Introduction

C. V. Jawahar

IIIT Hyderabad, India

www.iiit.ac.in/~jawahar

AI and ML at Work

Autonomous Cars and Navigation

"Alexa", "Siri", "Cortana" etc.

Creativity: Generated Images

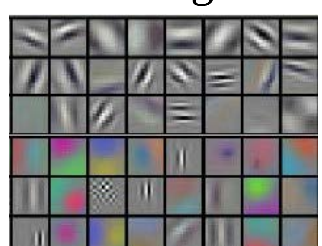
Playing Games better than Human

Modern AI: End 2 End Driving

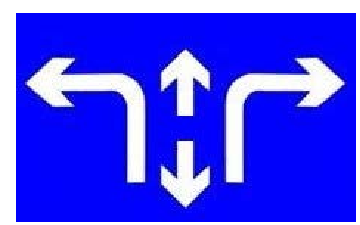
Magic Happens

Street Images

Important Features for Driving



Driving Actions



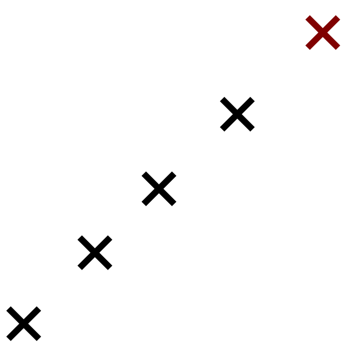
What is Modern AI and ML?

A simple question

- 1, 3, 5, 7, 9, …. What is the next number?
  - Ans: 11. ; Odd numbers, Or 2n+1
- 1, 3, 9, 19, 33, ... What is the next number?
  - Ans: 51; $2n^2+1$
- How do we solve such problems?
  - Find a pattern from the examples.
    - (function f(n) = 2n+1. Or model the data)
  - Use it to predict the next number (or solve the problem)
- How do we design a computational procedure?

A simple question (cont.)

- We know: 1, 3, 9, 19, 33, … What is the next number?
  - Ans: 51; $2n^2+1$
- 0.99, 3.02, 9.00, 18.98, 33.01, … What next?
- Consider a series of 2D points
  - (1,3), (2,6), (3,9), (4,12), ….
  - What is the next point?
  - (x,3x) Or
  - Function:
    - Y = f(X)=3.X



What makes it difficult?

- When numbers are “uncertain”
  - Noise in measurements
  - Missing values
- When numbers are not just “simple numbers”?
  - 2D points, 3D points
  - 100 Dimensional points
- When the function is complex or function nature is unknown
  - Simple linear functions are easy to guess.
  - Finding “best” parameters/coefficients can be hard.

More Examples

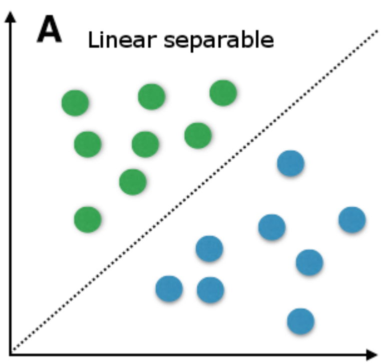
- Given a set of numbers {7,26,17,11,25,32,5,8,92}, partition into two sets: (Unsupervised Learning)
  - Odd (7,17,11,25,5) and Even (26,32,8,92)
  - Why this? Why not single and two digit?
- Given a set of male people with and without anemia, their hemoglobin levels are: (Supervised Learning)
  - Positive cases: {8.5.9.2.7.4.7.8}
  - Negative cases: {15.0, 14.9, 14.2,13.8}
  - Does a patient with 7.7 have anemia?
  - Classification is simple: "anemia if f(x) < 10"
  - Why 10? Why not 12?

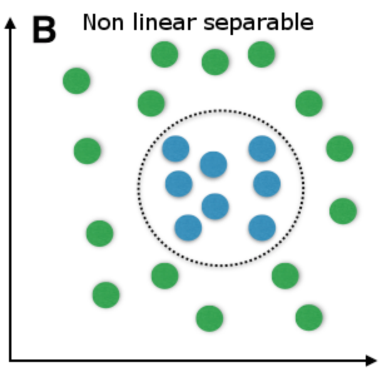
Closer Look ..

- Who gives samples/examples?
  - The Data
  - Data + interpretations (***X***,Y)=(sample, label)
- Who gives functional form?
  - Most problems need complex functions
  - ("Linear" solutions are also good in many cases.)
- How to find the "optimal" parameters?
  - Optimization problem. Training. Computing
- How do we expect that it will work well in the future?

"Classification": A Popular Problem

- Example:
  - Given medical records, predict presence of Malaria
- Data: A set of Samples { X } labeled by experts.
- Performance: Predict accurately on unseen data
- {0,1} classification
  - "Yes" or "No"
  - Yes if f(X) >0
- Multiclass classification
- Many more variants





Problem Space

- ***Feature Extraction:*** Find $\boldsymbol{X}$ corresponding to an entity/item $\boldsymbol{I}$ (such as an image, web page, ECG etc.)

- ***Classification:*** Find a parameterized function $\boldsymbol{f_W(X)}$ which can make the right predictions $\boldsymbol{Y}$.

- ***End to End:*** Can we learn $\boldsymbol{Y}$ directly from $\boldsymbol{I}$.

What is machine learning?

- A branch of **artificial intelligence,**
  - the design and development of algorithms
  - computers to capture and model behaviors
  - based on empirical data.

- Intelligence requires knowledge,
  - It is necessary for the computers to acquire knowledge.
  - Learn from external world; “teachers” etc. and solve problems.

- A very popular area now
  - Lots of data
  - Many recent success stories

What is Machine Learning?

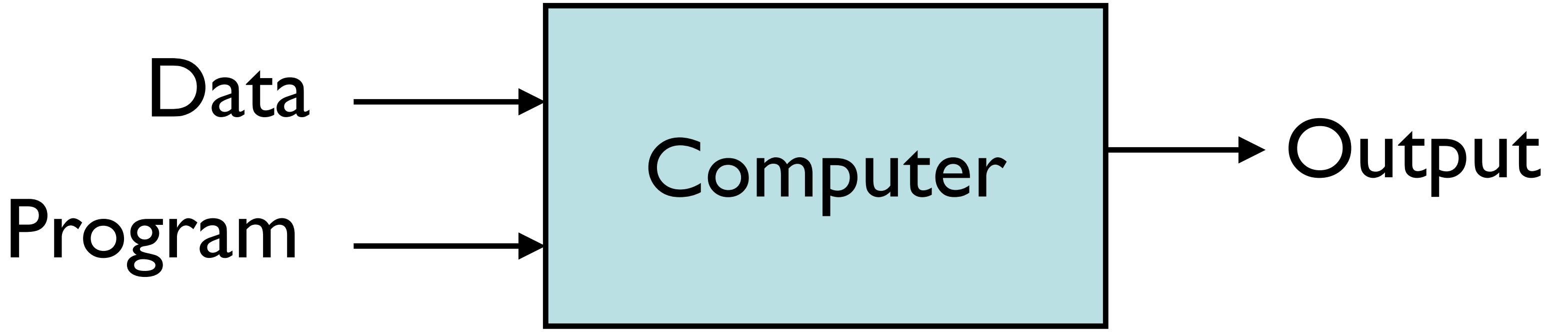
- [Arthur Samuel, 1959]
  - Field of study that gives computers
  - the ability to learn without being explicitly programmed

- [Kevin Murphy] algorithms that
  - automatically detect patterns in data
  - use the uncovered patterns to predict future data or other outcomes of interest

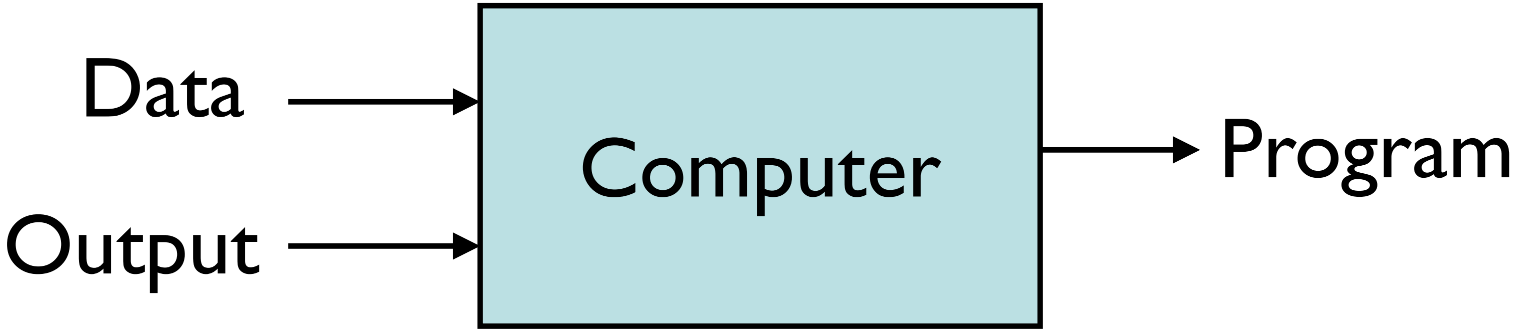
- [Tom Mitchell] algorithms that
  - improve their performance (P)
  - at some task (T)
  - with experience (E)

What is Machine Learning?

