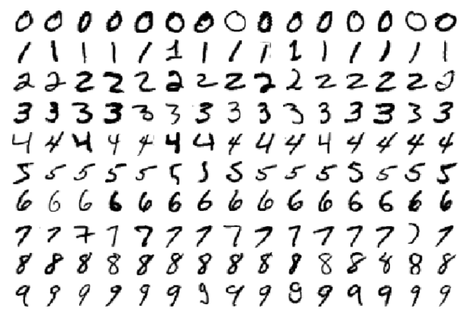
Nearest neighbor classifiers

Example: OCR for digits

1. Classify images of handwritten digits by the actual digits they represent.
2. Classification problem: $\mathcal{Y} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ (a discrete set).



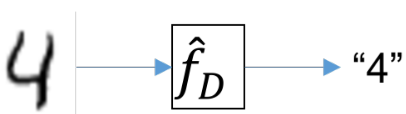
Nearest neighbor (NN) classifier

Given: labeled examples $D := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$

Predictor: $\hat{f}_D \colon \mathcal{X} \to \mathcal{Y}$

On input $\boldsymbol{x}$,

1. Find the point $\boldsymbol{x}_i$ among $\{x_i\}_{i=1}^n$ that is "closest" to $\boldsymbol{x}$ (the *nearest neighbor*).
2. Return y_i.

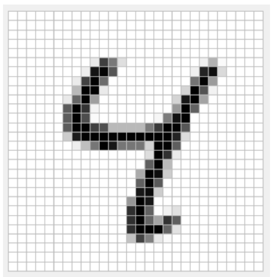


How to measure distance?

A default choice for distance between points in $\mathbb{R}^d$ is the *Euclidean distance* (also called ℓ_2 distance):

$$\|\boldsymbol{u} - \boldsymbol{v}\|_2 \;:=\; \sqrt{\sum_{i=1}^{d} (u_i - v_i)^2}$$

(where $\boldsymbol{u} = (u_1, u_2, \ldots, u_d)$ and $\boldsymbol{v} = (v_1, v_2, \ldots, v_d)$).



Grayscale 28×28 pixel images.

Treat as *vectors* (of 784 real-valued *features*) that live in $\mathbb{R}^{784}$.

Example: OCR for digits with NN classifier

- Classify images of handwritten digits by the digits they depict.

- $\mathcal{X} = \mathbb{R}^{784}$, $\mathcal{Y} = \{0, 1, \ldots, 9\}$.
- **Given**: labeled examples $D := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$.
- Construct NN classifier $\hat{f}_D$ using D.
- **Question**: Is this classifier any good?

Error rate

- *Error rate* of classifier f on a set of labeled examples D:

$$\mathrm{err}_D(f) \;:=\; \frac{\#\text{ of } (x, y) \in D \text{ such that } f(x) \neq y}{|D|}$$

  (i.e., the fraction of D on which f disagrees with paired label).
- Sometimes, we'll write this as $\mathrm{err}(f, D)$.
- **Question**: What is $\mathrm{err}_D(\hat{f}_D)$?

A better way to evaluate the classifier

- Split the labeled examples $\{(x_i, y_i)\}_{i=1}^n$ into two sets (randomly).
  - *Training data* S.
  - *Test data* T.
- Only use *training data* S to construct NN classifier $\hat{f}_S$.
  - Training error rate of $\hat{f}_S$: $\mathrm{err}_S(\hat{f}_S) = 0\%$.
- Use *test data* T to evaluate accuracy of $\hat{f}_S$.
  - Test error rate of $\hat{f}_S$: $\mathrm{err}_T(\hat{f}_S) = 3.09\%$.

  Is this good?

Diagnostics

- Some mistakes made by the NN classifier
  (test point in T, nearest neighbor in S):

- First mistake (correct label is "2") could've been avoided by looking at the *three* nearest neighbors (whose labels are "8", "2", and "2").

test point three nearest neighbors

k-nearest neighbors classifier

Given: labeled examples $D := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$
Predictor: $\hat{f}_{D,k} \colon \mathcal{X} \to \mathcal{Y}$:

On input $\boldsymbol{x}$,

1. Find the k points $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ among $\{x_i\}_{i=1}^n$ "closest" to x (the k nearest neighbors).
2. Return the plurality of $y_{i_1}, y_{i_2}, \ldots, y_{i_k}$.

