TFIP-AI - Machine Learning

Unit 4 K-Nearest Neighbor (KNN) Algorithm

Part 1 Classification and Regression

Outline

- Instance-based learning
- KNN (K-Nearest Neighbors)
- Distance
- Feature scaling, standardization
- Curse of dimensionality

Instance-Based Learning

- Key idea
  - Just store all training examples $\langle x_i, f(x_i) \rangle$
  - When a new query instance x_q is encountered, a set of similar related instances is retrieved from memory and used to classify x_q, i.e. $\hat{f}(x_q)$

A formal introduction

- Nearest neighbor
  - Given query instance x_q, first locate nearest training example x_n, then estimate $\hat{f}\ (x_q) \leftarrow f\,(x_n)$

- k-Nearest neighbor
  - Given x_q, take vote among its k nearest neighbors, if discrete-valued target function
  - Take mean of f values of k nearest nbrs, if real-valued

$$\hat{f}\ (x_q) \leftarrow \frac{\sum_{i=1}^{k} f\,(x_i)}{k}$$

$d(p, q)$

- Vectors
  - $p = \{p_1, \dots, p_n\}, q = \{q_1, \dots, q_n\}$

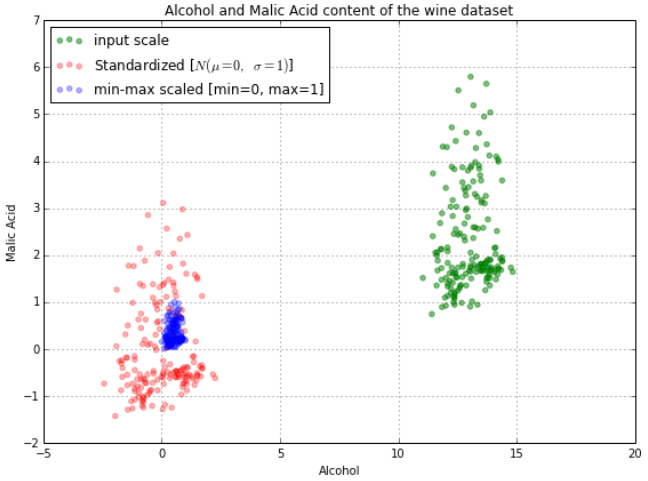
- Manhattan Distance

$$d(p, q) = \sum_{d=1}^{n} |p_d - q_d|$$

- Minkowski Distance

$$d(p, q) = (\sum_{d=1}^{n} |p_d - q_d|^k)^{1/k}$$

Feature Scaling



- Min-max normalization

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

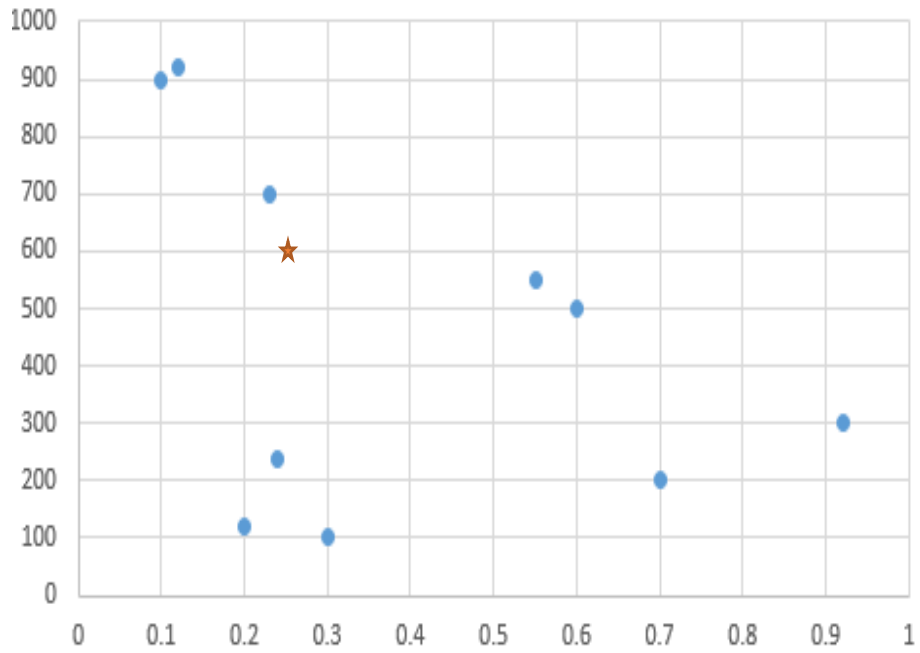
- Mean normalization

$$x' = \frac{x - avg(x)}{\max(x) - \min(x)}$$

Feature Scaling - Example

x1	x2	D(x_{new},X)
0.3	100	500
0.6	500	100
0.1	900	300
0.2	120	480
0.24	240	360
0.7	200	400
0.92	300	300
0.55	550	50
0.23	700	100
0.12	920	320

x_{new}=(0.25 600)



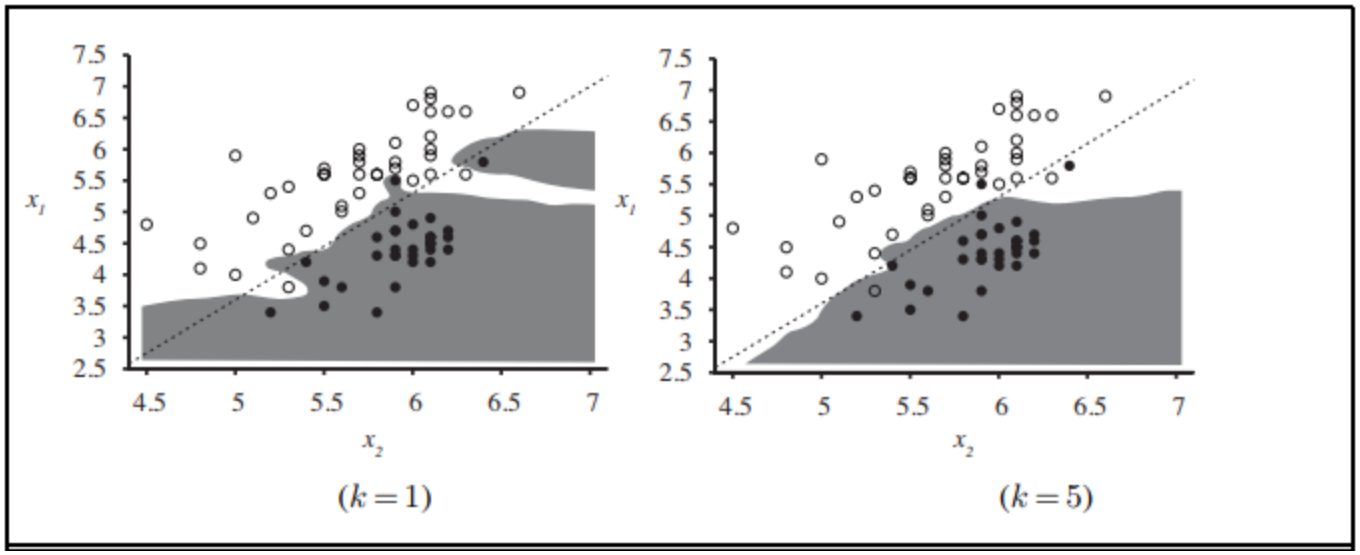
Feature Scaling - Example

x1	x2	Min-max – x1	Min-max – x2	D(x_{new},X)
0.3	100	0.24	0	0.61
0.6	500	0.61	0.49	0.38
0.1	900	0	0.98	0.44
0.2	120	0.12	0.02	0.60
0.24	240	0.17	0.17	0.45
0.7	200	0.73	0.12	0.69
0.92	300	1	0.24	0.83
0.55	550	0.55	0.54	0.30
0.23	700	0.16	0.73	0.15
0.12	920	0.02	1	0.45

$$\frac{x1 - 0.1}{0.92 - 0.1} \qquad \frac{x2 - 100}{920 - 100}$$

(0.25 600) →
(0.18 0.61)