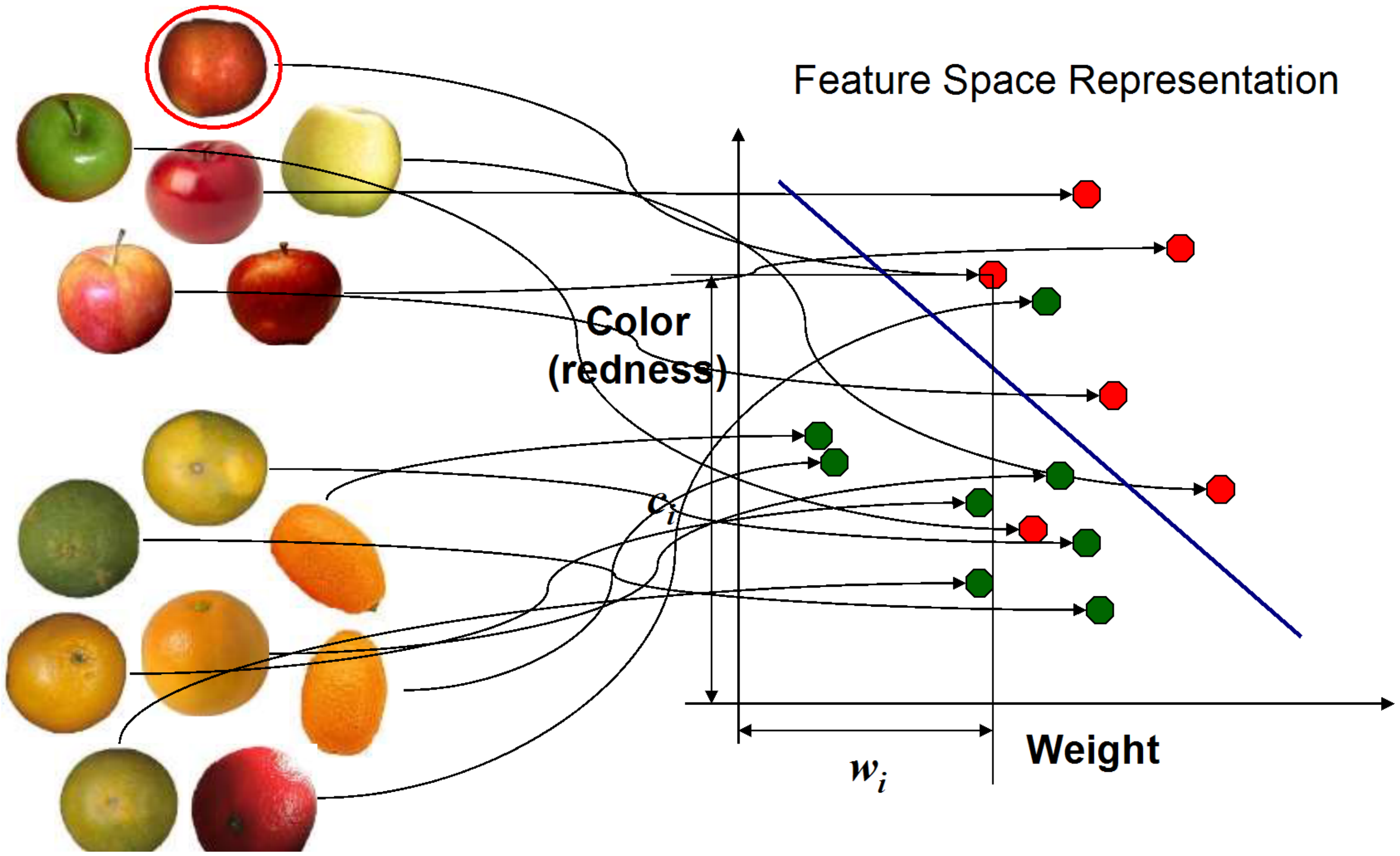
Traditional Programming



Machine Learning



Visualizing a Sample in 2D



Sample/Point and Representation

A sample is easy to visualize in 2D

$$\mathbf{p} = (x, y) \ \textbf{or} \ \ (x_1, x_2) \qquad \mathbf{p} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

and sometime in 3D with some effort

$$\mathbf{p} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

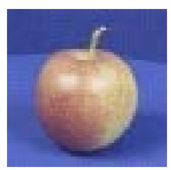
And we often need much larger dimensionality in practice

$$\mathbf{p} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_{100} \end{bmatrix} \qquad \mathbf{p} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_d \end{bmatrix}$$

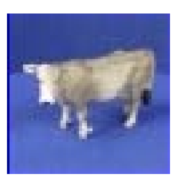
A bit more formal look

The machine learning framework

- Apply a prediction function to a feature representation of the “sample” to get the desired output:

f() = “apple”

f() = “tomato”

f() = “cow”

Slide credit: L. Lazebnik

The machine learning framework

$$y = f(\mathbf{x})$$

output | prediction function | feature or representation

- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error.
- **Testing:** apply f to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

