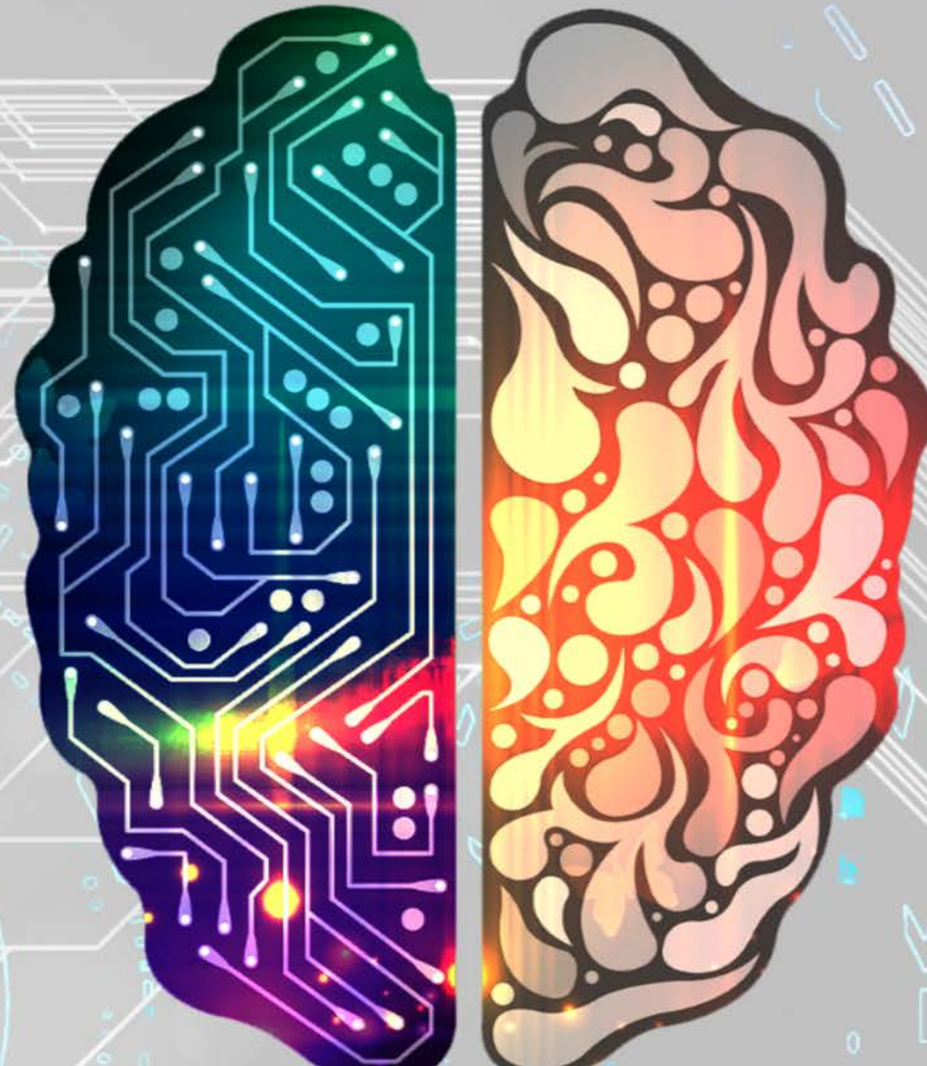
CMPS 460 – Spring 2022

MACHINE LEARNING

Tamer Elsayed



3.a

Geometry and Nearest Neighbors

3-3.3

Roadmap …

- Nearest Neighbors (NN) algorithms for classification
  - kNN, Epsilon ball NN

- Fundamental Machine Learning Concepts
  - Decision boundary

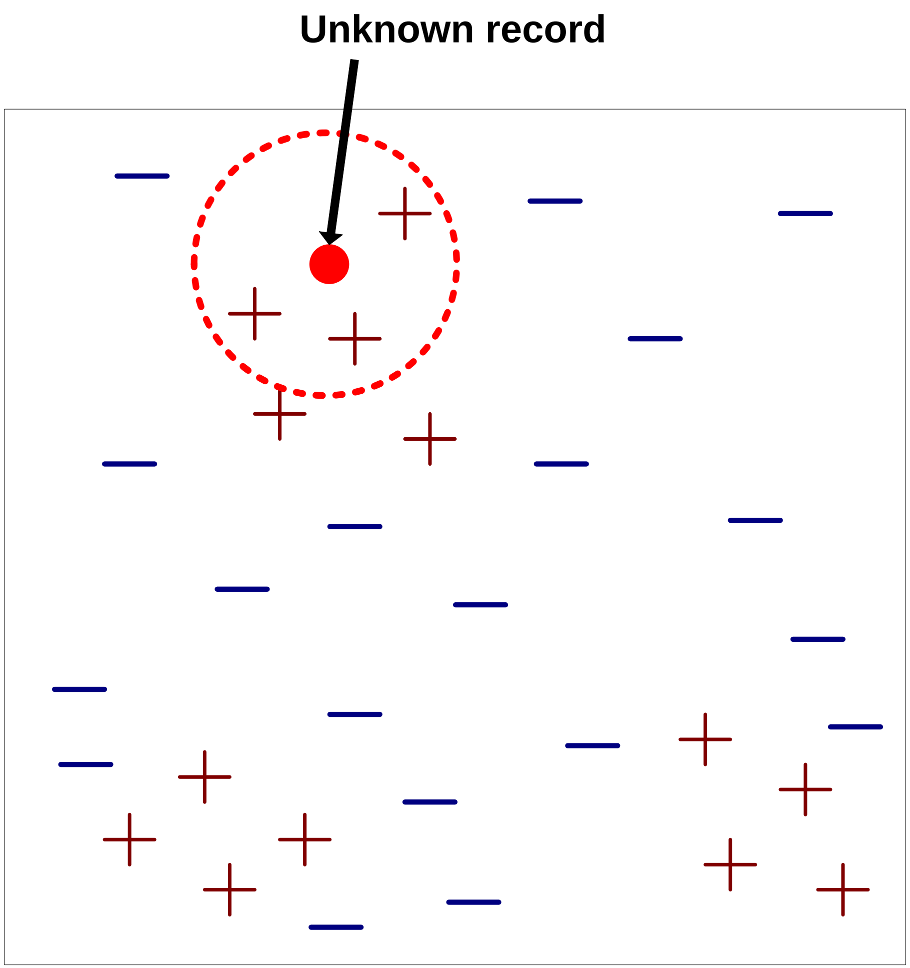
Intuition for NN …

This “rule of nearest neighbor” has considerable elementary intuitive appeal and probably corresponds to practice in many situations. For example, it is possible that much medical diagnosis is influenced by the doctor’s recollection of the subsequent history of an earlier patient whose symptoms resemble in some way those of the current patient.

(Fix and Hodges, 1952)

