10. Instance based learning

10.1 K-nearest neighbors

$F : X \rightarrow C$ with $D = \{(x_n, t_n)^N_{n=1}\}$,

Classification with K-NN,

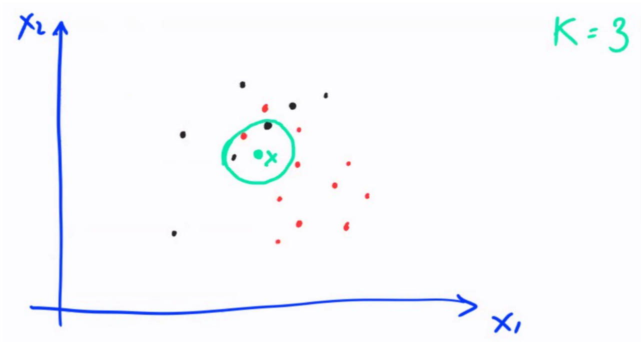
1. Find K nearest neighbors of new instance x

2. Assign to x the most common label among the majority of neighbors

Likelihood of c to new x:

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

with $N_k(x_n,D)$ nearest point and $I(e) =\{1$ if e is true, 0 if e is false.



Requires storage of all the data set, and depends on a distance function.

Increasing K brings to smoother regions (reducing overfitting).

With K=1 a point is the closest to itself (perfect, but expensive, overfit).

With K>1 reduce overfit, but doesn't ensure better performance.

Distance function: $\|x-x_n\|^2=x^Tx+x^T{}_nx_n-2x^Tx_n$.

can be kernelized by using a kernel $k(x,x_n)$

Regression $X \rightarrow R$

1. Compute $N_K(x_q,D)$: K-nearest neighbors of x_q

2. Fit a regression model $y(x;w)$ on $N_K(x_q,D)$

3. Return $y(x_q;w)$

Advantages of KNN: input space doesn't converge to an optimal solution, so use KNN transforming input space to feature space (expensive).