Slide credit: L. Lazebnik

Steps

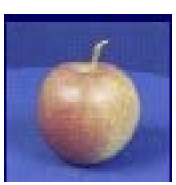
Training

Training Data

Features → Training → Learned model

Training Labels → Training

Testing

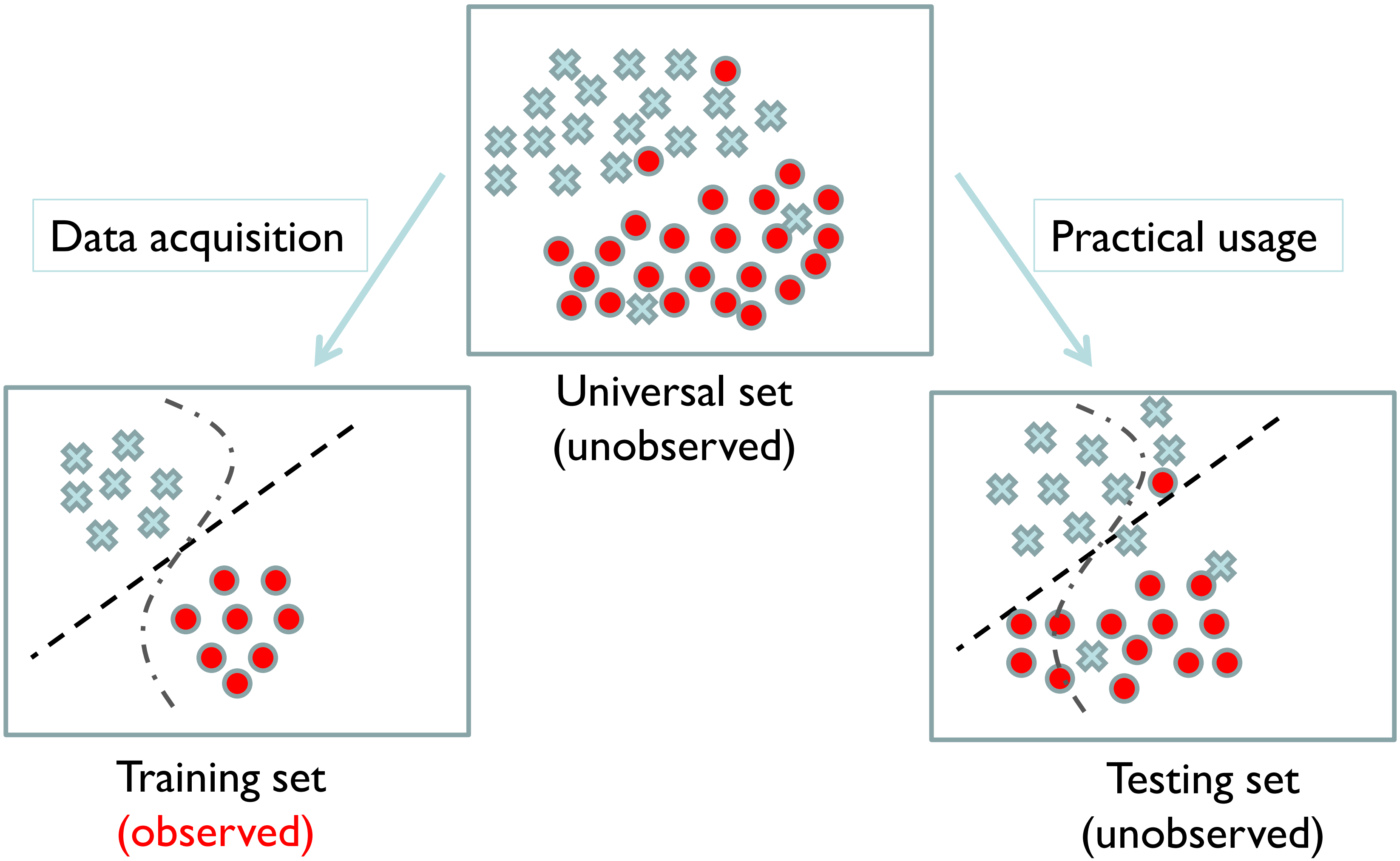


Test sample

Features → Learned model → Prediction

Slide credit: D. Hoiem and L. Lazebnik

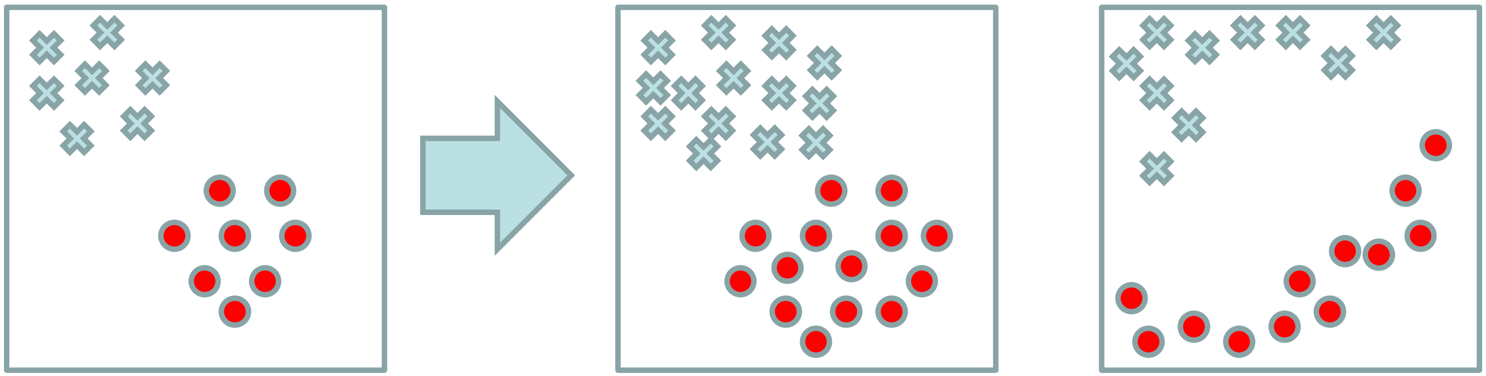
Training and testing



Training and testing

- Training is the process of making the system able to learn.

- Assumptions:
  - Training set and testing set come from the same distribution
  - Need to make some assumptions or bias



Two Prominent Learning Paradigms

- **Supervised learning** is the machine learning task of inferring a function from labeled training data.

- **Unsupervised:** Learn patterns from unlabeled data. Often look for a structure

Nearest Neighbour Methods

C. V. Jawahar
IIIT Hyderabad, India
www.iiit.ac.in/~jawahar

``similar people show similar characters''

Nearest Neighbours

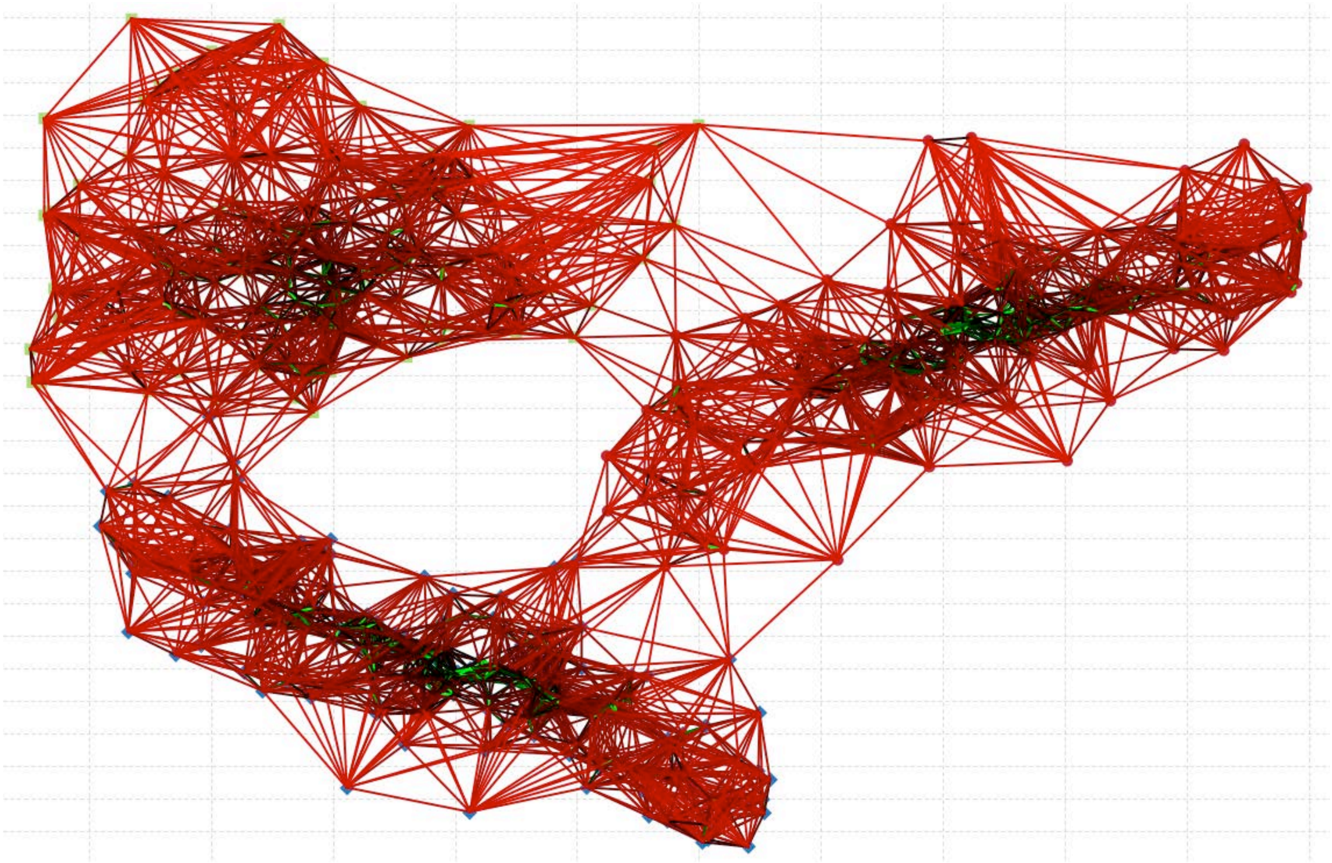
Two types of neighbourhoods:

- K Nearest Neighbours
  - K samples that are closest to you.
- Epsilon Neighbours
  - Samples that have distances smaller than epsilon.

