Handwritten Digit Classification

Ahmed Bou-Rabee   Stephen Boyd

EE103
Stanford University

August 24, 2014

Outline

Classification

k-means

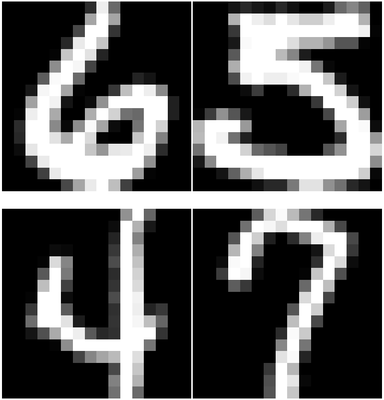
Binary (two-way) classification

10-way classification

Classification with random features

Handwritten digit classification

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



Classifier

- images are 16×16 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 $f : \mathbf{R}^{256} \to \{0, 1, \ldots, 9\}$
- our guess is $\hat{y} = f(x)$ for image x
- our classifier is wrong when $\hat{y} \neq y$

Data set

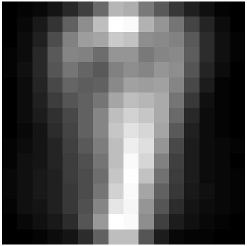
- NIST data from US Postal Service
- training set has $N = 7291$ images
  - we'll use this data set to develop our classifiers
- test set has $N^{\text{test}} = 2007$ images
  - we'll use this data set to test/judge our classifiers
- we'll look at error on training set and on test set

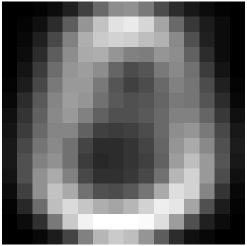
Outline

k-means

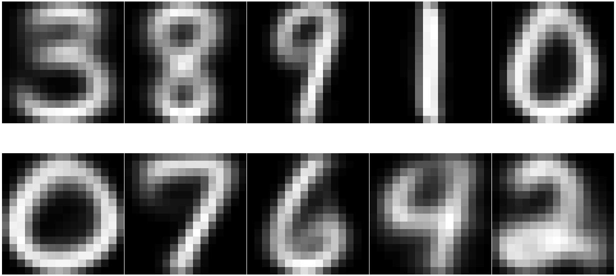
- start with a collection of image 256-vectors $x_1, \ldots, x_N$
- run k-means algorithm to cluster into k groups, 10 times with random initial centroids
- use best of these 10 (in mean-square distance to closest centroid)
- centroids/representatives $z_1, \ldots, z_k$ can be viewed as images

Centroids, $k = 2$

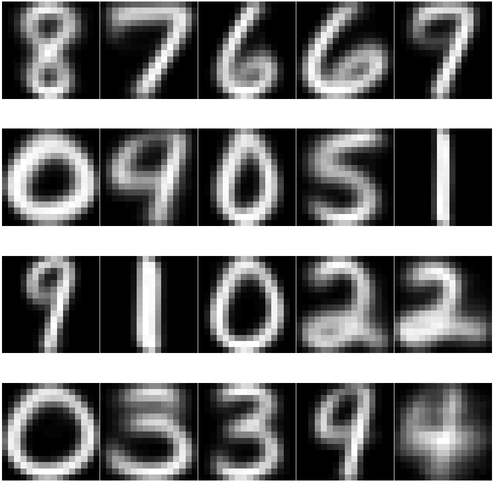




Centroids, $k = 10$



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 $\downarrow$ predicted $\rightarrow$

	0	1	2	3	4	5	6	7	8	9
0	338	0	2	3	6	0	9	0	0	1
1	0	253	0	1	4	0	2	0	0	4
2	7	1	131	10	29	1	3	2	13	1
3	4	0	1	143	3	6	1	1	6	1
4	1	4	4	0	103	0	1	4	2	81
5	10	0	0	50	8	78	7	0	0	7
6	6	0	2	0	4	2	154	0	1	1
7	0	3	0	0	6	0	0	113	1	24
8	5	2	5	16	10	7	0	1	107	13
9	0	2	0	0	18	1	0	43	3	110

Outline

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

$$\hat{y} = \mathbf{sign}(w^T x + v)$$

  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

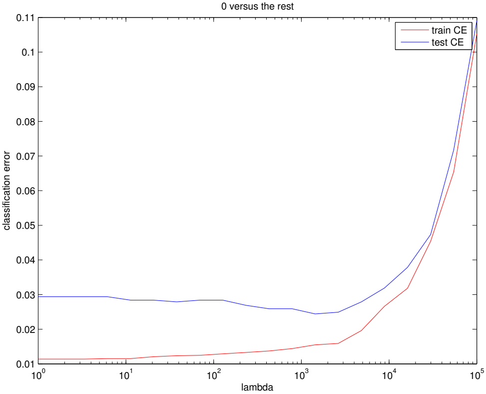
- want w, v for which $y_i \approx \hat{y}_i = \mathbf{sign}(w^T x_i + v) = \mathbf{sign}(\tilde{y}_i)$
- choose w, v to minimize

