
<td>0</td>
<td>43</td>
<td>3</td>
<td>110</td>
</tr>
</tbody>
</table>
Outline

Classification

$k$-means

Binary (two-way) classification

10-way classification

Classification with random features
Binary classifier

- a simpler problem: determine if an image $x$ is digit $k$ or not digit $k$
- we use label $y_i = 1$ if $x_i$ is digit $k$ and $y_i = -1$ if not
- classifier will have form

\[
\hat{y} = \text{sign}(w^T x + v)
\]

$w$ is weight 256-vector, $v$ is offset
- we’ll use training set to choose $w$ and $v$, and test the classifier on test data set
Least-squares binary classifier

- want $w, v$ for which $y_i \approx \hat{y}_i = \text{sign}(w^T x_i + v) = \text{sign}(\tilde{y}_i)$
- choose $w, v$ to minimize

$$
\sum_{i=1}^{N} (\tilde{y}_i - y_i)^2 + \lambda \|w\|^2 = \|X^T w + v1 - y\|^2 + \lambda \|w\|^2
$$

- $X = [x_1 \cdots x_N]$ is matrix of training image vectors
- $\lambda > 0$ is regularization parameter
least-squares binary classifier

classification error versus \( \lambda \) for predicting the digit 0

Binary (two-way) classification
Weight vector

Binary (two-way) classification
Outline

Classification

$k$-means

Binary (two-way) classification

10-way classification

Classification with random features
10-way classification

- let \( w_i, v_i \) be weight vector, offset for binary classification of digit \( i \)
- for image \( x \), \( \tilde{y}_i = w_i^T x + v_i \)
- the larger \( \tilde{y}_i \) is, the more confident we are that image is digit \( i \)
- choose \( \hat{y} = \arg\max_i(\tilde{y}_i) = \arg\max_i(w_i^T x + v_i) \)
- use the same regularization parameter \( \lambda \) for each digit \( i \)
- choose \( \lambda \) so that the total classification error on test set is small
Example

multi-class classification error versus $\lambda$

with $\lambda = 50$, test classification error is about 13%
Example

test confusion matrix
true ↓ predicted →

<table>
<thead>
<tr>
<th></th>
<th>0</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>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>348</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>256</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>3</td>
<td>160</td>
<td>7</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>140</td>
<td>2</td>
<td>8</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>173</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>20</td>
<td>2</td>
<td>120</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>151</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>131</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>3</td>
<td>2</td>
<td>14</td>
<td>4</td>
<td>7</td>
<td>1</td>
<td>2</td>
<td>119</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>2</td>
<td>157</td>
</tr>
</tbody>
</table>

10-way classification
Outline

Classification

\(k\)-means

Binary (two-way) classification

10-way classification

Classification with random features
Doing even better

- in classes you’ll take later (AI, statistics), you’ll see (and construct) way better classifiers
- we’ll look at a simple example here
Generating random features

- generate a random $2000 \times 256$ matrix $R$ with entries $+1$ or $-1$
- scale $R$ by $1/\sqrt{256}$, so each row has norm 1
- create 2000 new features $\tilde{x}$ from original $x$ via

$$\tilde{x}_i = \max\{Rx, 0\}$$

- now do least-squares classification with feature 2256-vectors $(x_i, \tilde{x}_i)$
Example

multi-class classification error versus $\lambda$

with $\lambda = 1$, test classification error is about 5%
**Example**

Test confusion matrix

<table>
<thead>
<tr>
<th></th>
<th>0</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>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>352</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>256</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>187</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>150</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>188</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>149</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>161</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>138</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>154</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>171</td>
</tr>
</tbody>
</table>

Classification with random features