University of Rome "La Sapienza"

Master in Artificial Intelligence and Robotics

Machine Learning

A.Y. 2018/2019

Prof. Luca Iocchi

Sapienza University of Rome, Italy
Master in Artificial Intelligence and Robotics
Machine Learning (2018/19)

13. Instance based Learning

Luca Iocchi

Summary

- Non-parametric models
- K-NN for classification
- Localized weighted regression

References

C. Bishop. Pattern Recognition and Machine Learning. Sect. 2.5

Parametric and non-parametric models

Parametric model: Model has a fixed number of parameters

Examples:

- Linear regression
- Logistic regression
- Perceptron
- ...

Non-parametric model: Number of parameters grows with amount of data

K-nearest neighbors

Simple non-parametric model: **instance-based learning**

Classification with K-NN (target $f : X \mapsto C$, data set $D = \{(x_i, y_i)_{i=1}^n\}$:

1. Find K nearest neighbors of new instance $\mathbf{x}$
2. Assign to $\mathbf{x}$ the most common label among the majority of neighbors

Likelihood of class c for new instance $\mathbf{x}$:

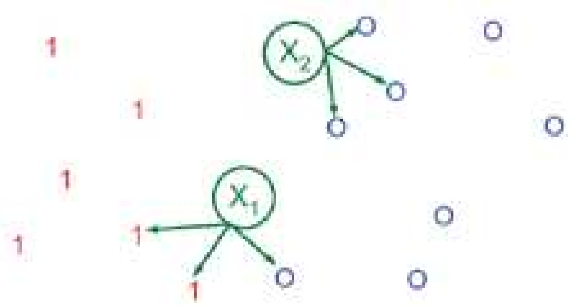
$$p(c|\mathbf{x}, D, K) = \frac{1}{K} \sum_{i \in N_K(\mathbf{x}, D)} \mathbb{I}(y_i = c),$$

with $N_K(\mathbf{x}, D)$ the K nearest points to $\mathbf{x}$ and $\mathbb{I}(e) = \begin{cases} 1 & \text{if } e \text{ is true} \\ 0 & \text{if } e \text{ is false} \end{cases}$.

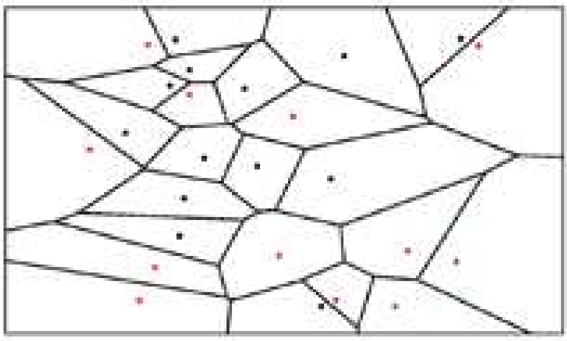
Requires storage of all the data set!

K-nearest neighbors

Example: K = 1



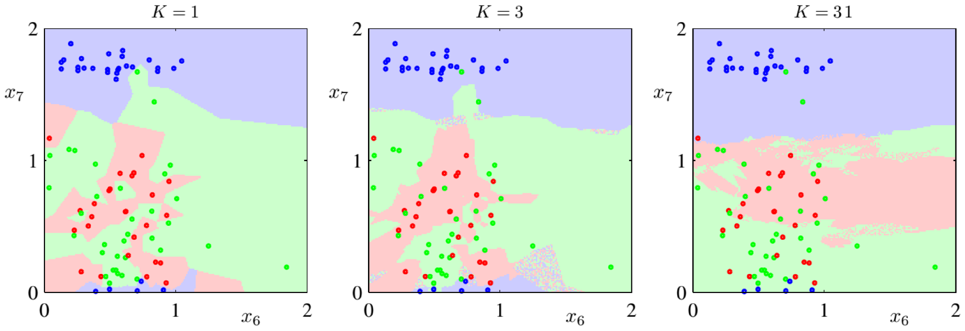
$K = 3$



Voronoi tesselation for $K = 1$

K-nearest neighbors

Increasing K brings to smoother regions (reducing overfitting)



Kernelized nearest neighbors

Distance function in computing $N_K(\mathbf{x}, D)$

$$\|\mathbf{x} - \mathbf{x}_i\|^2 = \mathbf{x}^T\mathbf{x} + \mathbf{x}_i^T\mathbf{x}_i - 2\mathbf{x}^T\mathbf{x}_i.$$

can be kernelized by using a kernel $k(\mathbf{x}, \mathbf{x}_i)$

Localized weighted regression

Regression problem $f : X \mapsto \Re$ with data set $D = \{(x_i, y_i)_{i=1}^N\}$

Given new instance $\mathbf{x}$

$$\hat{f}(\mathbf{x}) = \sum_{i \in N_K(\mathbf{x}, D)} y_i k(\mathbf{x}, \mathbf{x}_i)$$

i.e., fitting a constant function locally using kernel k

We can also fit other models (e.g., a linear model)