Foundations of Data Science & Analytics: Nearest Neighbor

Ezgi Siir Kibris

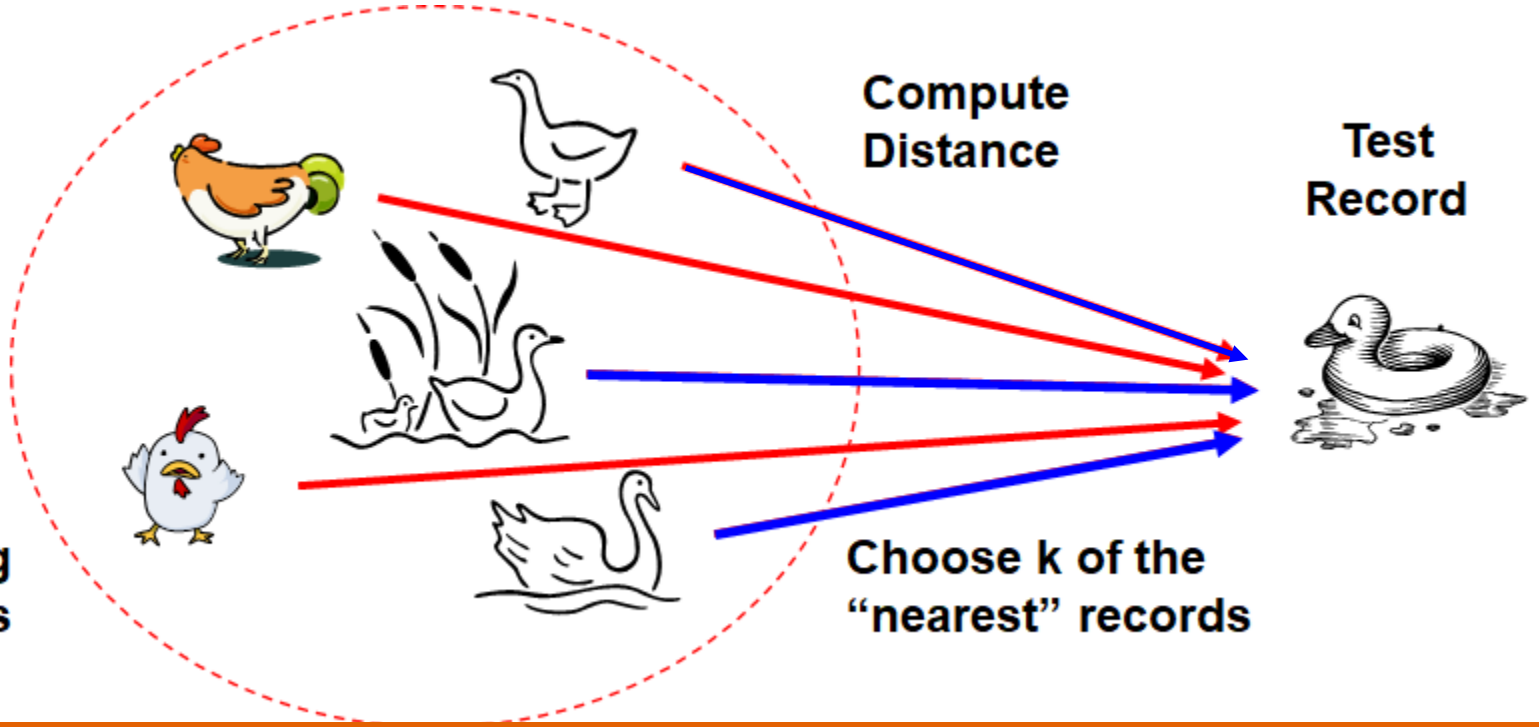
Introduction to Data Mining, 2nd Edition
by
Tan, Steinbach, Karpatne, Kumar

Classification Techniques

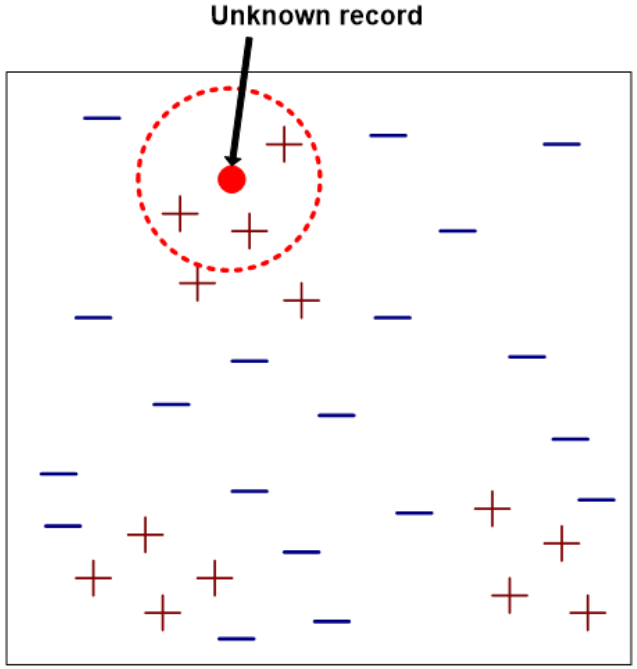
- **Base Classifiers**
  - Decision Tree based Methods
  - Rule-based Methods
  - **Instance-based Methods (Nearest-neighbor)**
  - Naïve Bayes
  - Support Vector Machines
  - Neural Networks and Deep Learning
- **Ensemble Classifiers**
  - Boosting, Bagging, Random Forests

