K-Nearest Neighbors : 1967

BIG IDEA: If it walks like a duck and it quacks like a duck → its probably a duck!

→ data points that are similar have similar labels

expressed in terms of distance! (other ways to express similarity is dot product)

→ Lazy learning

during training only a set of labeled data is stored!

1 Neighbor

prediction $y_{x_{test}}$

→ find training point closest to x_{test} AND

$\text{Label}_{x_{test}} \overset{set}{=}$ label closest data point

k-Neighbors

→ prediction $y_{x_{test}}$

→ find k training points that are closest to x_{test}

binary classification

→ predict majority class from $y_1, .., y_k$

AND weigh the data with respect to distance

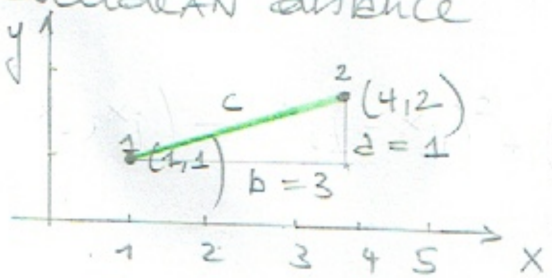
$(ex\ \frac{1}{d^2})$

Regression

→ predict average $y_1, .., y_k$

Distance Metric:

→ mainly Euclidean distance



$c^2 = a^2 + b^2$

$\rightarrow c = \sqrt{a^2+b^2}$

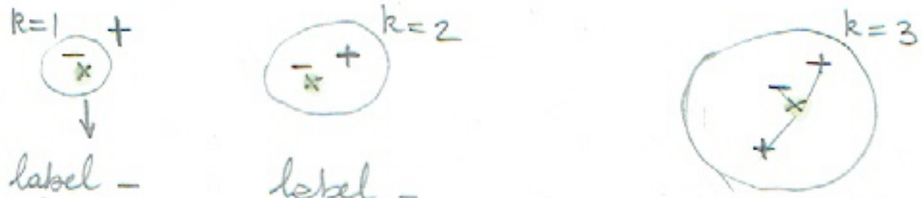
$a = y_2 - y_1$

$b = x_2 - x_1$ $\rightarrow c = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2} = \sqrt{10}$

Manhattan Distance $= |y_2 - y_1| + |x_2 - x_1| = 4$

Choice of k is important

↳ hyperparameter!



label −
Majority vote

label −

label → +
Majority vote

you could also multiply labels with $1/d^2$ → higher distance → less influence

k too small → sensitive to noise
too large → neighborhood contains points from other classes

→ Find optimum by trying different ks

Scale data because algorithm is sensitive to distance
→ prevents features from dominating!

Curse of Dimensionality

↳ in High Dim → All points are far from each other!
→ Eucl. distance not helpfull
→ KNN will not work!

solution
→ Dim Reduction
- PCA
- SVD

INTUITION:

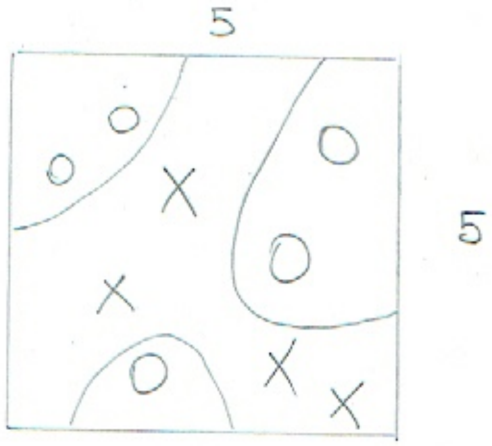
Assume 10 pics of cats (O) & dogs (X) (color images)

1D classifier |OXOXOOOXXXXX| width is 5 units
2 pics per unit → density = 2
↳ 1 feature
RED

→ not possible to find linear classifier that separates cats from dogs!

