k-nearest neighbors

(kNN)

The problem we'll solve today

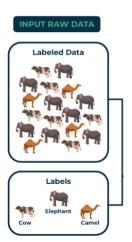
Given a 28x28 image, guess which digit it is



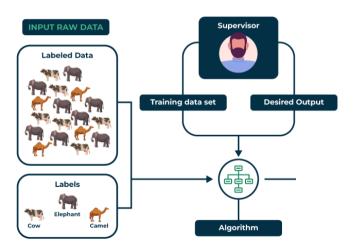
Some more examples:



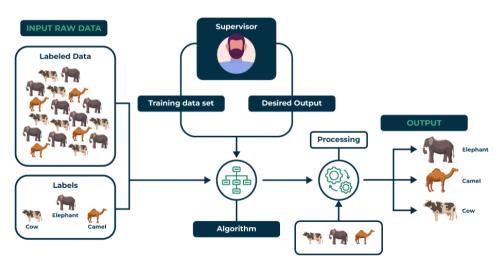
Supervised learning



Supervised learning



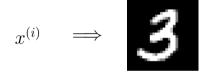
Supervised learning



Common supervised learning models

- · Logistic regression
- k-nearest neighbor (k-NN) ← Today
- Naïve Bayes
- Decision tree
- Random forest
- Support vector machine (SVM)

Notations: Data point



 $y^{(i)} \implies 3$

MNIST dataset

```
1416119134857268432264141
8443597202992997225100467
0130841145910106154061036
3110441110304752620011799
6689120867285571314279554
6010187801871129910899709
8401097075973319720155190
3510755182551828143580909
4917875X16554445544035460
5518255108503067520439401
```

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

Nearest neighbor classification

- Training images $x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(60000)}$
- Labels $y^{(1)}, y^{(2)}, y^{(3)}, \dots, y^{(60000)}$ are numbers from 0-9

```
1416119134857268U32264141
8663597202992997225100467
0130844145910106154061036
3110641110304752620099799
6689120%67285571314279554
6060177301871129930899709
8401097075973319720155190
5510755182551828143580909
```

Nearest neighbor classification

- Training images $x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(60000)}$
- Labels $y^{(1)}, y^{(2)}, y^{(3)}, \dots, y^{(60000)}$ are numbers from 0-9

```
416119134857268032264141
8443597202992997225100467
0130841115910106154061036
3110641110304752620011799
668412B847885541314274554
6010122301871129910899709
8401097075973319720155190
3510755182551828143880909
6317875X16554665546035460
5518255108503067520439401
```

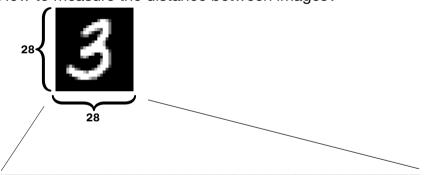


How to classify a new image x?

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

Data as vectors

How to measure the distance between images?



Stretch each image into a vector with 784 coordinates

$$x^{(1)} = (0, 0, 0, \dots, 0.6, 1, 1, 1, 0, 0, 1, 1, 0.8, \dots, 0, 0, 0)$$

 $v^{(1)} = 6$

The distance function

Euclidean distance in two dimensions is

Euclidean distance in higher dimension

Two images a and b:

$$a = (a_1, a_2, a_3, \dots, a_{784})$$

 $b = (b_1, b_2, b_3, \dots, b_{784})$

The Euclidean distance between a and b is

$$||a - b||_2 = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_{784} - b_{784})^2}$$
$$= \sqrt{\sum_{i=1}^{784} (a_i - b_i)^2}$$

Nearest neighbor classification

Training images $x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(60000)}$ Labels $y^{(1)}, y^{(2)}, y^{(3)}, \dots, y^{(60000)}$

```
1416119134857868U32264141
8663597202992997225100467
0130844145910106154061036
3110641110304752620097799
6689120867885571214279554
6010177501871189910899709
8401097075973319720155190
5610755182551828143580909
6317875416554605546035460
```

Nearest neighbor classification

Training images $x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(60000)}$ Labels $y^{(1)}, y^{(2)}, y^{(3)}, \dots, y^{(60000)}$

```
1416119134857268U322641418663S9720299299722510046701308441459101061540610363100641110304752620917799668912084111030475262091799668912089711299308997098401097075973319720155190561075518255182551825143580109631787546554603546035460
```

To classify a new image x



- Find its nearest neighbor in Euclidean distance, say $x^{(i)}$
- Return $v^{(i)}$

Accuracy of Nearest Neightbor on MNIST

Predictions on all points in the **Training set**

Question: What is the accuracy?

