Lecture 3: Nearest Neighbor Classification.

Classification: Given labelled data:

$$(x_i, y_i) \qquad i = 1, \ldots, n$$

x_i ↓ feature vectors

y_i ↓ labels (discrete)

Design a rule to predict y values for unseen x.

New (unseen) x (Called Test Data) ↓

Input examples (x_i, y_i) ↦ Classification Algorithm ↦ Prediction Rule. ↓ Label y (for x)

(called Training Data / Training Set)

Performance Measures:

1. Training error:

$$= \frac{\#\text{mistakes of the rule on training set}}{\text{Size of the training set.}}$$

2. Test error:

$$= \frac{\#\text{mistakes of the rule on test dataset}}{\text{Size of test set}}$$

* Training and test sets should be kept separate.
* Test error is a better measure than training error
* Training and test data should be "similar" in some sense.
* In fact, they are assumed to be drawn from same distribution.

Nearest Neighbor Classifier:

Given labelled examples (training data)

$(x_1, y_1), \ldots, (x_n, y_n)$

and a test example x.

Prediction Rule:

* Find the training data point x_j s.t. distance between x and x_j is minimum. (If tied, break ties uniformly at random)
* Output y_j.

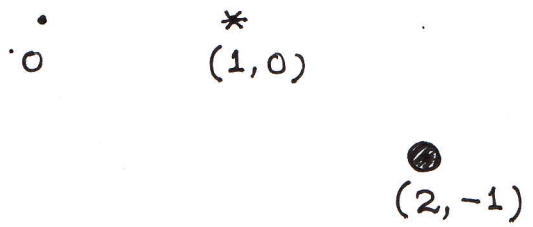
Example 1:

Training data:

$((1, 0), 0)$, $((1, 1), 0)$, $((2, -1), 1)$

Test points:

$(0, 0)$, $(2, 1)$, $(1.5, -0.5)$



- dist $((0,0), (1,0)) = 1$
  dist $((0,0), (1,1)) = \sqrt{2}$
  dist $((0,0), (2,-1)) = \sqrt{5}$

  so: closest point to $(0,0)$ = $(1,0)$, Label = 0

- dist ((2, 1), (1, 0)) = $\sqrt{2}$

  dist ((2, 1), (1, 1)) = 1

  dist ((2, 1), (2, −1)) = 2.

  closest : $(1, \frac{1}{\bullet})$

  output: $y = 0$

- dist ((1·5, −0·5), (1, 0)) = $\frac{1}{\sqrt{2}}$

  dist ((1·5, −0·5), (1, 1)) = $\sqrt{\frac{5}{2}}$

  dist ((1·5, −0·5), (2, −1)) = $\frac{1}{\sqrt{2}}$

  closest: (1, 0), (2, −1)

  Break ties at random

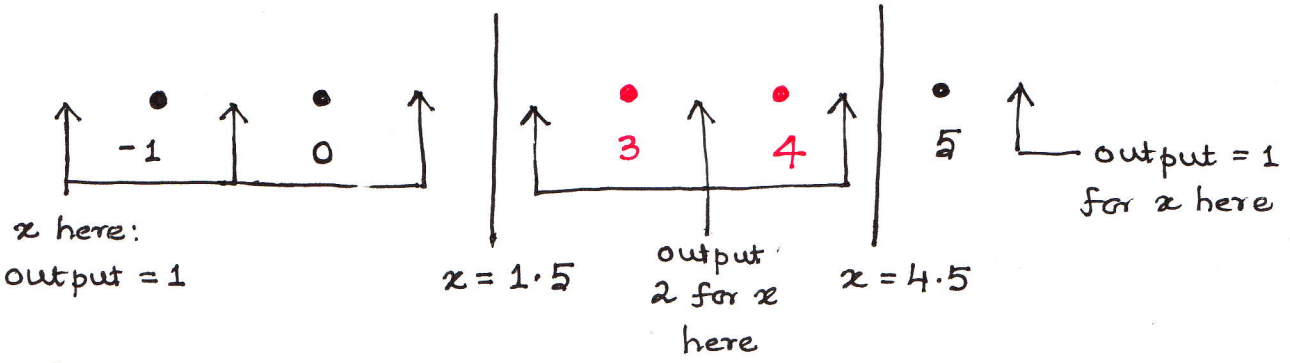
  (report $y = 0$ w.p. 1/2

  $y = 1$ w.p. 1/2)

Example 2:

Training data: (−1, 1), (0, 1), (3, 2), (4, 2), (5, 1)

x is a scalar.