Nearest Neighbors

If it walks like a duck, quacks like a duck, then it's probably a duck



Nearest Neighbors



Requires three things (inputs)

- The set of labeled 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)

Nearest Neighbors

Compute proximity between two points:

- Example: Euclidean distance

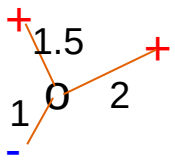
$$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_i (\boldsymbol{x_i} - \boldsymbol{y_i})^2}$$

Determine the class from nearest neighbor list

- Take the majority vote of class labels among the k-nearest neighbors
- Weight the vote according to distance
  - weight factor $w = 1/d^2$

Nearest Neighbors

- Weight the vote according to distance
  - weight factor $w = 1/d^2$



Vote:

+

-

Prediction Probability:

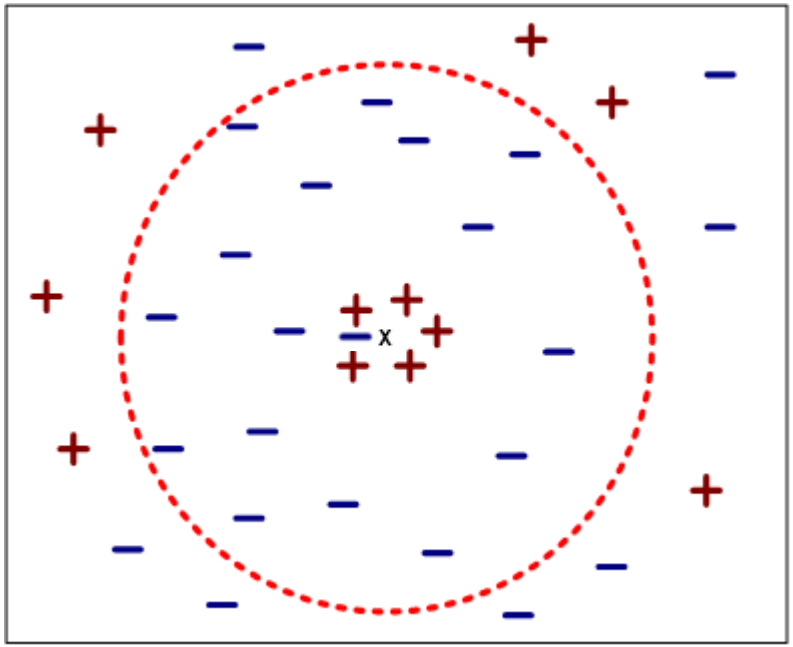
+

-

Choosing the value of k

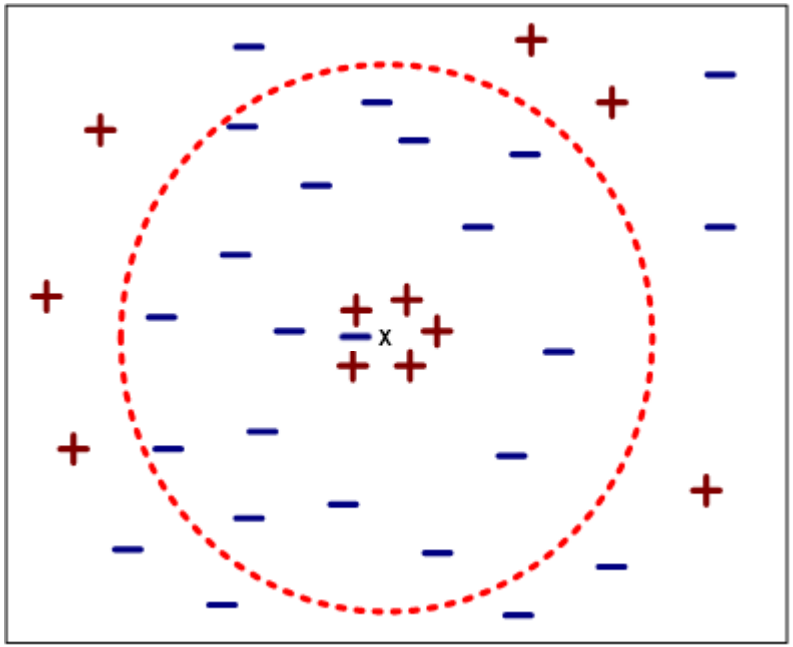
Euclidean distance
Majority vote

- K = 1?
- K = 3?
- K = 5?
- K = 11?