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

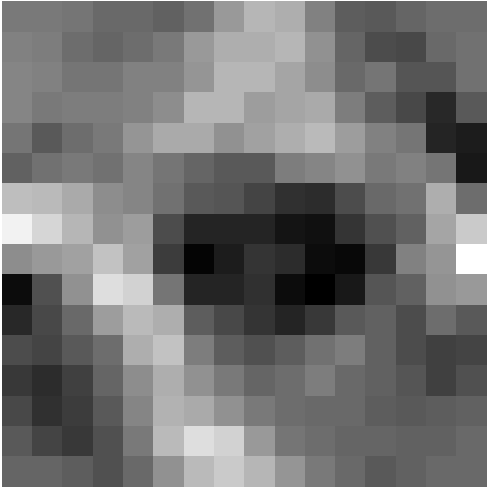
- $X = [x_1 \cdots x_N]$ is matrix of training image vectors
- $\lambda > 0$ is regularization parameter

Least-squares binary classifier

classification error versus λ for predicting the digit 0



Weight vector



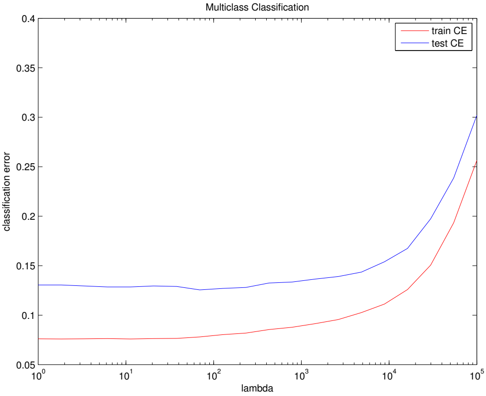
Outline

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 $\hat{y} = \mathrm{argmax}_i(\tilde{y}_i) = \mathrm{argmax}_i(w_i^T x + v_i)$
- use the same regularization parameter λ for each digit i
- choose λ so that the total classification error *on test set* is small

Example

multi-class classification error versus λ



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

Example

test confusion matrix

true ↓ predicted →

	0	1	2	3	4	5	6	7	8	9
0	348	2	0	1	3	1	3	0	0	1
1	0	256	0	2	3	0	1	0	1	1
2	8	3	160	7	9	1	1	1	8	0
3	5	0	3	140	2	8	0	2	3	3
4	3	6	4	0	173	0	3	1	0	10
5	10	1	0	20	2	120	0	1	1	5
6	3	1	4	0	5	5	151	0	1	0
7	2	1	1	1	6	0	0	131	0	5
8	10	3	2	14	4	7	1	2	119	4
9	0	3	0	1	7	0	0	7	2	157

Outline

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

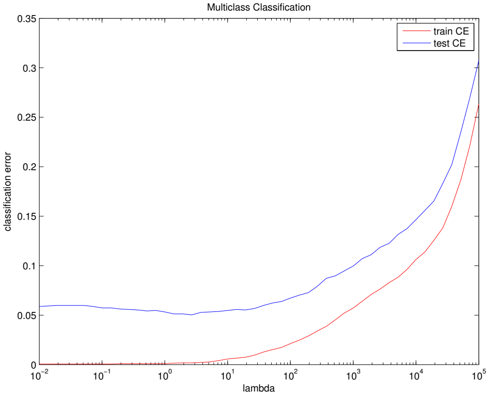
- generate a random 2000×256 matrix R with entries $+1$ or -1
- scale R by $1/\sqrt{256}$, so each row has norm 1
- create 2000 new features $\tilde{x}$ from original x via

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

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

Example

multi-class classification error versus λ



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

Example

test confusion matrix

true $\downarrow$ predicted $\rightarrow$

	0	1	2	3	4	5	6	7	8	9
0	352	0	3	0	2	0	1	0	0	1
1	0	256	0	0	4	0	3	1	0	0
2	1	0	187	3	2	0	0	1	4	0
3	1	0	4	150	0	7	0	0	3	1
4	0	1	3	0	188	0	1	0	1	6
5	2	0	0	3	1	149	0	0	1	4
6	3	0	3	0	2	1	161	0	0	0
7	0	0	1	0	6	0	0	138	1	1
8	3	0	3	3	0	1	2	0	154	0
9	0	0	0	0	3	1	0	1	1	171