10 Instance Based Learning

Other family of methods that work pretty well in some situations.

PARAMETRIC MODEL

VS

NON - PARAMETRIC MODEL

has a fixed number of parameters

(Linear regression
Logistic regression
Perceptron)

- Solution: finding a set of parameters

the number of parameters grows with the amount of data

- Easy to implement
- May have some drawbacks

Exam question : How large is the model? How many parameters?

No need to know the dataset, the size of the model for the parametric case is independent from the dataset

In instance based learning we have not the training phase.

This is perfectly reasonable. Once I have a new instance I try to do a direct prediction without building a model.

Actually the paramete of the instance based learning methods is the dataset itself, then the size of the model is the size of the dataset itself.

Classification of KNN (NN = nearest neighbor) is a region of the space that contains the set of samples around the instance I want to classify.

1. Find k nearest neighbors of new instance $\vec{x}$

2. Assign to $\vec{x}$ the most common label among the majority of neighbors

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

$$p(c|\vec{x}, D, k) = \frac{1}{k} \sum_{i \in N_k(\vec{x}, D)} \mathbb{I}(y_i = c)$$

$N_k(\vec{x}, D)$ is the k nearest points to $\vec{x}$

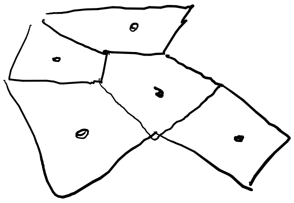
$$\mathbb{I}(e) = \begin{cases} 1 & \text{if } e \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

This method requires the storage of all the dataset, this is a main problem.

So KNN is ok if the dataset is not too large.

KNN is based on the definition of a distance, but is not easy, in most cases $\vec{x_1}$ and $\vec{x_2}$ are not just numbers and they may be representations of physical elements.
Futhermore if you scale one dimension, the solution changes.

$k = 1$ Voronoi Tessellation $\Rightarrow$



Note: the effect of k is critical to reduce overfitting

- Trade-Off on k : smoothing regions and reduce overfitting with big k, but increase computation)

We can kernelize also the instance-based-method instead of using a regular distance function based on the encoding of numbers you have in the dataset, you can use a kernel function (can solve the problems with the distance and dimensions of $\vec{x}$)

Regression problem $f: X \to \mathbb{R}$ with $D = \{(x_i, y_i)\}_{i=1}^{N}$

Fit a local regression model using kernel K

$$\boxed{\hat{f}(\vec{x}) = \sum_{i \in N_K(x,D)} y_i \, K(\vec{x}, \vec{x}_i)}$$

↓

fitting a cost function locally using kernel K