k-Nearest Neighbor & Instance-based Learning

Artificial Intelligence

Ko, Youngjoong

Index

Overview

- Geometric View of data
  - Examples are points in a high-dimensional space

- Nearest Neighbor Model
  - Suppose you need to predict whether "Alice will like "Algorithms."
  - If we can try to find another student who is most similar to Alice in terms of favorite courses and he/she liked "Algorithms," then we might guess that Alice will as well.

Adapted from Hal Daumé III's **lecture notes**

From Data to Feature Vectors

- Feature Values
  - Binary Features
  - Real Value Features

- Single example as a vector in a high dimensional feature space
  - Take a data set and map each example to a feature vector through the following mapping:
    - Real-valued features get copied directly.
    - Binary features become 0 (false) or 1 (true)
    - Categorical features with V possible values get mapped to V-many binary indicator features

Adapted from Hal Daumé III's **lecture notes**

From Data to Feature Vectors

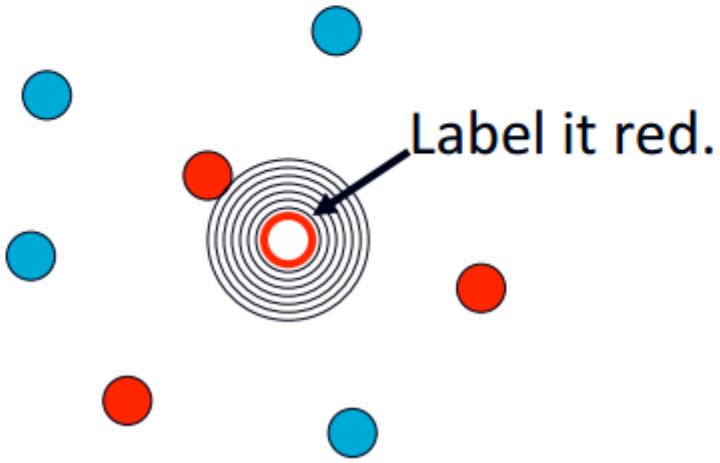
- Categorical Features
  - If our goal is to identify whether an object in an image is a tomato, blueberry, cucumber or cockroach.
  - We might want to know its color: Red, Blue, Green or Black?
- A solution is to turn a categorical feature that can take four different values into four binary features
  - Ex) Is it Red?, Is it Blue?, Is it Green?, Is it Black?

0	1	0	0

Adapted from Hal Daumé III's **lecture notes**

1-Nearest Neighbor

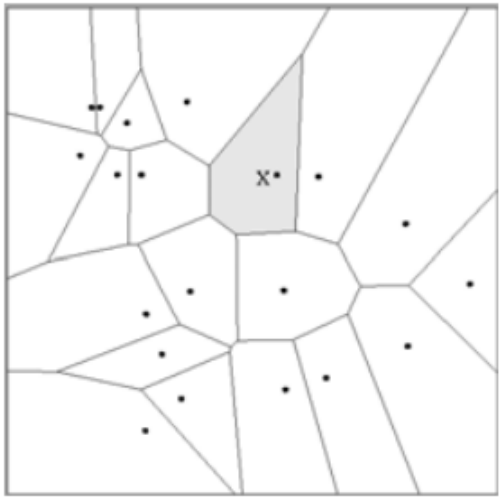
- One of the simplest of all machine learning classifiers
- Simple idea: label a new point the same as the closest known point



Slide Credit: Eric Eaton(adapted from "instance-based learning lecture by Andrew Moore, CMU"

1-Nearest Neighbor

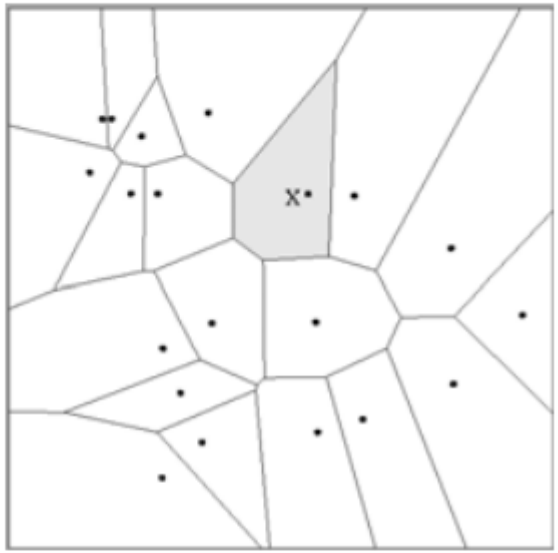
- A type of instance-based learning
  - Also known as "memory-based" learning
- Forms a Voronoi tessellation of the instance space



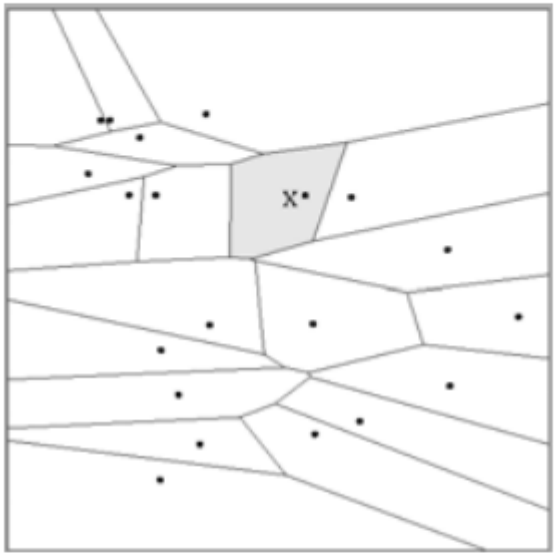
Slide Credit: Eric Eaton(adapted from "instance-based learning lecture by Andrew Moore, CMU"