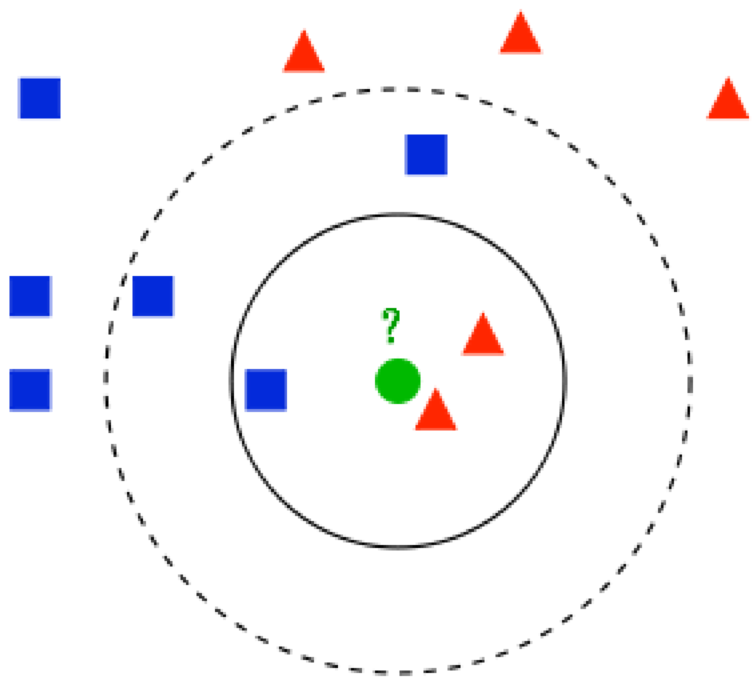
Does x is in the K Nearest neighbourhood of y imply that y is in the K nearest neighbourhood of x?

Does x is in the K Epsilon neighbourhood of y imply that y is in the Epsilon neighbourhood of x?

Nearest Neighbour Graph[Remark*]

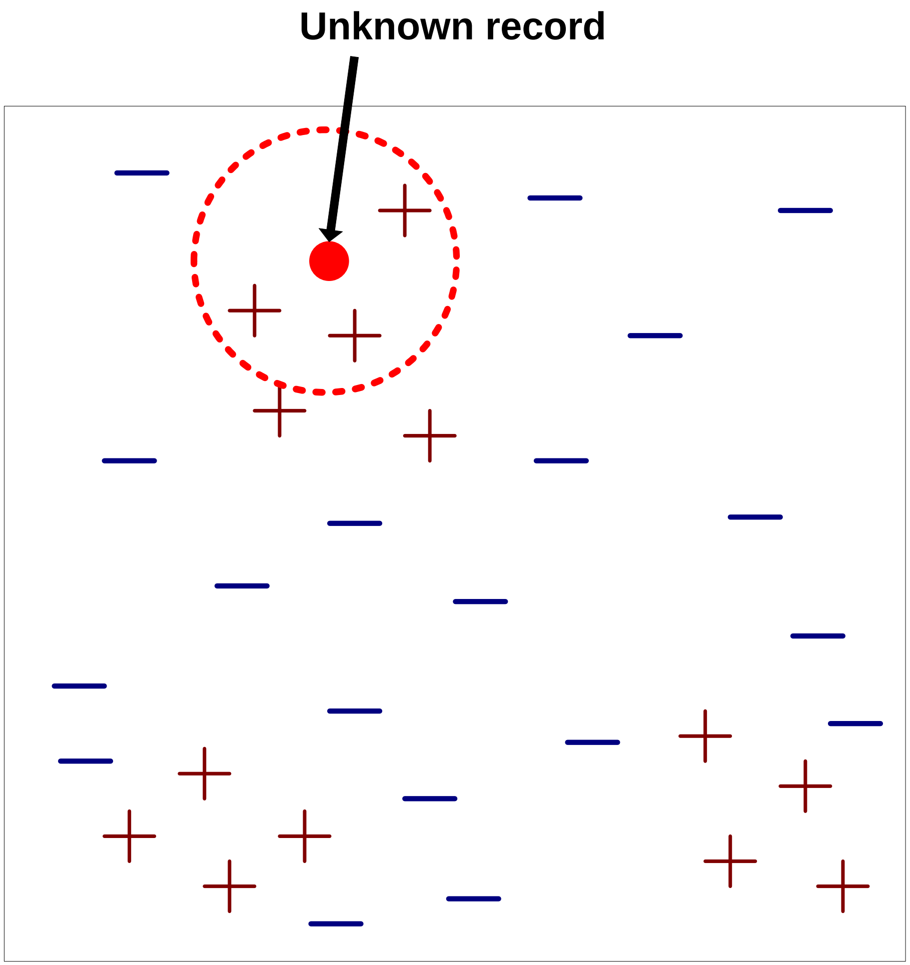


K Nearest Neighbours



Classify the "test" sample as the majority label among the "K-Nearest Neighbours"