Choosing the value of k

- If k is too small, sensitive to noise points
- If k is too large, neighborhood may include points from other classes
- Common default k=5



Distance measures

Minkowski Distance:

$$dist(x, y) = (\sum_i |x_i - y_i|^p)^{\frac{1}{p}}$$

- Euclidean distance (p=2)
- Manhattan distance (p=1)

Distance measures

Data preprocessing is often required

- Features may have to be scaled to prevent distance measures from being dominated by one of the features

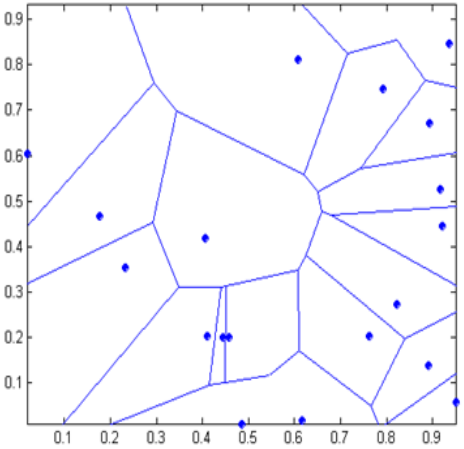
- Example:
  - Height of a person may vary from 1.5m to 1.9m
  - Weight of a person may vary from 90lb to 300lb
  - Income of a person may vary from $10K to $1M

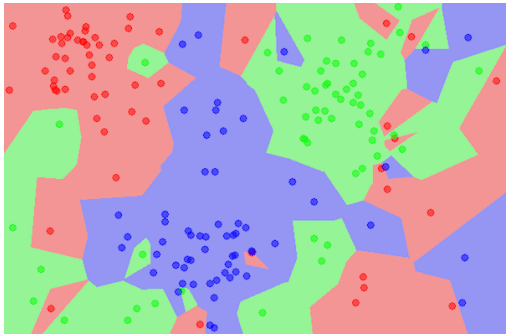
Nearest Neighbor

Nearest neighbor classifiers are local classifiers.

They can produce decision boundaries of arbitrary shapes.

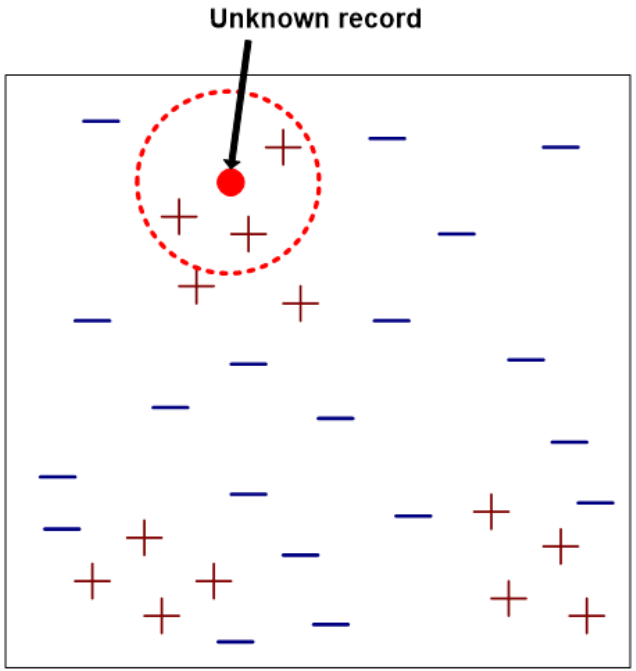
1-nn decision boundary is a Voronoi Diagram





Nearest Neighbor

- No training time
  - Just store all training data
- High computation in inference O(dN)
  - Sort by distances from the test data point to all training data
  - Prediction decided by the k nearest neighbors



1. Calculate distance

```
# calculate the Euclidean distance between two vectors
def euclidean_distance(row1, row2):
 distance = 0.0
 for i in range(len(row1)-1):
 distance += (row1[i] - row2[i])**2
 return sqrt(distance)
```

2. Get nearest neighbors

```python
# Locate the most similar neighbors
def get_neighbors(train, test_row, num_neighbors):
 distances = list()
 for train_row in train:
 dist = euclidean_distance(test_row, train_row)
 distances.append((train_row, dist))
 distances.sort(key=lambda tup: tup[1])
 neighbors = list()
 for i in range(num_neighbors):
 neighbors.append(distances[i][0])
 return neighbors
```

3. Prediction

```
# Make a classification prediction with neighbors
def predict_classification(train, test_row, num_neighbors):
 neighbors = get_neighbors(train, test_row, num_neighbors)
 output_values = [row[-1] for row in neighbors]
 prediction = max(set(output_values), key=output_values.count)
 return prediction
```

Assignment 6