Instance Based Learning

Bùi Tiến Lên

01/09/2019

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Contents

Notation

symbol	meaning
$a, b, c, N \ldots$	scalar number
$\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{y} \ldots$	column vector
$\boldsymbol{X}, \boldsymbol{Y} \ldots$	matrix
$\mathbb{R}$	set of real numbers
$\mathbb{Z}$	set of integer numbers
$\mathbb{N}$	set of natural numbers
$\mathbb{R}^D$	set of vectors
$\mathcal{X}, \mathcal{Y}, \ldots$	set
$\mathcal{A}$	algorithm

operator	meaning
$\boldsymbol{w}^\top$	transpose
$\boldsymbol{X}\boldsymbol{Y}$	matrix multiplication
$\boldsymbol{X}^{-1}$	inverse

Parametric vs Non-parametric Models

Parametric Models

- In the models that we have seen, we select a hypothesis space $\mathcal{H}$ and adjust a *fixed set of parameters* $\boldsymbol{w}$ with the training data $\mathcal{D}$
- We assume that the parameters $\boldsymbol{w}$ summarize the training data $\mathcal{D}$ and we can forget about it

Non-parametric Models

- A non parametric model is one that can not be characterized by a fixed set of parameters
- A family of non parametric models is **Instance Based Learning**

k-Nearest Neighbor

k-Nearest Neighbor

Key idea: just store all training examples $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}$

Nearest neighbor

- Given query instance $\boldsymbol{x}_q$, first locate *the nearest neighbhor* $\boldsymbol{x}^{(1)}$, then estimate

$$h(\boldsymbol{x}_q) = y^{(1)} \tag{1}$$

k-**Nearest neighbor** (k is a *hyper-parameter*)

- Given $\boldsymbol{x}_q$, take vote among its k *nearest neighbors* $\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(k)}\}$ (if discrete-valued target function)

$$h(\boldsymbol{x}_q) = vote\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(k)}\} \tag{2}$$

- Take mean of the values of k *nearest neighbors* (if real-valued target function)

$$h(\boldsymbol{x}_q) = \frac{\sum_{i=1}^{k} y^{(i)}}{k} \tag{3}$$

When To Consider Nearest Neighbor

- Instances map to points in $\mathbb{R}^D$
- Less than 20 attributes per instance
- Lots of training data $\mathcal{D}$

Advantages

- No training, just store data
- Learn complex target functions
- Don't lose information

Disadvantages

- Slow at query time
- Easily fooled by irrelevant attributes

Distance

Some common distances in space $\mathbb{R}^D$

- The Minkowski distance of order $p > 0$

$$d(\boldsymbol{x}, \boldsymbol{y}) = L_p(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{D} |x_i - y_i|^p \right)^{1/p} \tag{4}$$

- Euclidean distance

$$d(\boldsymbol{x}, \boldsymbol{y}) = L_2(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_{i=1}^{D} (x_i - y_i)^2} \tag{5}$$

Distance (cont.)

- Manhattan distance

$$d(\boldsymbol{x}, \boldsymbol{y}) = L_1(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{D} |x_i - y_i| \tag{6}$$

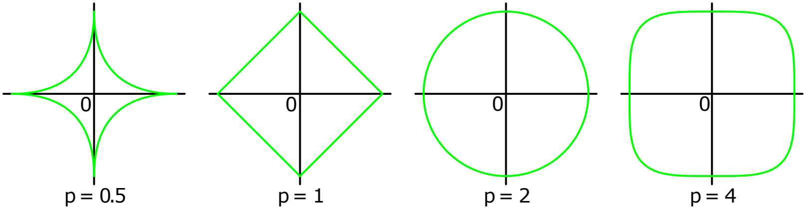
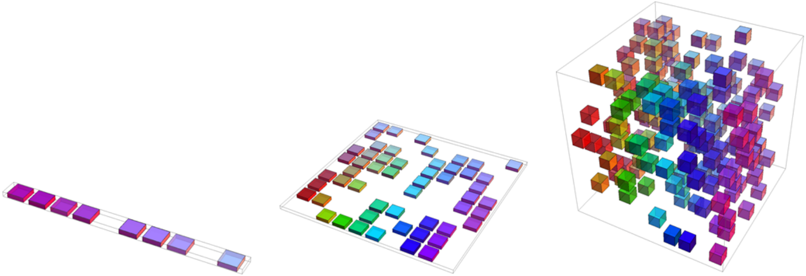


Figure 1: Contours of the distance from the origin O for various values of the parameter p

The Curse of dimensionality

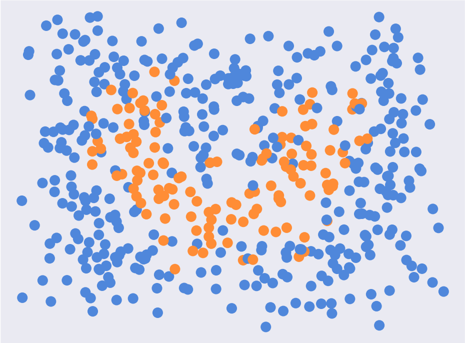
- The more dimensions we have, the more examples we need
- The number of examples that we have in a volume of space *decreases exponentially* with the number of dimensions
  - If the number of dimensions is very high the nearest neighbours can be very far away



Effects of Hyper-parameters

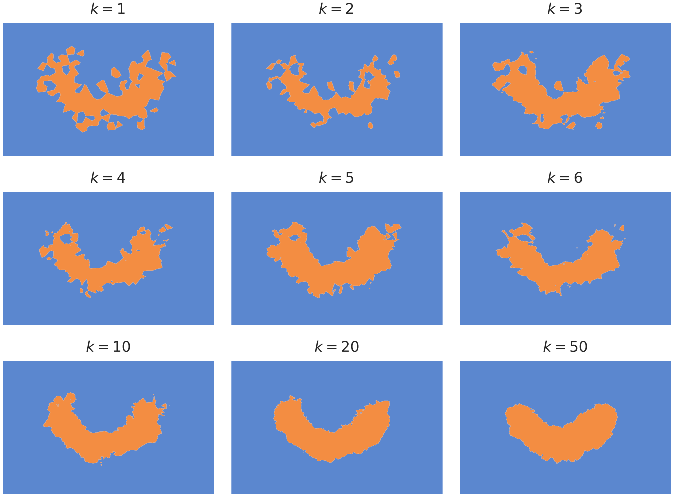
k Parameter

- Data set $\mathcal{D}$ with 500 samples belonging to two classes



k Parameter (cont.)

- Decision regions for various values of k



References

Goodfellow, I., Bengio, Y., and Courville, A. (2016).
Deep learning.
MIT press.

Lê, B. and Tô, V. (2014).
Cở sở trí tuệ nhân tạo.
Nhà xuất bản Khoa học và Kỹ thuật.

Nguyen, T. (2018).
Artificial intelligence slides.
Technical report, HCMC University of Sciences.

Russell, S. and Norvig, P. (2016).
Artificial intelligence: a modern approach.
Pearson Education Limited.