Decision Boundary : Boundary between regions of different classes.

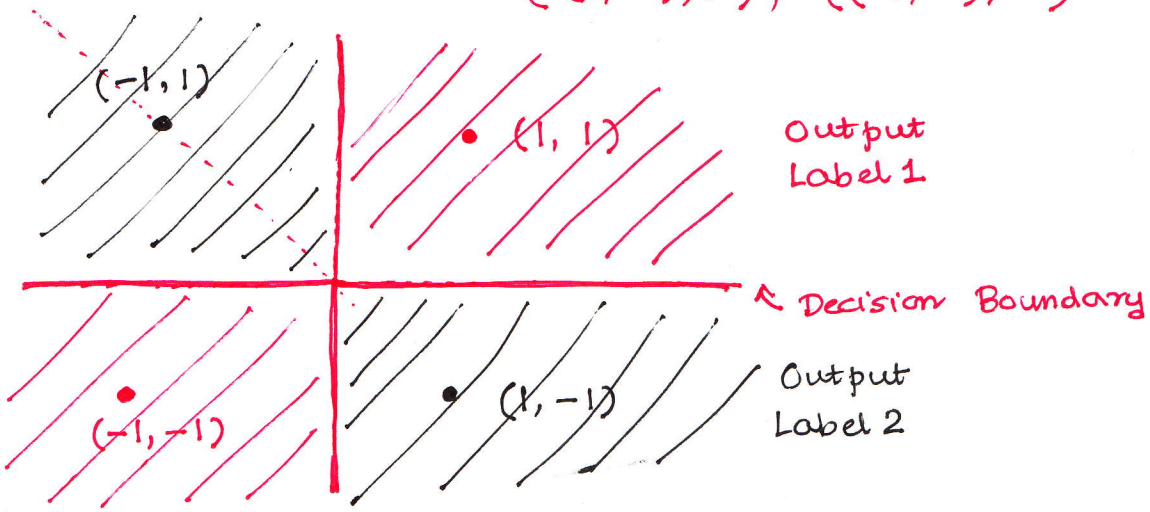
The output changes at the decision boundary.

NOTE: Decision boundary is a general concept,

applies to any classifier, not just NN.

Example 3:

Training data: $((1,1), 1)$, $((-1,-1), 1)$, $((1,-1), 2)$, $((-1,1), 2)$

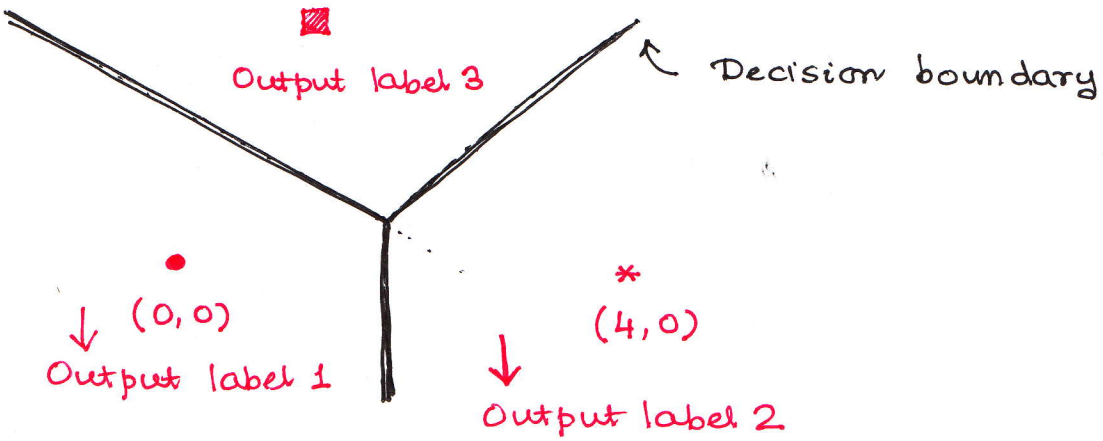


- $\|x - (1,1)\| \leq \min[\, \|x-(1,-1)\|, \|x-(-1,-1)\|, \|x-(-1,1)\| \,]$

(Equation represents all vectors x which are closer to $(1,1)$ than any other data point.)