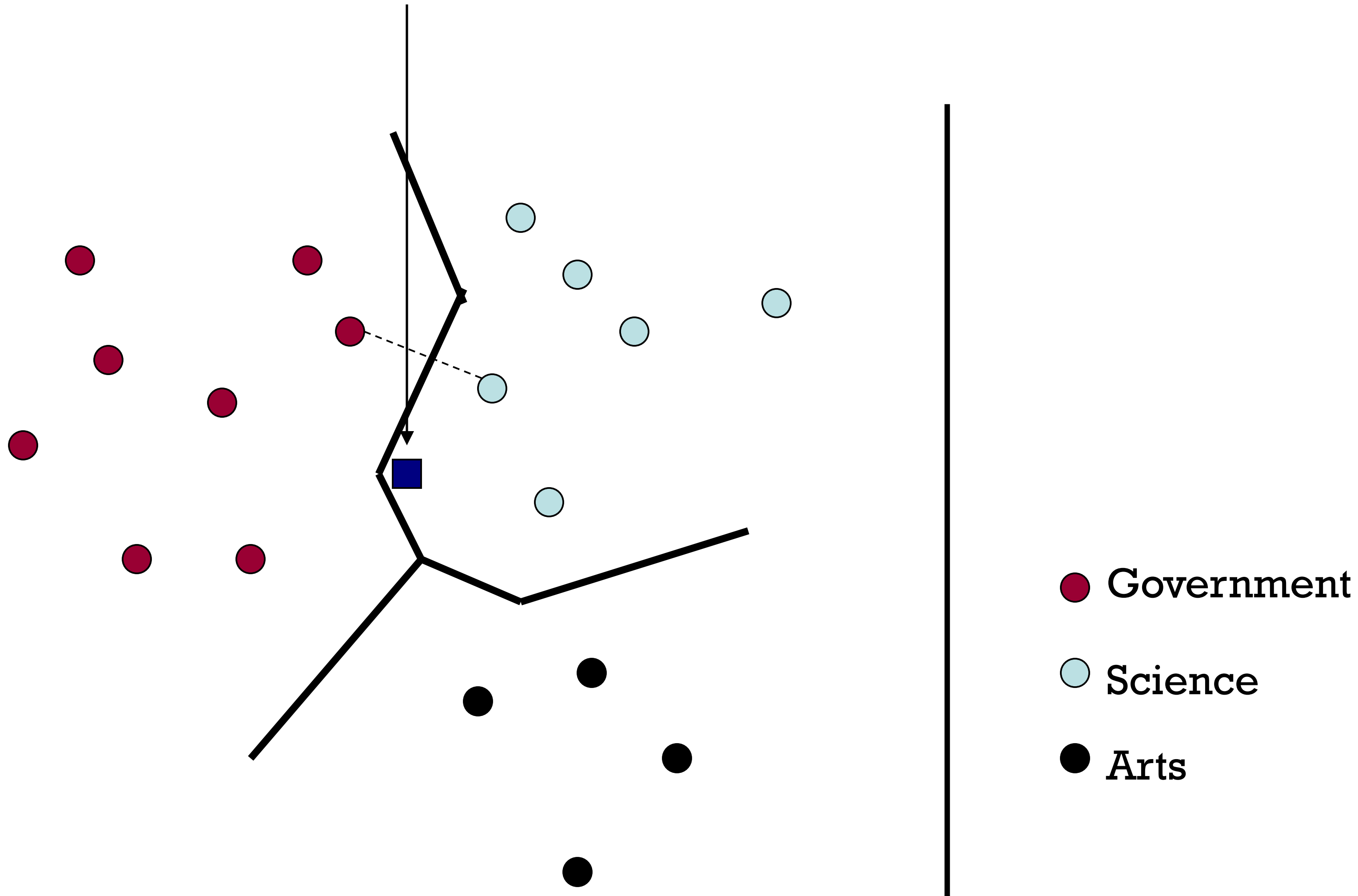
Nearest-Neighbor Classifiers



- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve

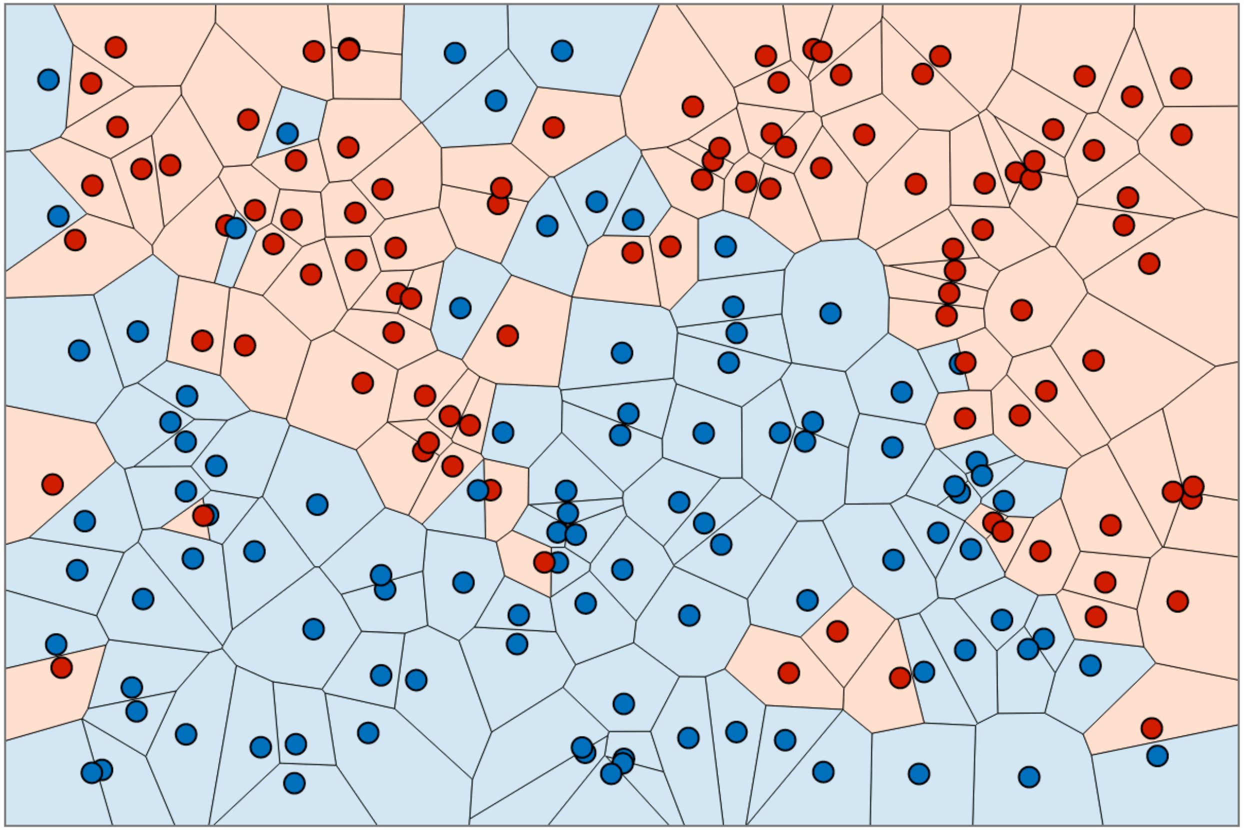
- To classify an unknown record:
  - Compute distance to other training records
  - Identify *k* nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Test Document = Science



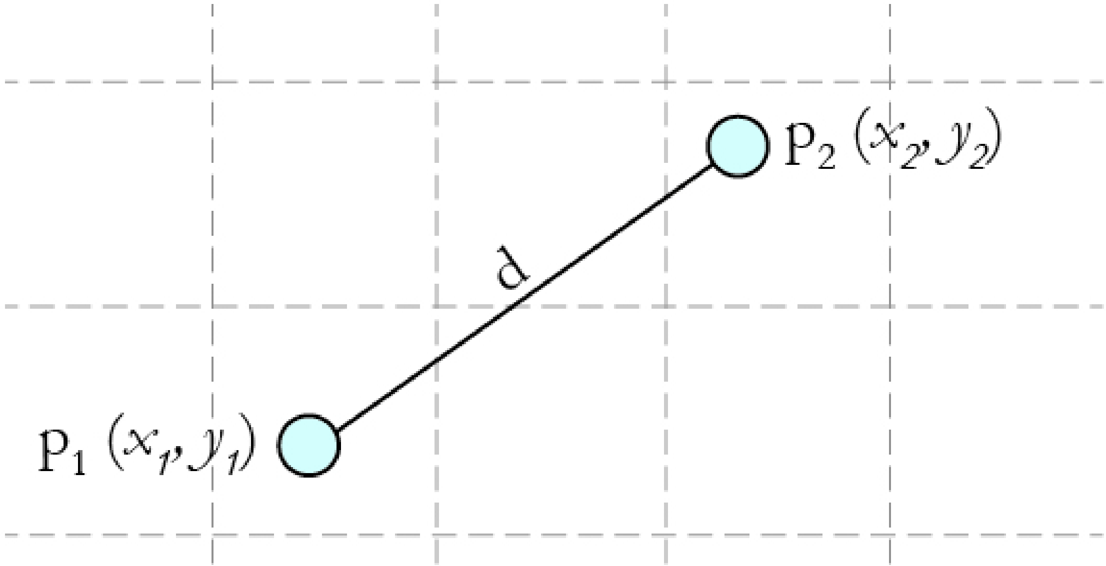
Voronoi diagram

Decision Boundary



Voronoi diagram

Euclidean Distance



$$\text{Euclidean distance (d)} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$\mathrm{d}(\mathbf{p}, \mathbf{q}) = \mathrm{d}(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

Parameters

- Distance measure
  - Not all distances are Euclidean.
  - Notion of similarity and distance.
- Value of K
  - Some “K”s are better than other.
  - Too small? Too large?
- Voting mechanism
  - Fusing results beyond a majority Vote.
  - Intuition: “far” imply less importance?
- Memory and Indexing (implimentation)

Finding the “best” K?

- Determined experimentally
- Start with k=1 and use a different “test set” to validate the error rate of the classifier
- Repeat with k=k+2
- Choose the value of k for which the error rate is minimum

- Note: k is often odd number to avoid ties

Is this simple algorithm of any use?

Example: Digit Recognition

- Yann LeCunn – MNIST Digit Recognition
  - Handwritten digits
  - 28x28 pixel images: d = 784
  - 60,000 training samples
  - 10,000 test samples
- Nearest neighbour is competitive