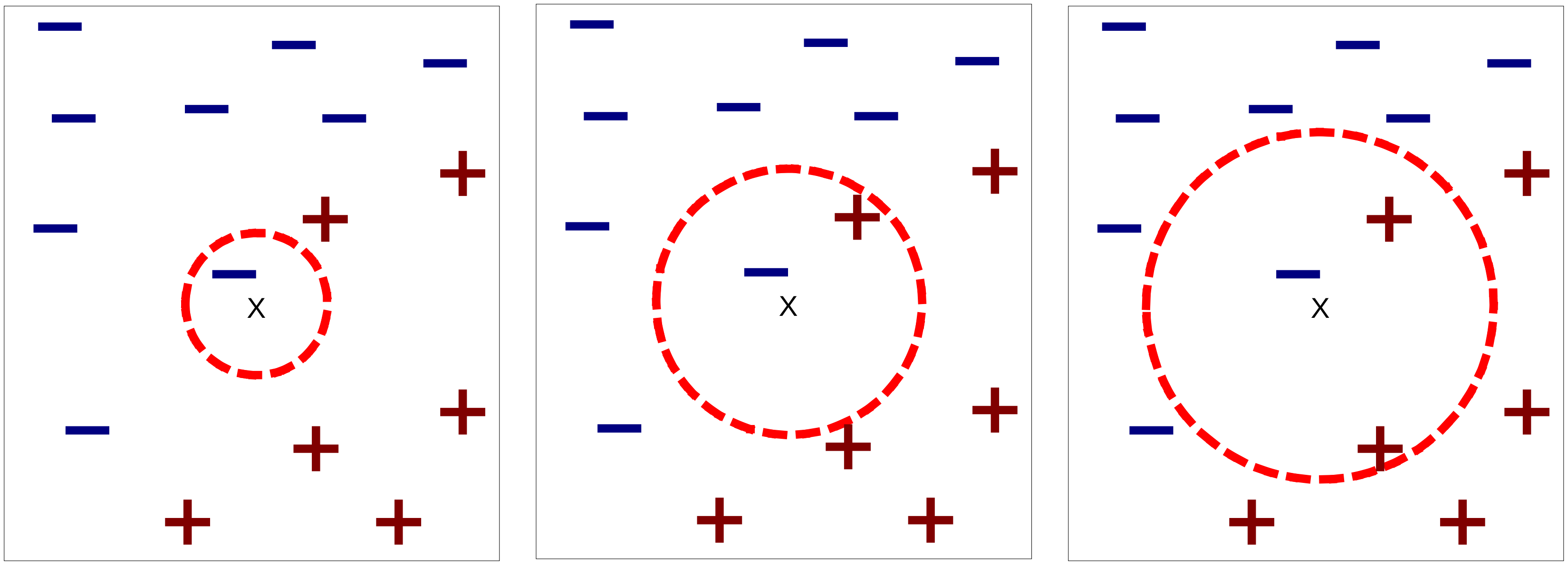
Simple idea …

- Store all training examples
  - Each point is a "vector" of attributes
- Classify new examples based on most "similar" training examples
  - Similar means "closer" in vector space



What's done in training?

K Nearest Neighbors (kNN)



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

kNN Algorithm

Algorithm 3 $\textsc{KNN-Predict}(\mathbf{D}, K, \hat{\boldsymbol{x}})$

$S \leftarrow [\,]$
for $n = 1$ **to** N **do**
  $S \leftarrow S \oplus \langle \mathrm{d}(\boldsymbol{x}_n, \hat{\boldsymbol{x}}), n \rangle$ // store distance to training example n
end for
$S \leftarrow \textsc{sort}(S)$ // put lowest-distance objects first
$\hat{y} \leftarrow 0$
for $k = 1$ **to** K **do**
  $\langle dist, n \rangle \leftarrow S_k$ // n this is the kth closest data point
  $\hat{y} \leftarrow \hat{y} + y_n$ // vote according to the label for the nth training point
end for
return $\textsc{sign}(\hat{y})$ // return $+1$ if $\hat{y} > 0$ and -1 if $\hat{y} < 0$

2 Approaches to Learning

Eager learning
(e.g., decision trees)

- Learn/Train
  - Induce an abstract model from data
- Test/Predict/Classify
  - Apply learned model to new data

Lazy learning
(e.g., kNN)

- Learn
  - Just store data in memory
- Test/Predict/Classify
  - Compare new data to stored data
- Properties
  - Retains all information seen in training
  - Complex hypothesis space
  - Classification can be very slow

Components of a kNN Classifier

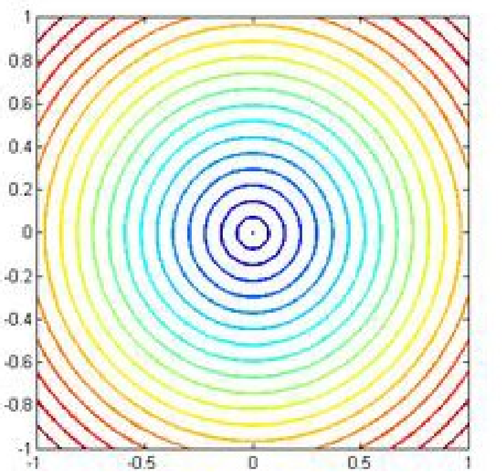
- Distance metric
  - How do we measure distance between instances?
  - Determines the layout of the example space

- The k hyper-parameter
  - How large a neighborhood should we consider?
  - Determines the complexity of the hypothesis space