Overfitting - KNN



overfitting

Overfitting problem goes away

Distance-Weighted k-NN

- Might want weight nearer neighbors more heavily

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i}$$

where

$$w_i \equiv \frac{1}{d(x_q, x_i)^2} \text{ or } \frac{1}{d(x_q, x_i)}$$

and $d(x_q, x_i)$ is distance between x_q and x_i

- It makes sense to use all training examples instead of just k

Distance-Weighted k-NN - example

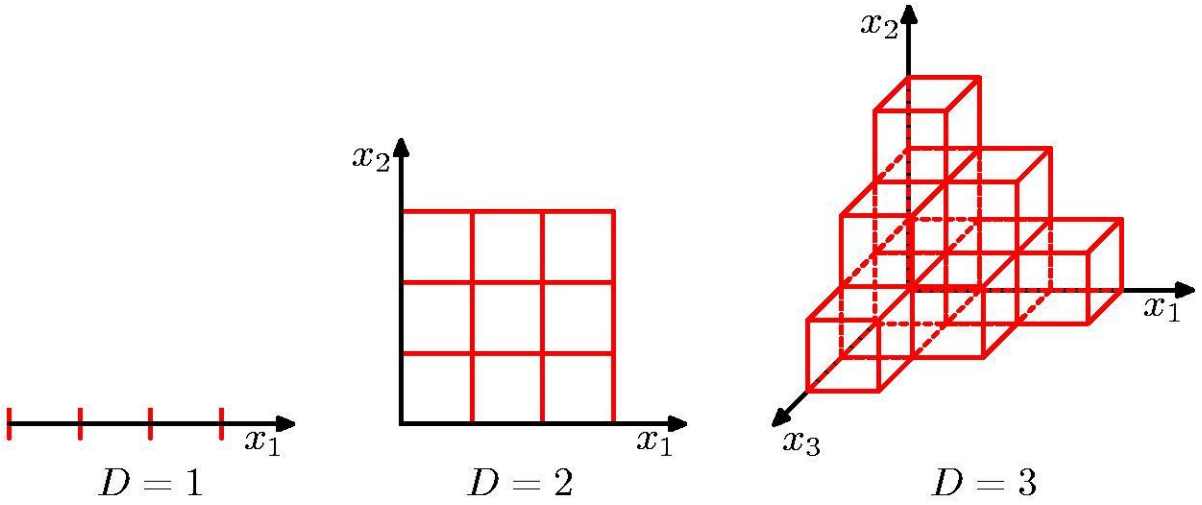
discrete-valued

- Target value of top k=4 nbrs: 0 0 1 1
- Distances
  - 0.15 0.12 0.78 0.10
- Decision making
  - Weights=1/distance
  - 0: 1/0.15+1/0.12=15
  - 1: 1/0.78+1/0.1=11.3

Real-valued

- Target value of top k=4 nbrs: 100 101 120 90
- Distances
  - 0.15 0.12 0.78 0.10
- Weights(1/distance^2)
  - 44, 69, 1.64, 100
- Price
  - price=96
  - $\frac{44*100+69*101+1.64*120+100*90}{44+69+1.64+100}$

Curse of Dimensionality



Not all intuitions developed in spaces of low dimensionality will generalize to spaces of many dimensions.

Volume of a sphere in a high dimensional space

- Consider a sphere of radius r=1 in a space D dimensions, what is the fraction of the volume of the sphere that lies between radius $r = 1 - \varepsilon$ and $r = 1$
- Volume of a sphere must scale as r^D
  - $V_D(r) = K_D \cdot r^D$
- The required fraction is
  - $\frac{V_D(1) - V_D(r-\varepsilon)}{V_D(1)} = \frac{K_D \cdot 1^D - k_D(1-\varepsilon)^D}{K_D \cdot 1^D} = 1 - (1-\varepsilon)^D$

Thus, in spaces of high dimensionality, most of the volume of a sphere is concentrated in a thin shell near the surface!