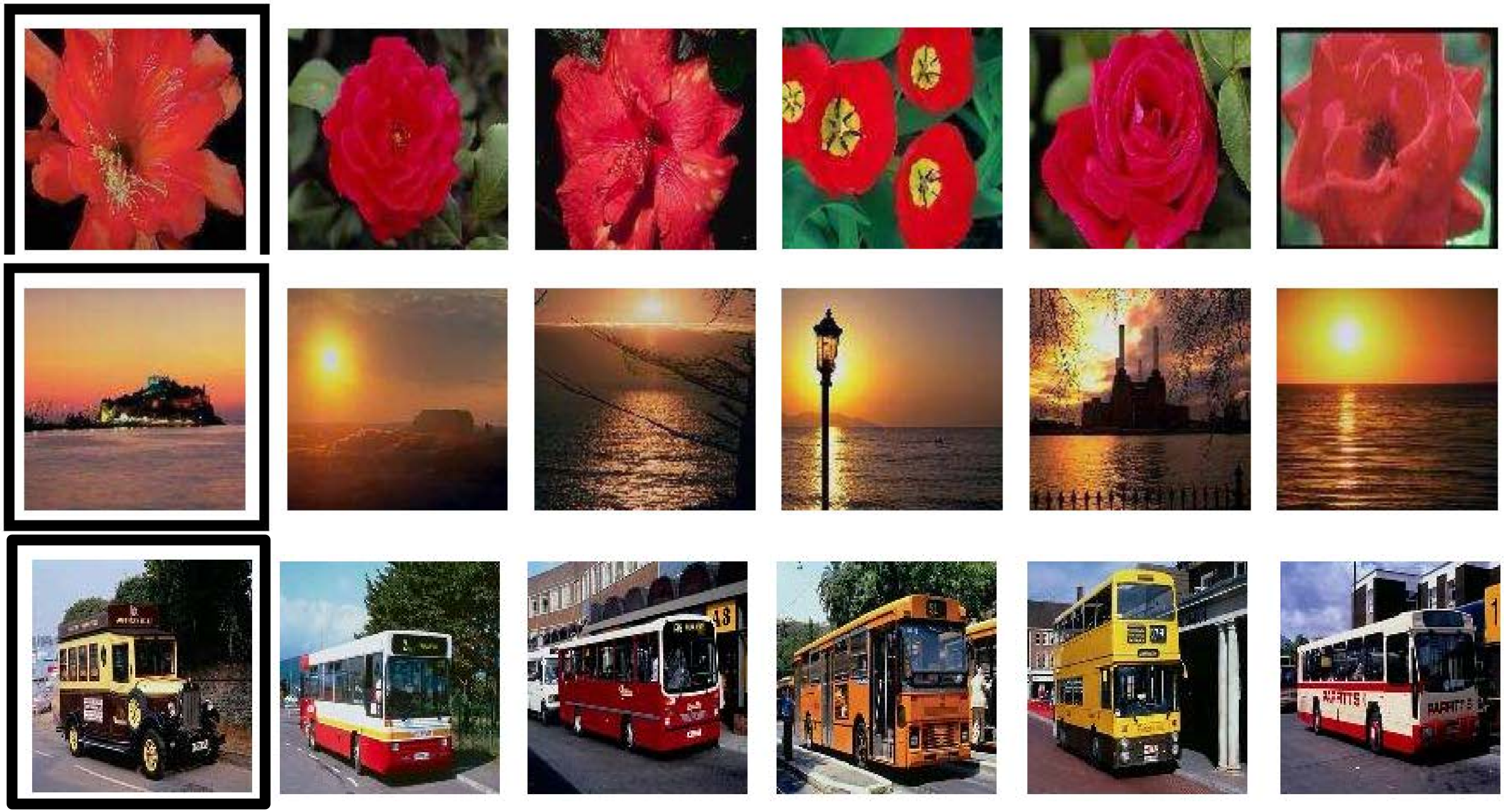
	Test Error Rate (%)
Linear classifier (1-layer NN)	12.0
K-nearest-neighbors, Euclidean	5.0
K-nearest-neighbors, Euclidean, deskewed	2.4
K-NN, Tangent Distance, 16x16	1.1
K-NN, shape context matching	0.67
1000 RBF + linear classifier	3.6
SVM deg 4 polynomial	1.1
2-layer NN, 300 hidden units	4.7
2-layer NN, 300 HU, [deskewing]	1.6
LeNet-5, [distortions]	0.8
Boosted LeNet-4, [distortions]	0.7

UNITED STATES POSTAL SERVICE ®

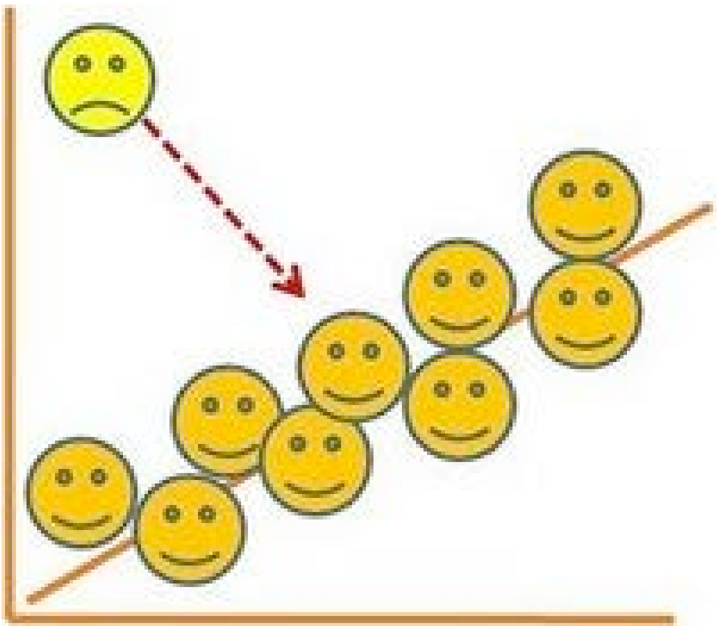
Still competitive for many problems!!

Problem of Retrieval

Input: query q; Outout: Ranked list
Given a Query "q", Sort and rank the database entries based of distance to the query d(q,x).



Outlier



If a sample is not in the "K Neighbours" of at least T samples, it is an outlier.

Advantages

- KNN is Simple
- KNNs are good if there are many many classes.
- Learn complex functions/decision boundaries (in fact piece-wise linear)
  - We will see linear methods soon, and also will see the need of nonlinearity
- No loss in information (everything is remembered). Fast to Train (no training?).

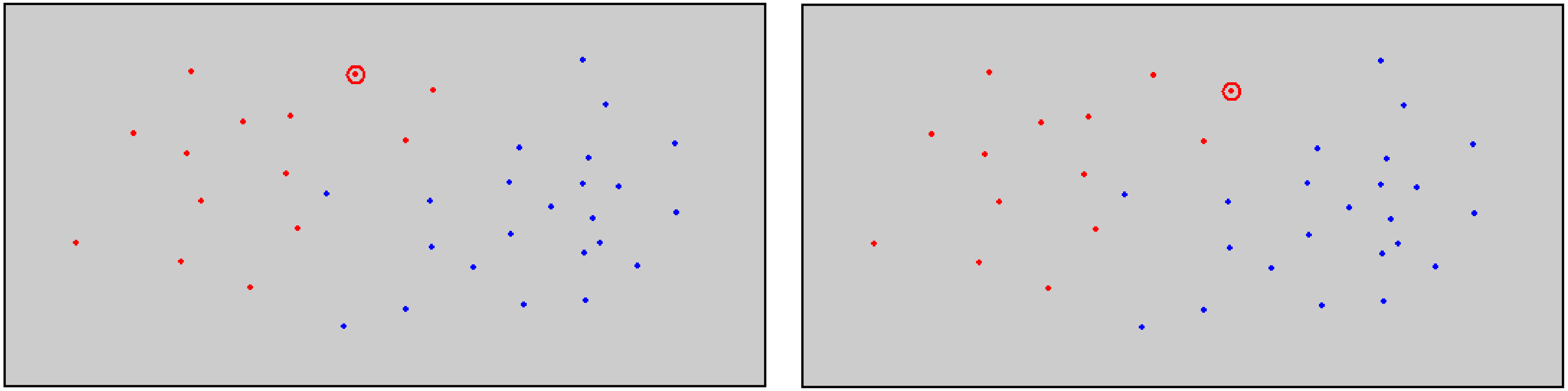
Disadvantages

- High cost for classification/inference
  - Work “only” at the inference/test time
  - Carry examples (implication of memory)
- Uses all features/attributes:
  - Do not learn which features are important
  - Can be fooled easily with irrelevant features

Condensing

- Condensed Nearest Neighbour (CNN) Hart 1968
  - Incremental
  - Order dependent
  - Neither minimal nor decision boundary consistent
  - $O(n^3)$ for brute-force method
  - Can follow up with reduced NN [Gates72]
    - Remove a sample if doing so does not cause any incorrect classifications

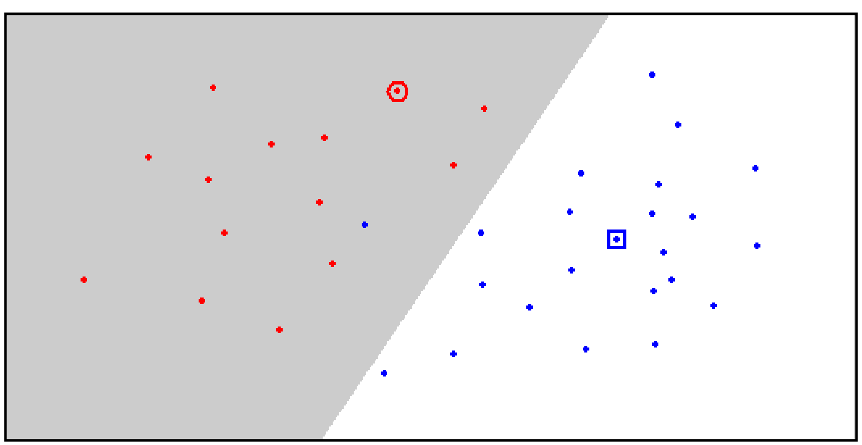
1. Initialize subset with a single training example
2. Classify all remaining samples using the subset, and transfer any incorrectly classified samples to the subset
3. Return to 2 until no transfers occurred or the subset is full

