DSC 255 - MACHINE LEARNING FUNDAMENTALS

NEAREST NEIGHBOR CLASSIFICATION

SANJOY DASGUPTA, PROFESSOR



COMPUTER SCIENCE & ENGINEERING
HALICIOĞLU DATA SCIENCE INSTITUTE



The problem we'll solve today

Given an image of a handwritten digit, say which digit it is.



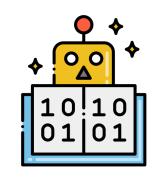
More examples



The machine learning approach

1. Assemble a data set

```
1416119134857268U32264141
8663597202992997225100467
0130844145910106154061036
3(10641110304752620099799
6684120%67285571314279554
6060177501871129930899709
8401097075973319720155190
5510755182551828143580909
```



The MNIST data set of handwritten digits:

- **Training set** of 60,000 images and their labels.
- **Test set** of 10,000 images and their labels.

2. let the machine figure out the underlying patterns.

Nearest neighbor classification

Labels

```
Training images x^{(1)}, x^{(2)}, x^{(3)}, ..., x^{(60000)}
                         y^{(1)}, y^{(2)}, y^{(3)}, ..., y^{(60000)} are numbers in the range 0 - 9
```

```
1416119134857268432264141
5518255108503047520439401
```



How to **classify** a new image x?

- Find its nearest neighbor amongst the $x^{(i)}$
- Return $v^{(i)}$

The Data Space

How to measure the distance between images?



MNIST images

- Size 28 X 28 (total: 784 pixels)
- Each pixel is grayscale: 0-255

Stretch each image into a vector with 784 coordinates:

- Data space $\chi = \mathbb{R}^{784}$
- Label space $\gamma = \{0, 1, ..., 9\}$

The Distance Function

Remember Euclidean distance in two dimensions?

$$z = (3, 5)$$

$$x = (1, 2)$$

Euclidean distance in higher dimension

Euclidean distance between 784-dimensional vectors x, z is

$$\| x - z \| = 1 \sum_{i=1}^{784} (x_i - z_i)^2$$

Here x_i is the *i*th coordinate of x.

Nearest neighbor classification

```
1416119134857868U32264141
86635972029929977225100467
0130844145910106154061036
31106411103047526200997999
6689120867885571314279554
60101775018711299108997709
8401097075973319720155190
5510755182551825518260439401
```

```
Training images x^{(1)}, ..., x^{(60000)}
Labels y^{(1)}, ..., y^{(60000)}
```



To classify a new image x:

- Find its nearest neighbor amongst the $x^{(i)}$ using Euclidean distance in \mathbb{R}^{784}
- Return $y^{(i)}$

How accurate is this classifier?

Training set of 60,000 points

What is the error rate on training points?

Training set of 60,000 points

What is the error rate on training points? Zero In general, training error is an overly optimistic predictor of future performance.

- What is the error rate on training points? Zero In general, training error is an overly optimistic predictor of future performance.
- A better gauge: separate test set of 10,000 points.
 Test error = fraction of test points incorrectly classified.

- What is the error rate on training points? Zero In general, training error is an overly optimistic predictor of future performance.
- A better gauge: separate test set of 10,000 points.
 Test error = fraction of test points incorrectly classified.
- What test error would we expect for a random classifier?
 (One that picks a label 0 9 at random?)

- What is the error rate on training points? Zero In general, training error is an overly optimistic predictor of future performance.
- A better gauge: separate test set of 10,000 points.
 Test error = fraction of test points incorrectly classified.
- What test error would we expect for a random classifier?
 (One that picks a label 0 9 at random?) 90%

- What is the error rate on training points? Zero In general, training error is an overly optimistic predictor of future performance.
- A better gauge: separate test set of 10,000 points.
 Test error = fraction of test points incorrectly classified.
- What test error would we expect for a random classifier?
 (One that picks a label 0 9 at random?) 90%
- Test error of nearest neighbor: 3.09%