Sapienza University of Rome

Master in Artificial Intelligence and Robotics
Master in Engineering in Computer Science

Machine Learning

A.Y. 2020/2021

Prof. L. Iocchi, F. Patrizi

no create a model but use directly the dataset for prediction

10. Instance based learning

L. Iocchi, F. Patrizi

Summary

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

References

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

Parametric and non-parametric models

we have focussed on the use of probability distributions having specific functional forms governed by a small number of parameters whose values are to be determined from a data set. This is called the parametric approach to density modelling. An important limitation of this approach is that the chosen density might be a poor model of the distribution that generates the data, which can result in poor predictive performance. We consider some nonparametric approaches to density estimation that make few assumptions about the form of the distribution.

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

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

K-nearest neighbors

One of the difficulties with the kernel approach to density estimation is that the parameter h governing the kernel width is fixed for all kernels. In regions of high data density, a large value of h may lead to over-smoothing and a washing out of structure that might otherwise be extracted from the data. However, reducing h may lead to noisy estimates elsewhere in data space where the density is smaller. Thus the optimal choice for h may be dependent on location within the data space. This issue is addressed by nearest-neighbour methods for density estimation.

Classification problem: $f : X \mapsto C$ with data set $D = \{(\mathbf{x}_n, t_n)_{n=1}^{N}\}$

N.B: Note that the model produced by K nearest neighbours is not a true density model because the integral over all space diverges.

Classification with K-NN,

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 $\mathbf{x}$:

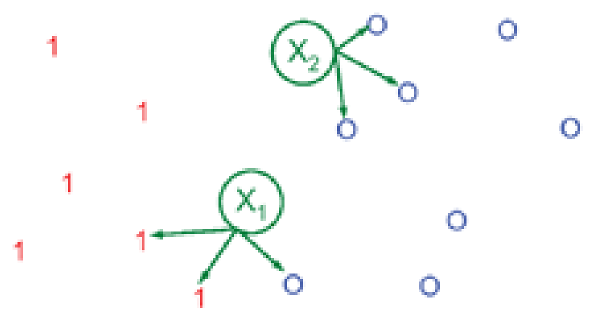
it collects the nearest K neighbors

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

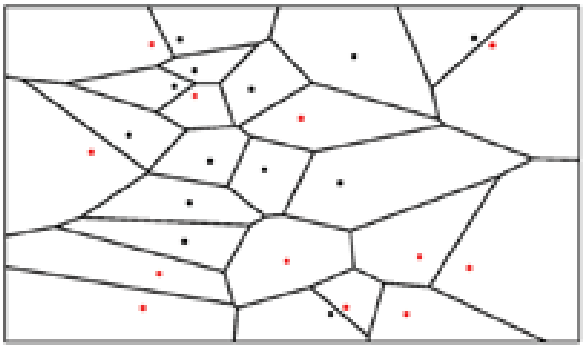
with $N_K(\mathbf{x}_n, D)$ the K nearest points to $\mathbf{x}_n$ and $\mathbb{I}(e) = \begin{cases} 1 & \text{if } e \text{ is true} \\ 0 & \text{if } e \text{ is false} \end{cases}$.

if we have a majority of +, the prediction will be +

K-nearest neighbors examples



$K = 3$



Voronoi tesselation for $K = 1$

Requires storage of all the data set!

Depends on a distance function!

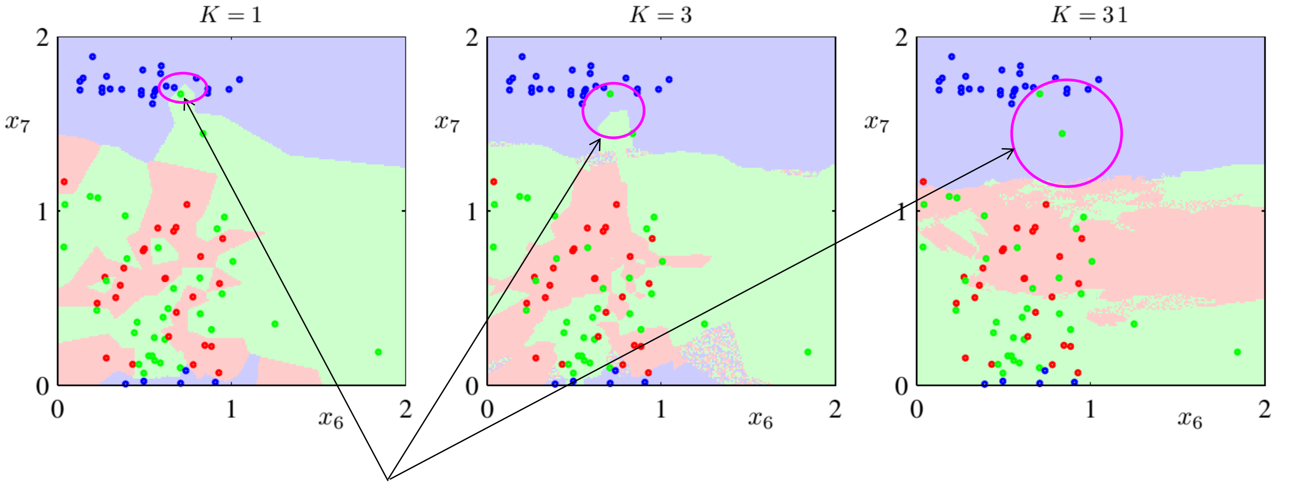
if we define this well, the methods works well