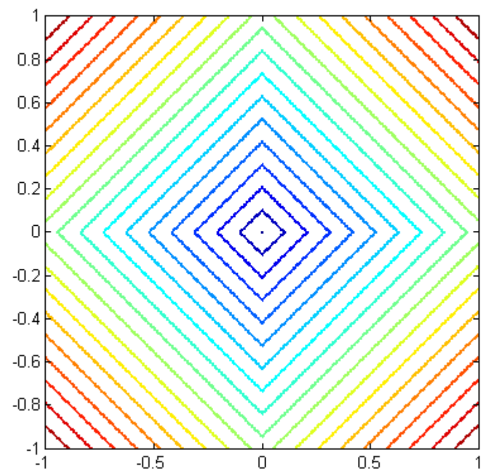
Distance Metrics

- We can use any distance function to select nearest neighbors.
- Different distances yield different neighborhoods

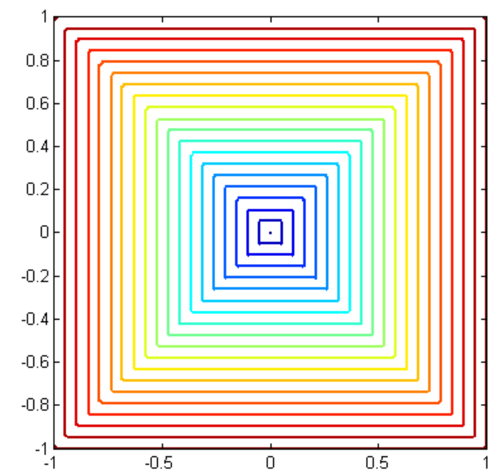
L2 distance
(= Euclidean distance)