(Break ties in both steps arbitrarily.)

Effect of k

- Smaller k: smaller training error rate.
- Larger k: higher training error rate, but predictions are more "stable" due to voting.

OCR digits classification

k	1	3	5	7	9
Test error rate	0.0309	0.0295	0.0312	0.0306	0.0341

Choosing k

The hold-out set approach

1. Pick a subset $V \subset S$ (*hold-out set*, a.k.a. *validation set*).
2. For each $k \in \{1, 3, 5, \ldots\}$:
   - Construct k-NN classifier $\hat{f}_{S \setminus V,k}$ using $S \setminus V$.
   - Compute error rate of $\hat{f}_{S \setminus V,k}$ on V ("hold-out error rate").
3. Pick the k that gives the smallest hold-out error rate.

(There are many other approaches.)

Better distance functions

- **Strings**: edit distance

  $$\mathrm{dist}(u, v) = \#\text{ insertions/deletions/mutations needed to change } u \text{ to } v\,.$$

- **Images**: shape context distance

  $$\mathrm{dist}(u, v) = \text{how much "warping" is required to change } u \text{ to } v\,.$$

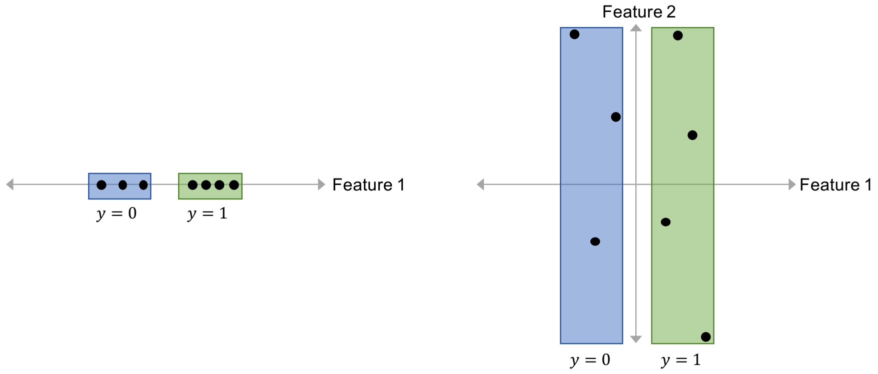
- **Audio waveforms**: dynamic time warping
- Etc.

OCR digits classification

Distance	ℓ_2	ℓ_3	Tangent	Shape
Test error rate	3.09%	2.83%	1.10%	0.63%

Bad features

Caution: nearest neighbor classifier can be broken by bad/noisy features!

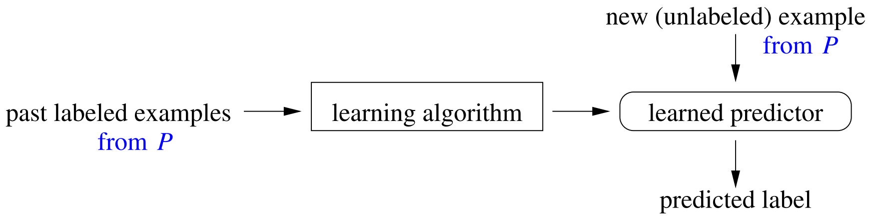


Questions of interest

1. How good is the classifier learned using NN *on your problem*?
2. Is NN a good learning method *in general*?

Statistical learning theory

Basic assumption (main idea):
labeled examples $\{(x_i, y_i)\}_{i=1}^n$ come from same source as future examples.



More formally:
$\{(x_i, y_i)\}_{i=1}^n$ is an *i.i.d. sample* from a probability distribution P over $\mathcal{X} \times \mathcal{Y}$.