Accuracy of Nearest Neightbor on MNIST

```
1416119134857268 U32264141
8663597202992997225100467
0130844145910106154061036
3(10641110304752620071799
6689120%47285571214279554
60101723018711299108997709
8401097075973319720155190
651075518255(828143580109
6317875416554605546035460
```

Predictions on all points in the Test set

Question: What is the accuracy?

Test accuracy and Test error

Test set of 10,000 points

· 309 are misclassified

Test accuracy =
$$\frac{\# \text{ correct classification}}{\# \text{ all points}}$$

Test error =
$$\frac{\# \text{ incorrect classification}}{\# \text{ all points}}$$

Examples of errors

Test set of 10,000 points

• 309 are misclassified

Examples of errors:

Test image

4

0

8

מ

•

9

How to improve?

Nearest neighbor

k-nearest neighbor classification

To classify a new point:

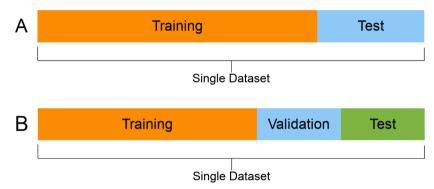
- Find the *k* nearest neighbors in the training set
- Return the most common label amongst them

MNIST:

| k | 1 | 3 | 5 | 7 | 9 | 11 |
|----------------|------|------|------|------|------|------|
| Test error (%) | 3.09 | 2.94 | 3.13 | 3.10 | 3.43 | 3.34 |

need to find k before final eval on the test set

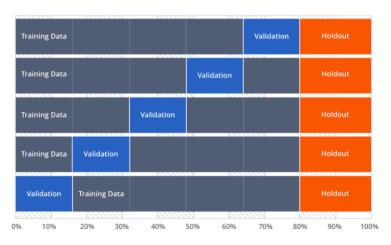
Validation



Train on Training set with $k=1\Longrightarrow$ Evaluate on the Validation set Train on Training set with $k=3\Longrightarrow$ Evaluate on the Validation set Train on Training set with $k=5\Longrightarrow$ Evaluate on the Validation set

:

Cross-validation



Other distance function

$$a = (a_1, a_2, \dots, a_m)$$
 $b = (b_1, b_2, \dots, b_m).$

Cosine similarity

$$d_{\cos}(a,b) = \frac{a \cdot b}{\|a\|_2 \|b\|_2} = \frac{a}{\|a\|_2} \cdot \frac{b}{\|b\|_2}$$

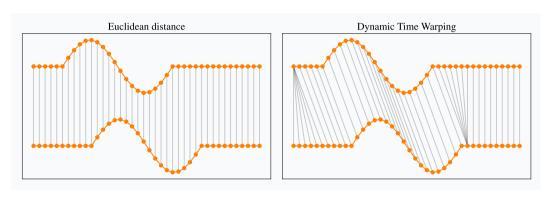
measures the angle between vector a and b.

$$-1 \le d_{\cos}(a, b) \le 1.$$

Examples

$$a = (1, 2, 2)$$
 $b = (3, 4, 0)$

Distance between time series



use dynamic time warping

k-NN regression

y is continuous x = test data

 $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$ are k-nearest neighbors of x.

Prediction:

$$\hat{y} = \frac{1}{k} \sum_{i=1}^{k} y_i$$

Example of 5-NN

Training set:

$$(2,1)$$
 $(5,3)$ $(10,6)$ $(4,9)$ $(1,3)$ $(8,2)$ $(5,8)$ $(7,8)$ $(1,4)$

New point: x = 6. The 5-NN prediction is