Example 4:

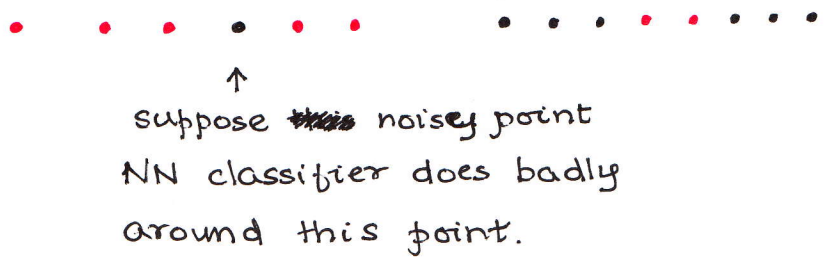
Training data: $((0,0), 1)$, $((4,0), 2)$, $((1,3), 3)$



When does NN work well or not?

- works well away from decision boundary
- not so well at the boundary
- also does not work well when data is noisy.

eg:



To make it more robust, k-NN classifier.

The k-Nearest Neighbor Classifier:

Given labelled examples (training data)

$(x_1, y_1), .., (x_n, y_n)$

and a test example x,

Prediction Rule:

1. Find $j_1, .., j_k$, the indices of the k points closest to x in the training data.
2. Output the majority of the labels $y_{j_1}, y_{j_2}, .., y_{j_k}$.

   [Majority label ≡ one that occurs most often]

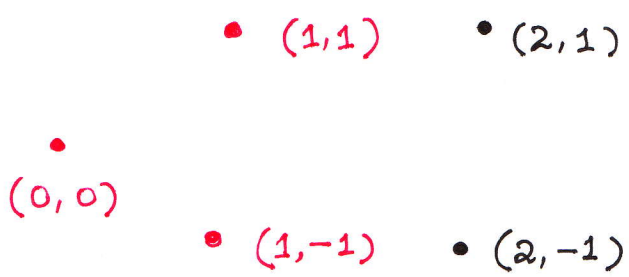
   If there is a tie, resolve uniformly at random.

Example 1: 3-NN

Training data:

((0, 0), 0) ((1, 1), 0) ((1, -1), 0)

((2, 1), 1) ((2, -1), 1)



Test points: (1, 0).

dist ((1, 0), (0, 0)) = 1 dist ((1, 0), (2, 1)) = $\sqrt{2}$

dist ((1, 0), (1, 1)) = 1 dist ((1, 0), (2, -1)) = $\sqrt{2}$

dist ((1, 0), (1, -1)) = 1

closest 3 points: (0, 0), (1, 1), (1, -1)

Their labels : 0 0 0

So output = 0.

Test point: (2, 0.5)

dist ((2, 0.5), (0, 0)) = $\frac{\sqrt{17}}{2}$ dist ((2, 0.5), (2, 1)) = $\frac{1}{2}$

dist ((2, 0.5), (1, 1)) = $\frac{\sqrt{5}}{2}$ dist ((2, 0.5), (2, -1)) = $\frac{5}{2}$

dist ((2, 0.5), (1, -1)) = $\frac{\sqrt{13}}{2}$

Closest 3 points: (2, 1), (1, 1), (2, -1)

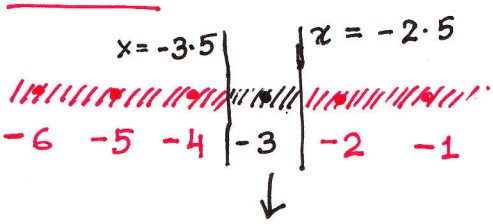
Labels: 1 0 1

Majority: 1 = output label.