L1 distance



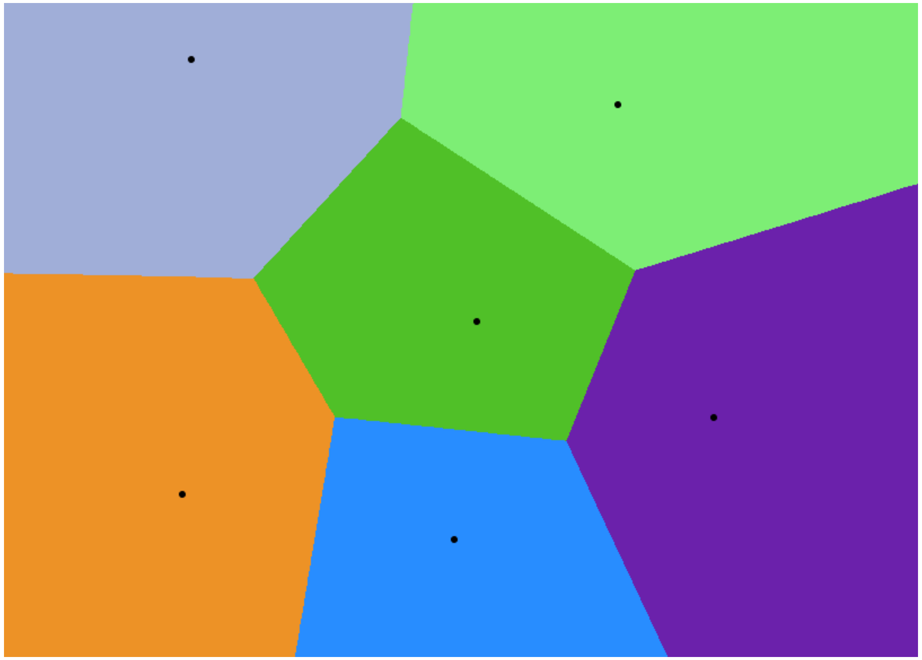
Max norm



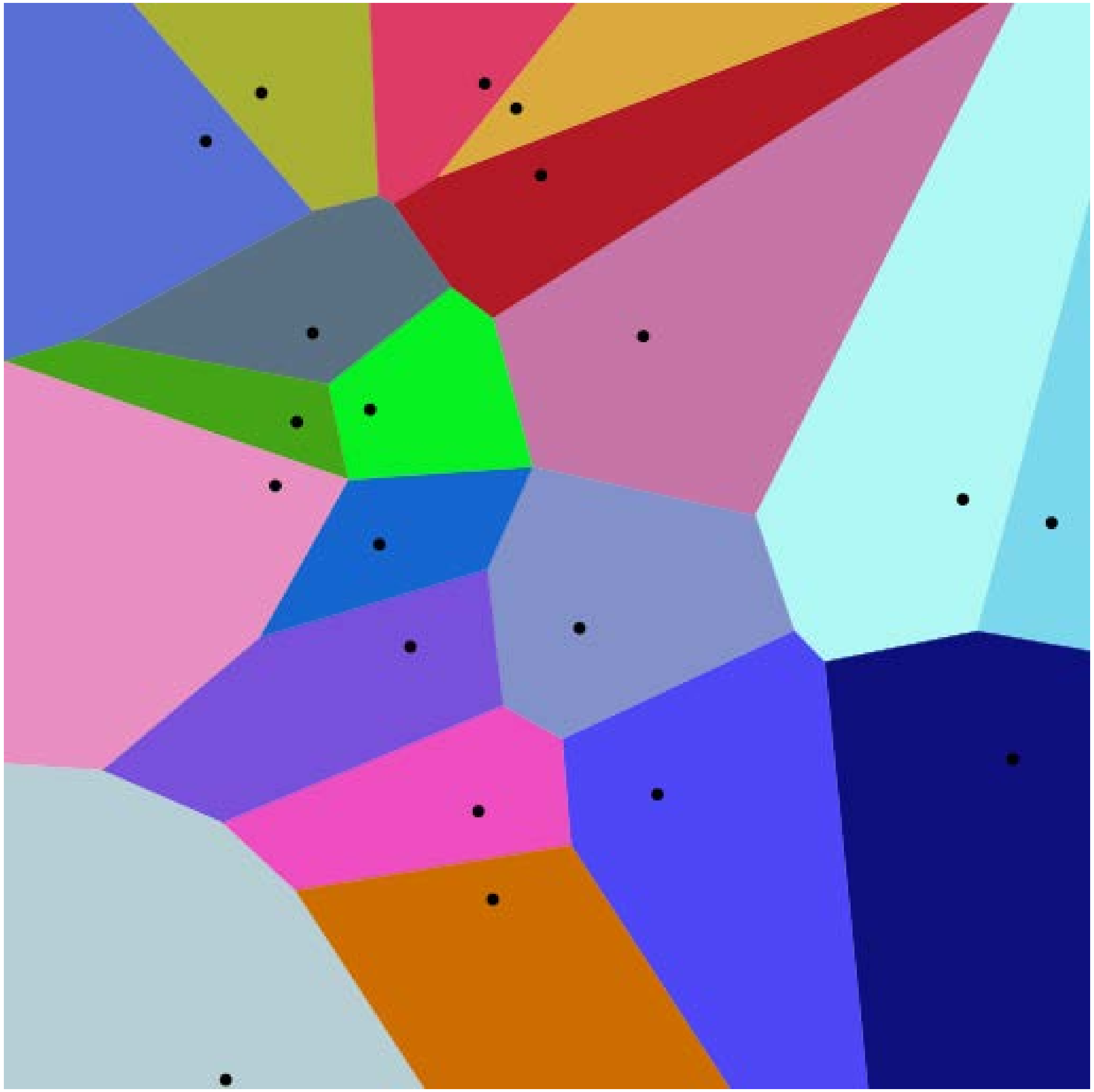
Decision Boundaries

Voronoi Diagram (k=1)

- Regions in feature space closest to every training example
- If test point is in the region corresponding to a given input point, return its label