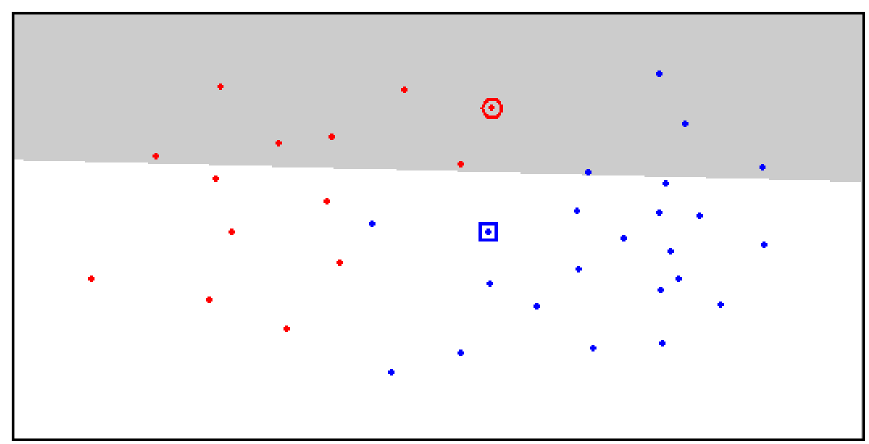


Condensing

- Condensed Nearest Neighbour (CNN) Hart 1968
  - Incremental
  - Order dependent
  - Neither minimal nor decision boundary consistent
  - $O(n^3)$ for brute-force method
  - Can follow up with reduced NN [Gates72]
    - Remove a sample if doing so does not cause any incorrect classifications

1. Initialize subset with a single training example
2. Classify all remaining samples using the subset, and transfer any incorrectly classified samples to the subset
3. Return to 2 until no transfers occurred or the subset is full

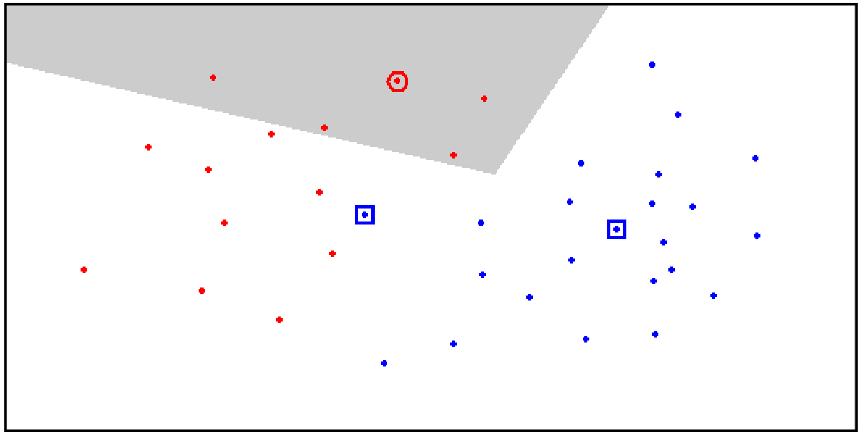


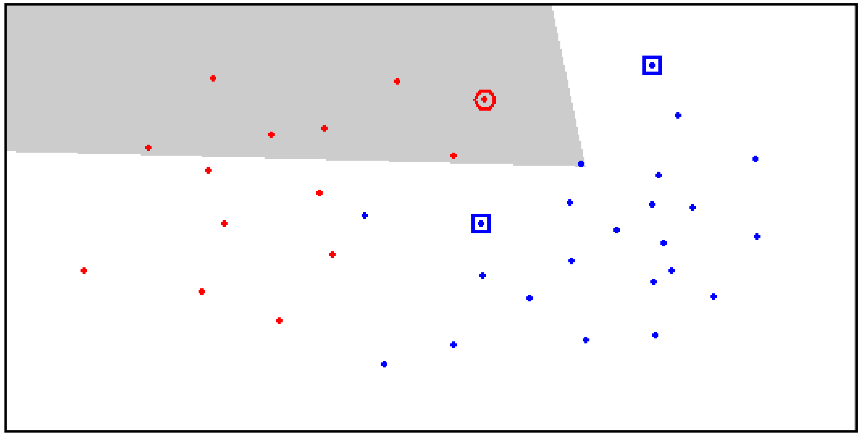


Condensing

- Condensed Nearest Neighbour (CNN) Hart 1968
  - Incremental
  - Order dependent
  - Neither minimal nor decision boundary consistent
  - $O(n^3)$ for brute-force method
  - Can follow up with reduced NN [Gates72]
    - Remove a sample if doing so does not cause any incorrect classifications

1. Initialize subset with a single training example
2. Classify all remaining samples using the subset, and transfer any incorrectly classified samples to the subset
3. Return to 2 until no transfers occurred or the subset is full

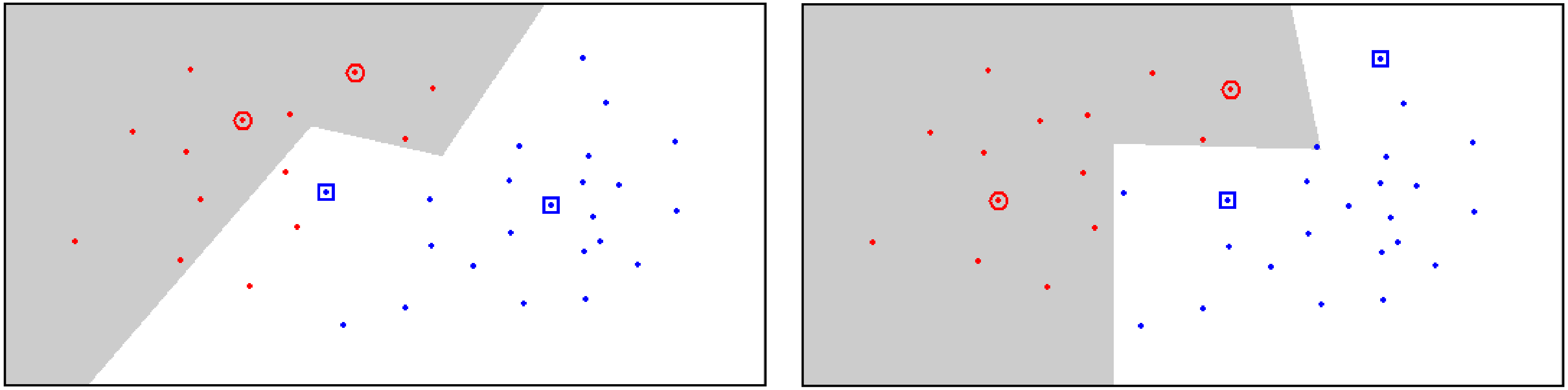




Condensing

- Condensed Nearest Neighbour (CNN) Hart 1968
  - Incremental
  - Order dependent
  - Neither minimal nor decision boundary consistent
  - $O(n^3)$ for brute-force method
  - Can follow up with reduced NN [Gates72]
    - Remove a sample if doing so does not cause any incorrect classifications