K-nearest neighbors

K-nearest-neighbour algorithm to the oil flow data for various values of K.

we see that K controls the degree of smoothing, so that small K produces many small regions of each class, whereas large K leads to fewer larger regions.

Increasing K brings to smoother regions (reducing overfitting)



when k changes we have that this point attracts always less the region of its color. K=30 is smoothed and more robust

An interesting property of the nearest-neighbour (K = 1) classifier is that, in the limit N → ∞, the error rate is never more than twice the minimum achievable error rate of an optimal classifier, i.e., one that uses the true class distributions

Kernelized nearest neighbors

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

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

can be kernelized by using a kernel $k(\mathbf{x}, \mathbf{x}_n)$ using the kernel we can generalize the concept of distance

both the K-nearest-neighbour method, and the kernel density estimator, require the entire training data set to be stored, leading to expensive computation if the data set is large.

Locally weighted regression

Now an example for applying this in regression (previously was for classification)

Regression problem $f : X \mapsto \Re$ with data set $D = \{(x_n, t_n)_{n=1}^N\}$

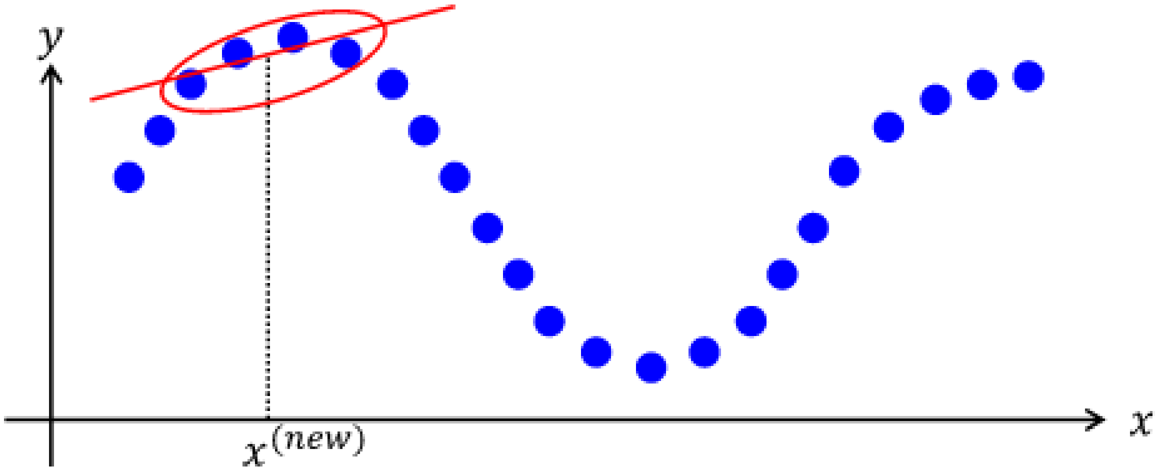
Fit a local regression model around the query sample $\mathbf{x}_q$

1. Compute $N_K(\mathbf{x}_q, D)$: K-nearest neighbors of $\mathbf{x}_q$
2. Fit a regression model $y(\mathbf{x}; \mathbf{w})$ on $N_K(\mathbf{x}_q, D)$
3. Return $y(\mathbf{x}_q; \mathbf{w})$

if K=2 we have interpolation

Locally weighted regression

Example with linear kernel



Summary

1. Non-parametric models based on storing data (lazy approaches)
2. No explicit model
3. Sensitive to parameters and distance function
4. Require storage of all data