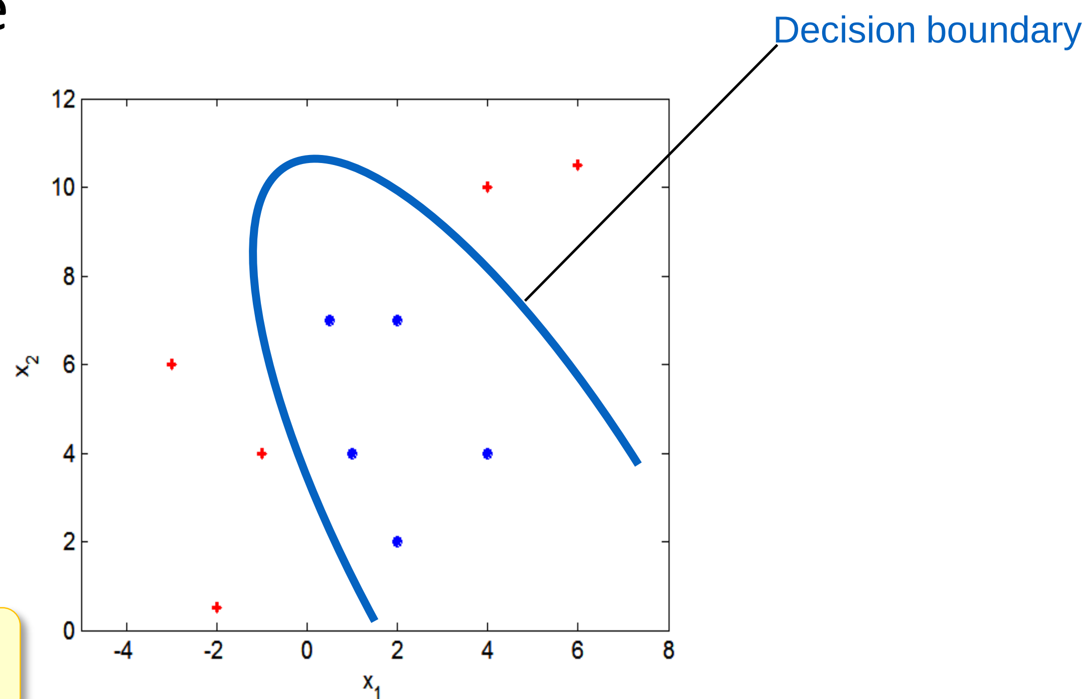


Voronoi Diagram (k=1)



Decision Boundary of a Classifier

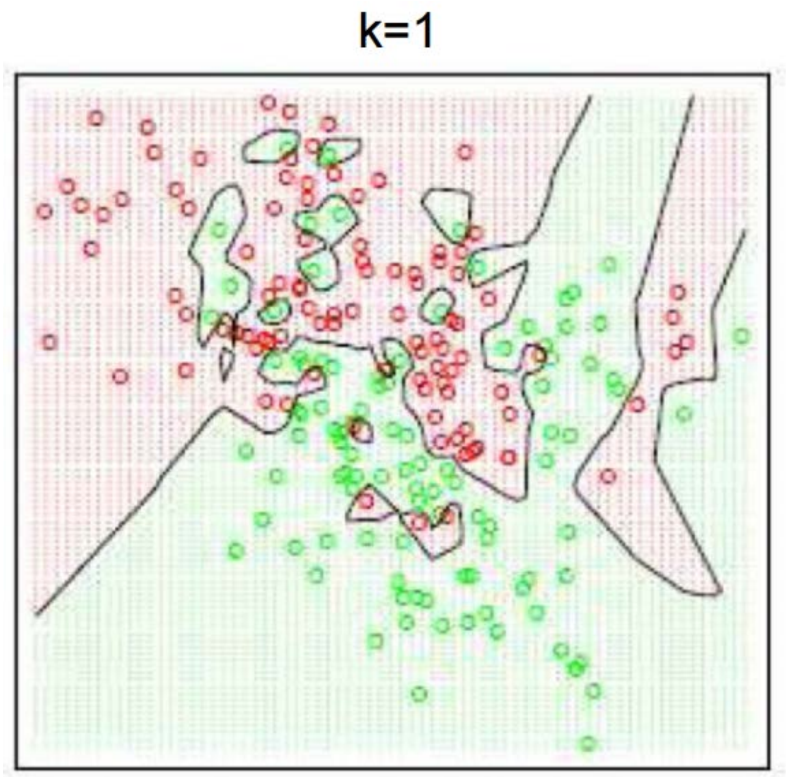
- It is the line that separates positive and negative regions in the feature space