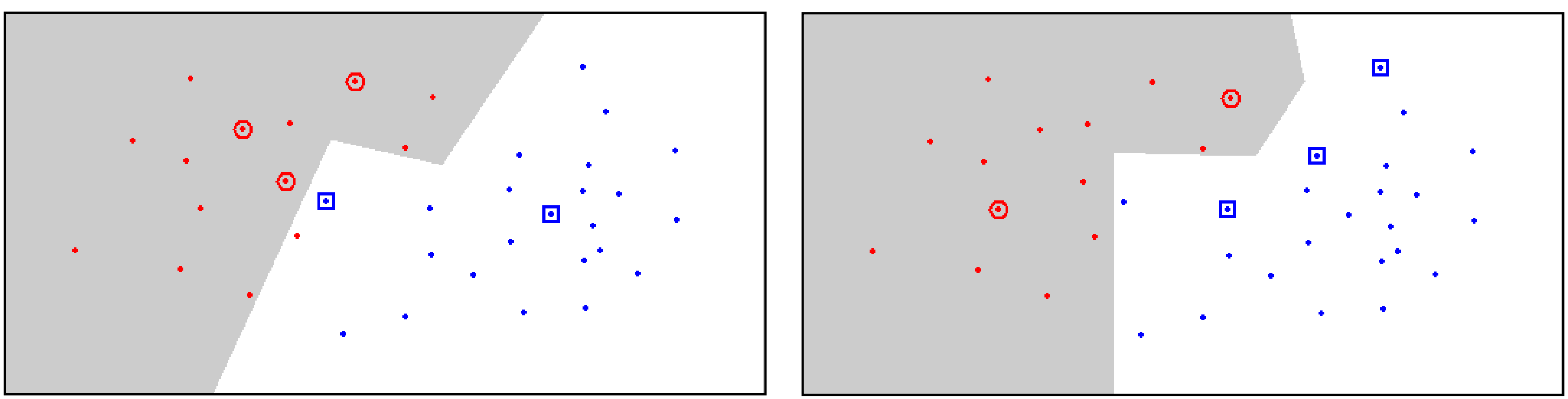
1. Initialize subset with a single training example
2. Classify all remaining samples using the subset, and transfer any incorrectly classified samples to the subset
3. Return to 2 until no transfers occurred or the subset is full



Condensing

- Condensed Nearest Neighbour (CNN) Hart 1968
  - Incremental
  - Order dependent
  - Neither minimal nor decision boundary consistent
  - $O(n^3)$ for brute-force method
  - Can follow up with reduced NN [Gates72]
    - Remove a sample if doing so does not cause any incorrect classifications

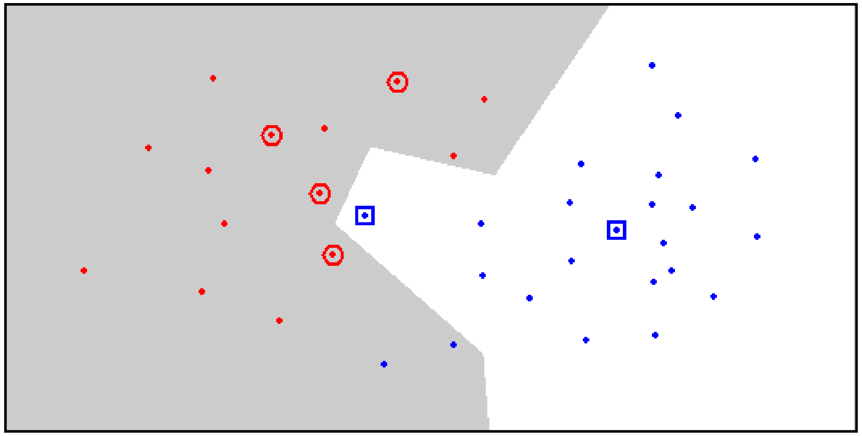
1. Initialize subset with a single training example
2. Classify all remaining samples using the subset, and transfer any incorrectly classified samples to the subset
3. Return to 2 until no transfers occurred or the subset is full

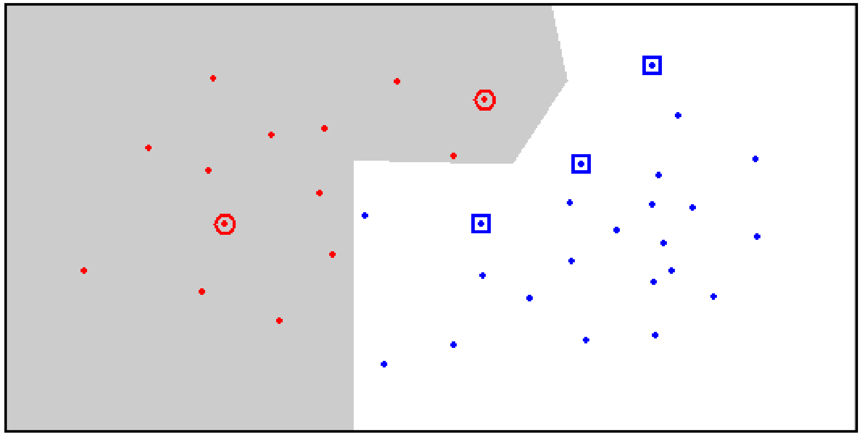


Condensing

- Condensed Nearest Neighbour (CNN) Hart 1968
  - Incremental
  - Order dependent
  - Neither minimal nor decision boundary consistent
  - $O(n^3)$ for brute-force method
  - Can follow up with reduced NN [Gates72]
    - Remove a sample if doing so does not cause any incorrect classifications

1. Initialize subset with a single training example
2. Classify all remaining samples using the subset, and transfer any incorrectly classified samples to the subset
3. Return to 2 until no transfers occurred or the subset is full

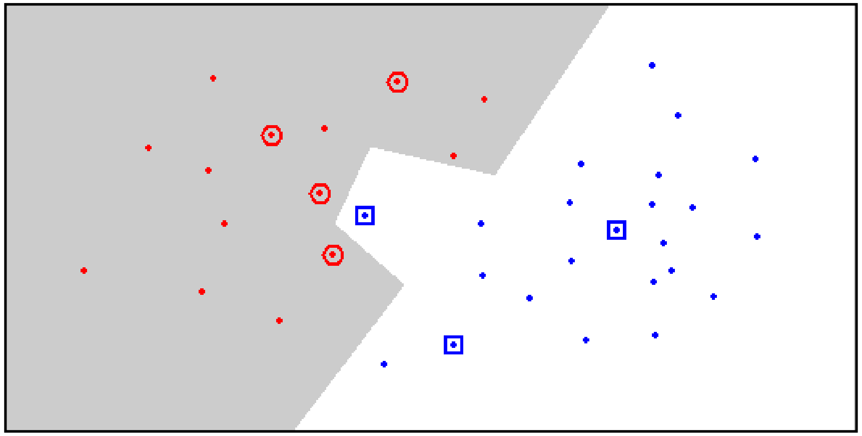


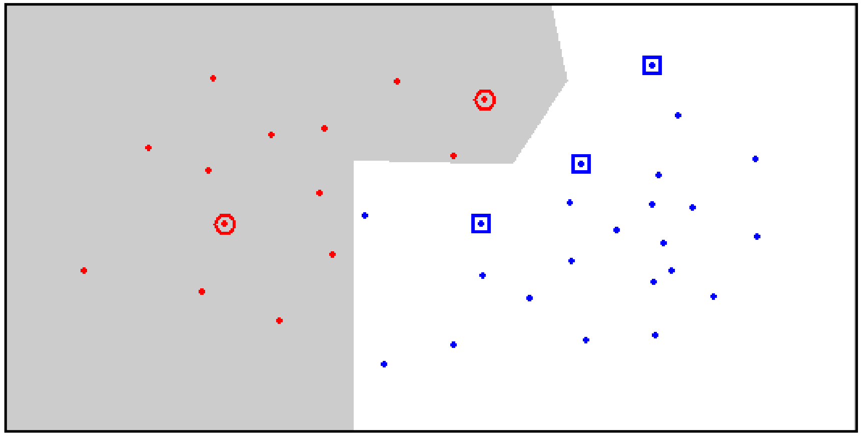


Condensing

- Condensed Nearest Neighbour (CNN) Hart 1968
  - Incremental
  - Order dependent
  - Neither minimal nor decision boundary consistent
  - $O(n^3)$ for brute-force method
  - Can follow up with reduced NN [Gates72]
    - Remove a sample if doing so does not cause any incorrect classifications

1. Initialize subset with a single training example
2. Classify all remaining samples using the subset, and transfer any incorrectly classified samples to the subset
3. Return to 2 until no transfers occurred or the subset is full





Nearest Neighbour Learning

- A popular learning perspective:
  - “Instance based learning”
- Lazy Learners
  - K NN and related methods
- Eager Learning Methods
  - Neural networks
  - Decision Trees
  - Support Vector Machines

Summary

- Nearest Neighbour Methods:
  - Simple methods. Intuitive.
- Many Applications
  - Classification
  - Nearest Neighbour Graphs
  - Retrieval
  - Recommendations
- Intuitive in nature.

Thanks!!

Questions?