2D
↓
2 features

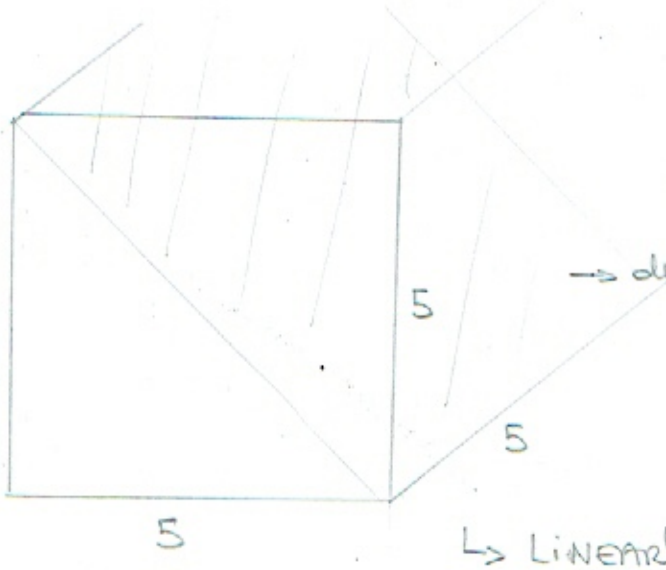
RED
GREEN



→ density $= \frac{10}{25}$ OR 0.4 per interval

3D
↓
3 features

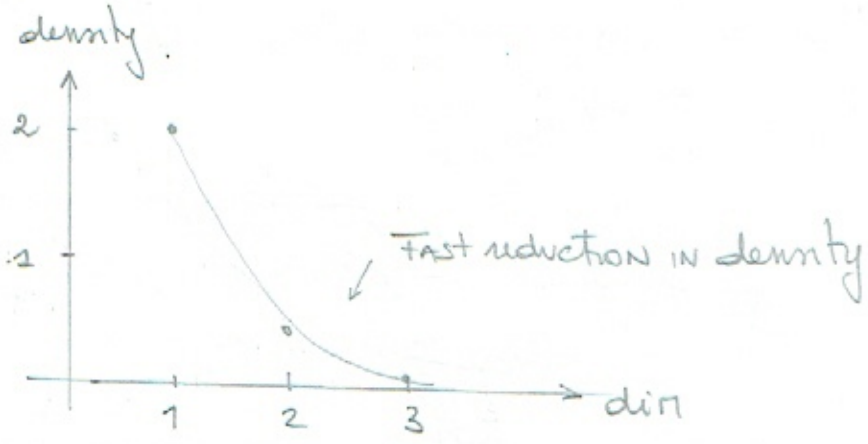
Red
Green
Blue



→ density $= \frac{10}{125} = 0.08$ per interval.

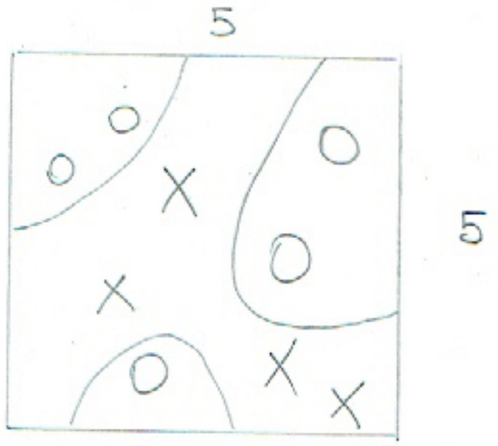
↳ Linearly separable (hyperplane)

If # features (dimensions) ↑ → data becomes more sparse



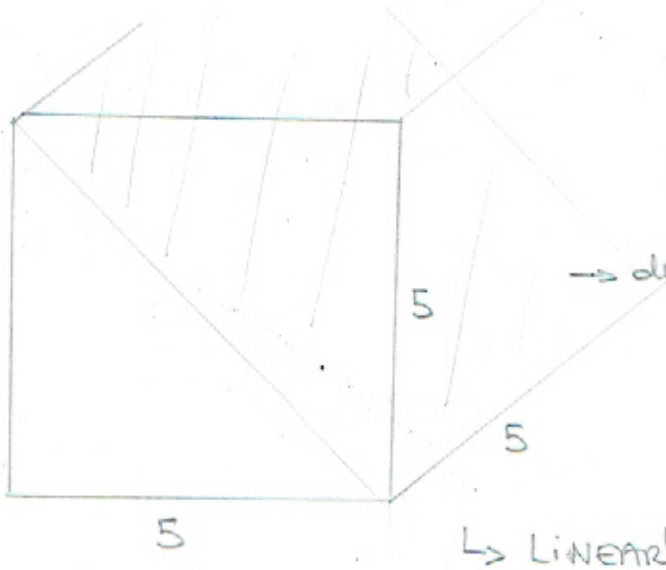
4D → density = 0.016
5D → density = 0.003
⋮

2D
↓
2 features
RED
GREEN



→ density = $\frac{10}{25}$ OR 0.4 per interval

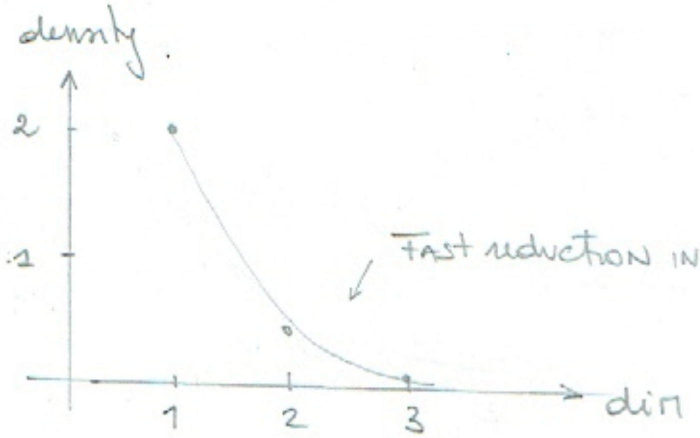
3D
↓
3 features
Red
Green
Blue



→ density = $\frac{10}{125}$ = 0.08 per interval.

↳ Linearly separable (hyperplane)

If # features (dimensions) ↑ → data becomes more sparse



4D → density = 0.016
5D → density = 0.003
⋮