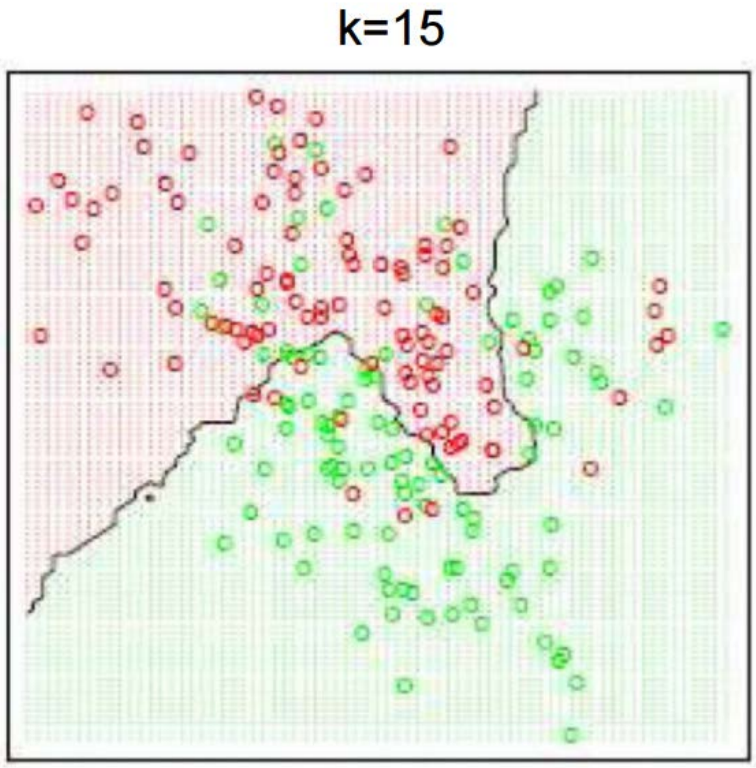


Why is it useful?

- helps visualize how examples will be classified for the entire feature space
- helps visualize the complexity of the learned model

Decision Boundaries of kNN





The k hyper-parameter

- Tunes the complexity of the hypothesis space
  - If k = 1, every training example has its own neighborhood
  - If k = N, the entire feature space is one neighborhood!

- Higher k yields smoother decision boundaries

How would you set k in practice?

Variations on kNN

Weighted Voting

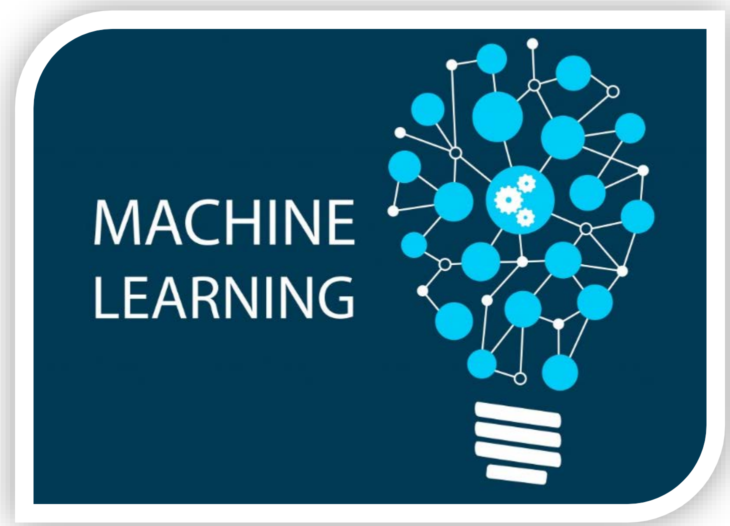
- Default: all neighbors have equal weight
- Extension: weight votes of neighbors by (inverse) distance

Epsilon-Ball Nearest Neighbors

- Same general principle as K-NN, but change the method for selecting which training examples vote

- Instead of using K nearest neighbors, use all examples x such that

$$distance(\hat{x}, x) \leq \varepsilon$$



Issues with kNN

What is the inductive bias of kNN?

- Nearby instances should have the same label
- All features are equally important

Feature Scale

- Example:
  - height of a person may vary from 1.5m to 1.8m
  - weight of a person may vary from 90lb to 300lb
  - income of a person may vary from $10K to $1M

What’s the problem here?

- Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes.

Will fix later …

Irrelevant Features

- There may be non-useful features amongst all features – ***curse of dimensionality***.
- kNN can be easily fooled by irrelevant features.

Will fix later too …