Distance Metrics

- Different metrics can change the decision surface



Dist(**a**,**b**) =$(a_1 - b_1)^2 + (a_2 - b_2)^2$



Dist(**a**,**b**) =$(a_1 - b_1)^2 + (3a_2 - 3b_2)^2$

- Standard Euclidean distance metric:
  - Two-dimensional: Dist(a,b) = sqrt($(a_1 - b_1)^2 + (a_2 - b_2)^2$)
  - Multivariate: Dist(a,b) = sqrt($\sum (a_i - b_i)^2$)

Slide Credit: Eric Eaton(adapted from "instance-based learning lecture by Andrew Moore, CMU"

Instance-Based Learning

- Four Aspects

1. A distance metric
2. How many nearby neighbors to look at?
3. A weighting function (optional)
4. How to fit with the local points?

Slide Credit: Eric Eaton(adapted from "instance-based learning lecture by Andrew Moore, CMU"

Instance-Based Learning

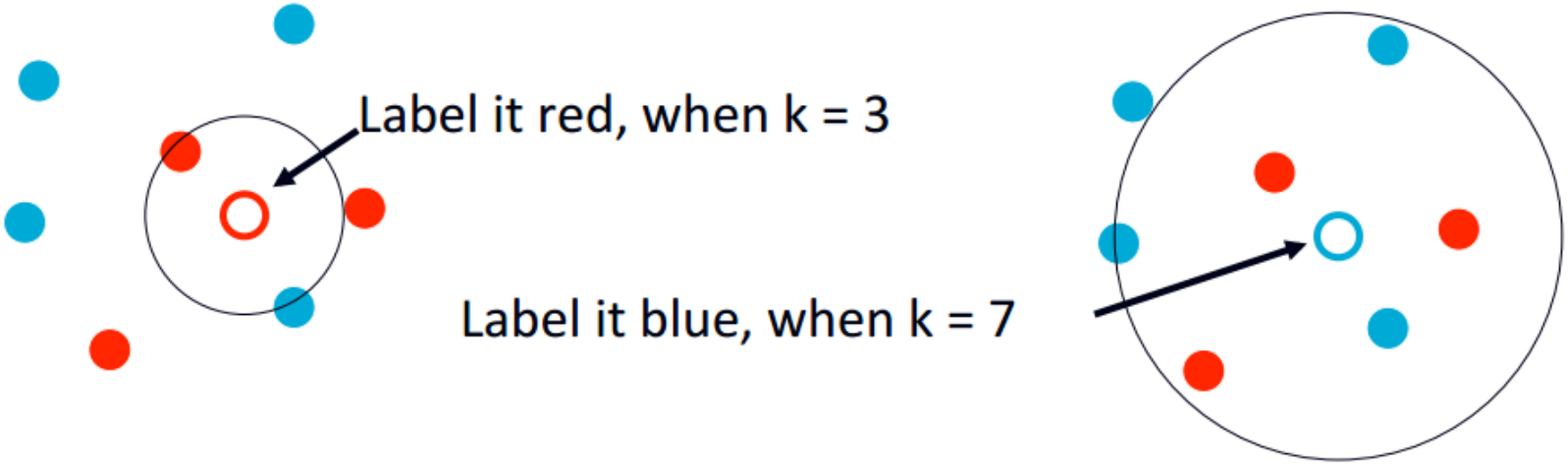
- 1-NN's Four Aspects as an Instance-Based Learner

1. A distance metric
   - *Euclidian*
2. How many nearby neighbors to look at?
   - *One*
3. A weighting function (optional)
   - *Unused*
4. How to fit with the local points?
   - *Just predict the same output as the nearest neighbor.*

Slide Credit: Eric Eaton(adapted from "instance-based learning lecture by Andrew Moore, CMU"

k-Nearest Neighbors (kNN)

- Generalizes 1-NN to smooth away noise in the labels
- A new point is now assigned the most frequent label of its *k* nearest neighbors



Slide Credit: Eric Eaton(adapted from "instance-based learning lecture by Andrew Moore, CMU"

k-Nearest Neighbors (kNN)

- Pseudo Code of kNN

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

```
S ← [ ]
for n = 1 to N do
    S ← S ⊕ ⟨d(x_n, x̂), n⟩          // store distance to training example n
end for
S ← SORT(S)                          // put lowest-distance objects first
ŷ ← 0
for k = 1 to K do
    ⟨dist, n⟩ ← S_k                  // n this is the kth closest data point
    ŷ ← ŷ + y_n                      // vote according to the label for the nth training point
end for
return SIGN(ŷ)                       // return +1 if ŷ > 0 and −1 if ŷ < 0
```

Pseudo Code from Hal Daumé III's **lecture notes**