Prediction error rate

- Define the *(true) error rate* of a classifier $f : \mathcal{X} \to \mathcal{Y}$ w.r.t. P to be

$$\operatorname{err}_P(f) \;:=\; P(f(X) \neq Y)$$

  where (X, Y) is a pair of random variables with joint distribution P (i.e., $(X, Y) \sim P$).
- Let $\hat{f}_S$ be classifier trained using labeled examples S.
- True error rate of $\hat{f}_S$ is

$$\operatorname{err}_P(\hat{f}_S) \;:=\; P(\hat{f}_S(X) \neq Y)\,.$$

- We cannot compute this without knowing P.

Estimating the true error rate

- Suppose $\{(x_i, y_i)\}_{i=1}^n$ (assumed to be an i.i.d. sample from P) is randomly split into S and T, and $\hat{f}_S$ is based only on S.
- $\hat{f}_S$ and T are *independent*, and the *test error rate* of $\hat{f}_S$ is an *unbiased* estimate of the true error rate of $\hat{f}_S$.
- If $|T| = m$, then the test error rate $\mathrm{err}_T(\hat{f}_S)$ of $\hat{f}_S$ (conditional on S) is a *binomial random variable* (scaled by $1/m$):

$$m \cdot \mathrm{err}_T(\hat{f}_S) \mid S \ \sim \ \mathrm{Bin}(m, \mathrm{err}_P(\hat{f}_S)).$$

- The expected value of $\mathrm{err}_T(\hat{f}_S)$ is $\mathrm{err}_P(\hat{f}_S)$.
  (This means that $\mathrm{err}_T(\hat{f}_S)$ is an *unbiased estimator* of $\mathrm{err}_P(\hat{f}_S)$.)
- The standard deviation of $\mathrm{err}_T(\hat{f}_S)$ is at most $\frac{1}{\sqrt{m}}$.

Limits of prediction

- Binary classification: $\mathcal{Y} = \{0, 1\}$.
- Probability distribution P over $\mathcal{X} \times \{0, 1\}$; let $(X, Y) \sim P$.
- Think of P as being comprised of two parts.
  1. Marginal distribution μ of X (a distribution over $\mathcal{X}$).
  2. Conditional distribution of Y given $X = x$, for each $x \in \mathcal{X}$:

$$\eta(x) \ := \ P(Y = 1 \mid X = x).$$

- If $\eta(x)$ is 0 or 1 for all $x \in \mathcal{X}$ where $\mu(x) > 0$,
  then optimal error rate is zero (i.e., $\min_f \mathrm{err}_P(f) = 0$).
- Otherwise it is non-zero.

Bayes optimality

- What is the classifier with smallest true error rate?

$$f^\star(x) \ := \ \begin{cases} 0 & \text{if } \eta(x) \le 1/2; \\ 1 & \text{if } \eta(x) > 1/2. \end{cases}$$

  (Do you see why?)
- $f^\star$ is called the *Bayes (optimal) classifier*, and

$$\mathrm{err}_P(f^\star) \ = \ \min_f \mathrm{err}_P(f) \ = \ \mathbb{E}\Big[\min\{\eta(X), 1 - \eta(X)\}\Big]$$

  which is called the *Bayes (optimal) error rate*.

Question:
How far from optimal is the classifier produced by the NN learning method?

Consistency of k-NN

We say a learning algorithm A is **consistent** if

$$\lim_{n \to \infty} \mathbb{E}\Big[\mathrm{err}_P(\hat{f}_n)\Big] \ = \ \mathrm{err}(f^\star),$$

where $\hat{f}_n$ is the classifier learned using A on an i.i.d. sample of size n.

Theorem (e.g., Cover and Hart 1967)

Assume η is continuous. Then:

- *1-NN is consistent if* $\min_f \mathrm{err}_P(f) = 0$.
- *k-NN is consistent, provided that $k := k_n$ is chosen as an increasing but sublinear function of n:*

$$\lim_{n \to \infty} k_n \ = \ \infty, \qquad \lim_{n \to \infty} \frac{k_n}{n} \ = \ 0.$$

Key takeaways

1. k-NN learning procedure; role of k, distance functions, features.
2. Training and test error rates.
3. Framework of statistical learning theory; estimating the "true" error rate; Bayes optimality; high-level idea of consistency.