k-Nearest Neighbor

$$'S' = \{(\bar{x}_1, y_1), \ldots, (\bar{x}_n, y_n)\}.$$

$(\bar{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$

$\mathcal{X}$ – feature space (all possible values the features can take).

e.g., $\mathcal{X} = \{0, \ldots 255\}^{784}$ for MNIST, etc...

QN:- What is $\mathcal{X}$ for the "TENNIS"?

$\mathcal{Y}$ – label space (possible values of labels).

$d : \mathcal{X} \times \mathcal{X} \longrightarrow [0, \infty)$, a notion of distance between features.

i.e., for $\bar{x}_i, \bar{x}_j \in \mathcal{X}$, $d(\bar{x}_i, \bar{x}_j)$ = distance between $\bar{x}_i, \bar{x}_j$.

k-NN algorithm

- Given $k \in \mathbb{N}$, 'S' (trng set), 'd' (distance).
- For a new test feature $\bar{x}_{n+1} \in \mathcal{X}$,

- find the k closest examples to x_{n+1} in 'S'.
- out put the $\boxed{\text{majority}}$ of the labels of these k examples.

1-NN :- find $j^* = \arg\min_j d(\bar{x}_{n+1}, \bar{x}_j)$ $\left\{ x_{j^*} \text{ - closest to } \bar{x}_{n+1} \right\}$

output y_{j^*}

3-NN :- find '3' closest points to $\bar{x}_{n+1}$ in 'S'.

say $\bar{x}_{j_1}, \bar{x}_{j_2}, \bar{x}_{j_3}$ ⟶ closest

then consider $\underline{y_{j_1}, y_{j_2}, y_{j_3}}$ → o/p majority.

Advantages :- 1. Good performance.

2. No training time
3. Simplest (?)

Disadvantages :- 1. High run-time for each example

2. Storage.
3. choose right '$\underline{d}$'.
4. Normalization (related to '$\underline{\underline{3}}$').

EXAMPLE :- Let $\mathcal{X} = \mathbb{R}^2$ (2-dimensional features).

$\mathcal{Y} = \{-1, 1\}$ (binary labels)

d :- squared Euclidean distance between features

Consider data :-

Each $(\bar{x}_i, y_i) = ((\bar{x}_i^1, \bar{x}_i^2), y_i)$ is generated independently as follows:-

$x_i^1 \longrightarrow$ uniform $(-10^4, 10^4)$

$\bar{x}_i^2 \longrightarrow$ uniform $(-1, 1)$.

Label $y_i = \text{sign}(\bar{x}_i^2) = \begin{cases} 1 & \text{if } \bar{x}_i^2 \geq 0, \\ 0 & \text{otherwise}. \end{cases}$

sign of second attribute is the label ⟶ perfect classification.

Suppose we get about 40 examples.

⊛ distance to closest point will be dominated by closeness in first coordinate, which is just <u>noise</u> wrt labels.

--- a decision tree would be much better