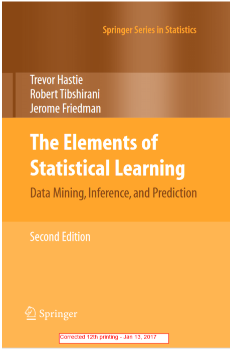
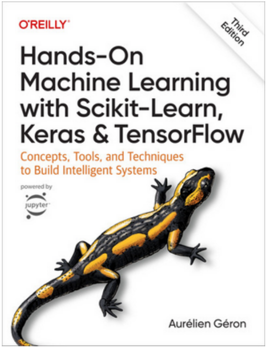
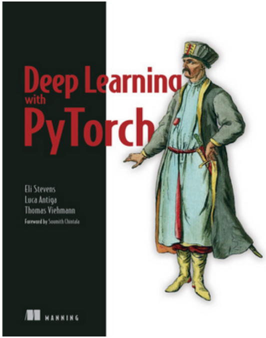
Machine Learning
Lecture 3

Literature







(HTF) T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning. Second Edition. 2017

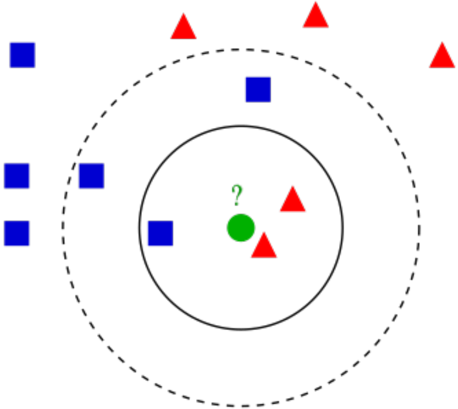
(AG) Au. Geron. Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow. Third Edition. 2022

(AVS) L. Antiga, Th. Viehmann, E. Stevens. Deep Learning with PyTorch. 2020

Content

- **Weighted kNN**
- **kNN for Regression**
- **Pandas**

Weighted k Nearest Neighbors (kNN)



https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Weighted k Nearest Neighbors (kNN)

$$a(x) = \underset{c \in \mathbb{Y}}{argmax} \sum_{s=1}^{k} w_s \left[y^{(i_s)} = c \right].$$

$w_s = \frac{k+1-s}{k}$
$w_s = q^s$
$w_s = K\left(\frac{d(x,\ x^{(i_s)})}{h}\right)$, where K is a non-increasing function defined on $[0,\ 1]$.

k Nearest Neighbors (kNN) for Regression: training

- We are given $\left(x^{(i)},\ y^{(i)}\right)$, $i = 1,\ 2, \ldots,\ N$.
- $\mathbb{Y} = \mathbb{R}$ (classification problem).
- training $=$ memorizing of the given data.

k Nearest Neighbors (kNN) for Regression: prediction

- We have a new feature x.
- Define the distance between new feature and $x^{(i)}$: $d\left(x,\ x^{(i)}\right)$.
- Rearrange the objects by the closeness to x :

$$d\left(x,\ x^{(i_1)}\right) \le d\left(x,\ x^{(i_2)}\right) \le \ldots \le d\left(x,\ x^{(i_N)}\right).$$

- Look at the first k labels and assign the class with the highest number of representatives:

$$a(x) = \frac{1}{k}\sum_{s=1}^{k} y^{(i_s)}.$$

k Nearest Neighbors (kNN) Weighted Regression

$$a(x) = \frac{\sum_{s=1}^{k} w_s y^{(i_s)}}{w_s}.$$

If $w_s = K\left(\frac{d(x,\ x^{(i_s)})}{h}\right)$, it is a Nadaraya — Watson estimator.

Pros

- If there are many examples, the algorithm gives a good prediction.
- Simple training.
- Few hyperparameters.
- There exist problems, where kNN works better, e.g., texts classification with many classes.

Cons

- Usually, other models work better.
- Extensive use of storage.
- Sorting for finding the neighbors.
- Limited tuning of the model.

Pandas

https://github.com/anton-selitskiy/RIT_ML