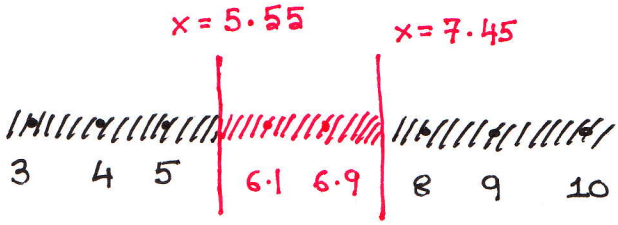
Example 2:

Suppose the points at -3 and 6.1 and 6.9 are noisy.

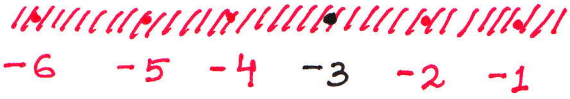
1-NN:



Output label is Black here

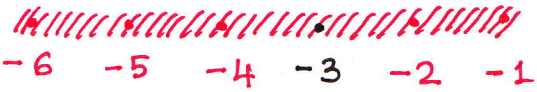


3-NN:



Output label is red on this entire region

5-NN

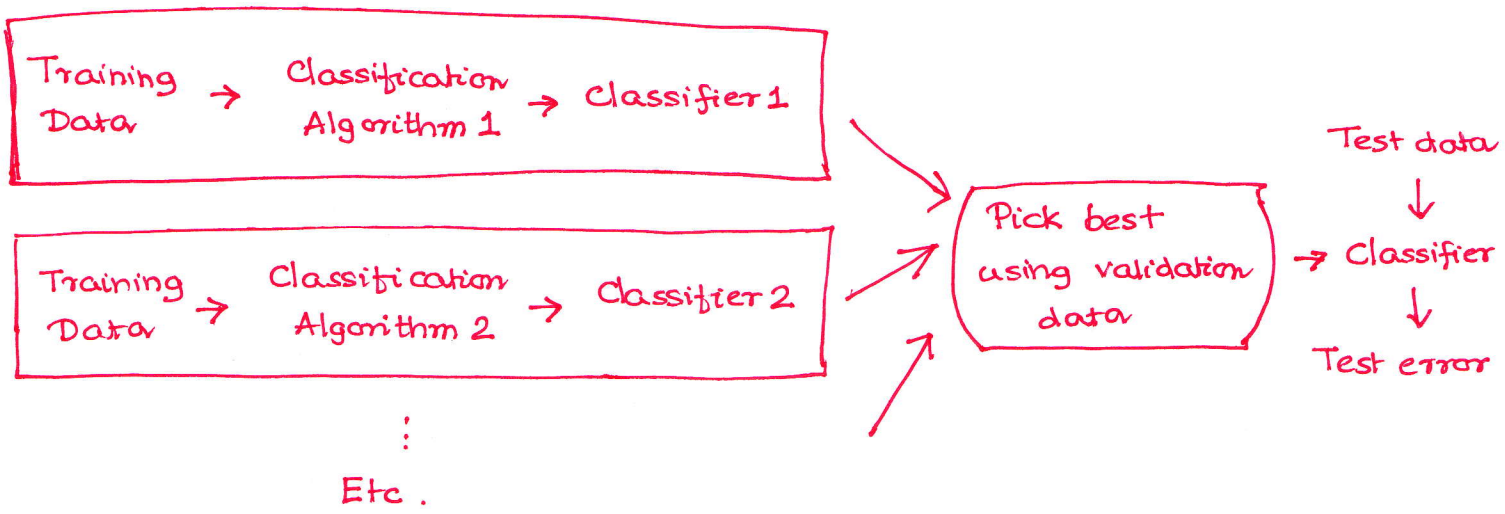


Output label red in this region

Output label black ein this entire region.

How to Choose k? Through Validation

1. Split data into training set and validation set.
2. Train classifier on training set for $k = 1, 3, 5, \ldots$.
3. Evaluate the error of each classifier trained on validation set and pick the one with the lowest error.



Distance Measure: Most common is Euclidean distance. Others used too.

How to find NNs? In 1-d, binary search $O(\log n)$
Higher d, advanced data structures such as Locality Sensitive Hashing.

Advantages + Disadvantages:

- Simple, flexible, easy to implement
- Classification time is high, Space requirement high, Doesn't work very well in high dimensions.