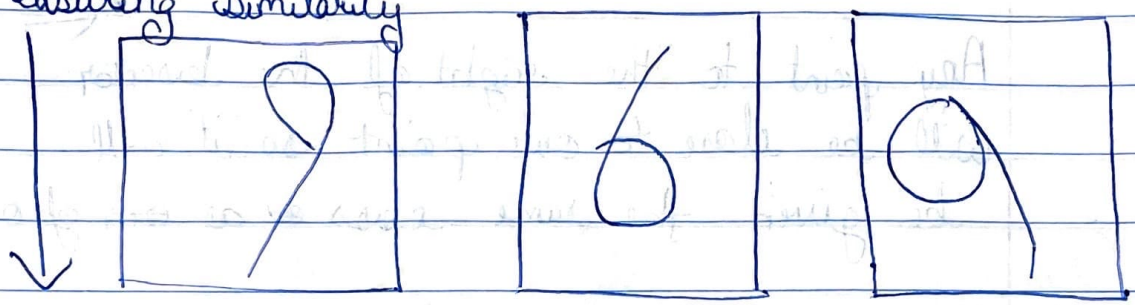
vRPnuC

Lecture - 16

vRPnuC

✱ Similarity

Measuring Similarity



Compute features.

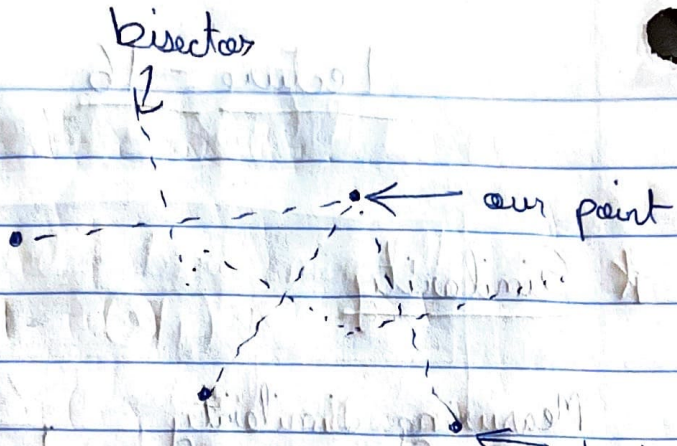
✱ The k nearest neighbours Algorithm (KNN).

Given a test point x;
classify using k of its nearest neighbour.

$x_{[1]}$ → closest to our test point.
$x_{[2]}$ → A bit far
$x_{[N]}$ → too far ~~far~~ from our test data point.

where $d(x_{[1]}, x) \leq d(x_{[2]}, x) \ldots d(x_{[N]}, x)$

output = $sign(y_{[1]})$.



Any point to the right of the bisector will be close to our point so it will be given the same class as of our point.

Any point to the left will be close to that point.

We dont need to make tesselation (region of the data points)

Just calculate the distance, and find the nearest data point. Assigning them +ve or -ve depending on the distance.

* KNN algorithm will give you 0 in sample error. Just by considering itself as the closest data point.

KNN will always have 0 in sample error ($E_{in}=0$)

though we cannot calculate E_{in} & relation.
We can calculate E_{out}.

KNN algo. → $E_{out}(g) \leq 2E^*_{out}$ → Noisy target func^n.

$$E_{out}(g) \leq 2E^*_{out}$$

as $N \to \infty$ (for high data sample)
the $E_{out}(g) \approx 2E^*_{out}$.
i.e $E_{out}(g)$ will be small.

Noisy → $\pi(x) = P[y = +1 \mid x]$
(given x what is the probability of it to give $y = +1$]

Unknown Target.

$$1 - \pi(x) = P[y = -1 \mid x]$$

* Optimal classifier.

$\pi(x) \geq \frac{1}{2} \longrightarrow +1$

$\pi(x) < \frac{1}{2} \longrightarrow -1.$

E^*_{out} (Optimal classifier.

$$E^*_{out}(x) = \begin{cases} 1 - \pi(x) & \pi(x) \geq \frac{1}{2} \\ \pi(x) & \pi(x) < \frac{1}{2} \end{cases}$$

$\pi(x) \geq \frac{1}{2} \rightarrow = 1$

$\pi(x) < \frac{1}{2} \xrightarrow{\text{!!!}} + 1.$

$$= \min\{\pi(x), 1 - \pi(x)\}$$

$$= \eta(x).$$

$$E^*_{out} = \int dx \, P(x) E^*_{out}(x).$$

↑ optimal.

★ A few assumptions:

★ [Property 1]: $\pi(x)$ is continuous.

[Property 2]: for all $N \to \infty$

$$x_{[1]} \to x.$$

$E_{out}(x)$ = prob. of error (x).

$$= P[y_{[1]} \neq y].$$

↑ Nearest label ↑ True label.

$$E_{out}(x) = \underbrace{P[y_{[1]} = +1}_{\pi(x_{[1]})}, y = -1] + P[y_{[1]} = -1, y = +1]$$

$$= \pi(x_{[1]})(1 - \pi(x)) + (1 - \pi(x_{[1]}))(\pi(x))$$

* As $N \to \infty$
As $x_{[1]}$ converges to x [Continuous]
$\pi(x)_{[1]}$ also converges to $\pi(x)$.

$$E_{out}(x) = 2\pi(x)(1 - \pi(x))$$
$$\leq 2 \min\{\pi(x), 1 - \pi(x)\}$$

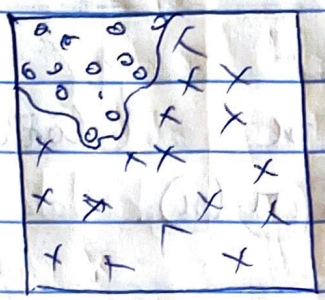
$$E_{out} \leq 2E^*_{out}$$

So if E^*_{out} is small, E_{out} will be small and we will have tight bound.

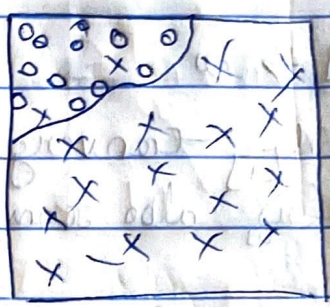
If E^*_{out} is large, E_{out} will be large and we will have loose bound.

With N ↑ the model will become complex.
We can tackle this by increasing K↑

But if K is very ~~big~~ large ~~will~~ we will underfit.

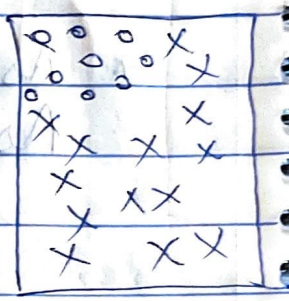


1 NN rule
=
↓
K



21 NN rule
↓
k.

Will eliminate outliers.



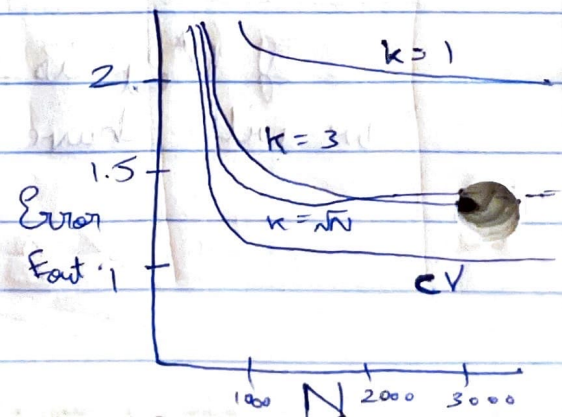
127 NN rule
↓
K

As N ↑ , try to ↑ K

Rule of thumb $K = \lceil \sqrt{N} \rceil$

Choosing K.

CV ⇒ Cross Validation.

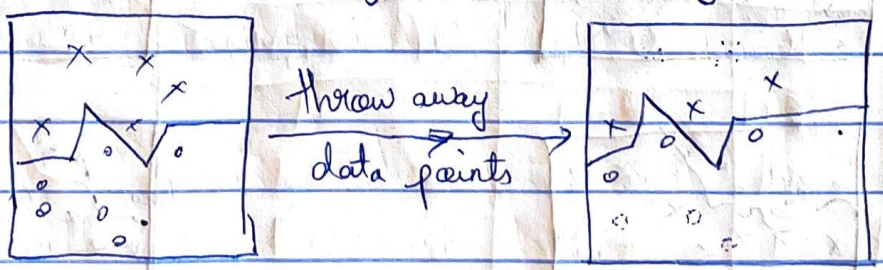


NN	Linear
• No parameters	d+1 parameters.
• Expressive, flexible	Rigid, always linear
• $g(x)$ defined by the data.	$g(x)$ defined by weights.
• generic, can model any target	specialized.

4 KNN :- Simple
No training.
Near Optimal out of sample error.
Can be used for regression or logistic regression

Problem :- computationally demanding.

Sol^ :- Condensed Nearest Neighbour.
↳ choosing a subset of the dataset.



but still maintains the boundary.