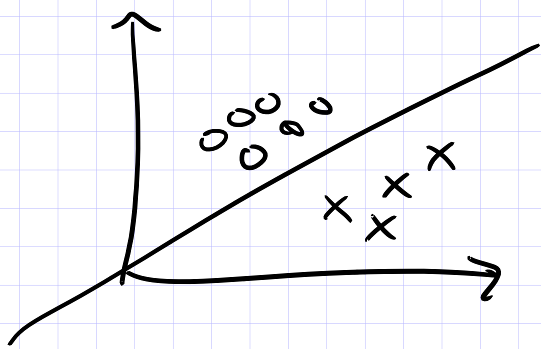
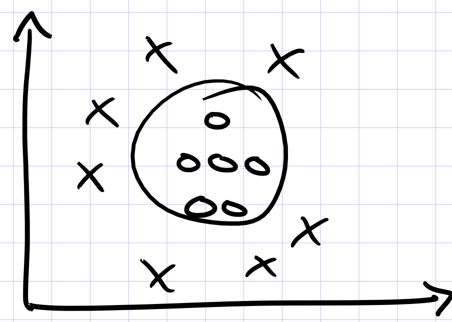
KNN Decision Rule





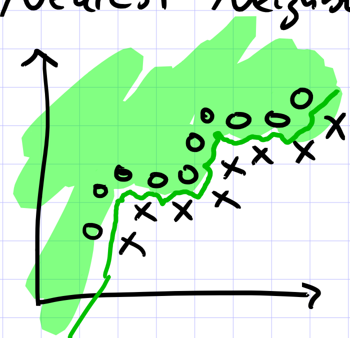
Depends:

1. our dataset D
2. our distance d
3. our choice of k

Choosing k

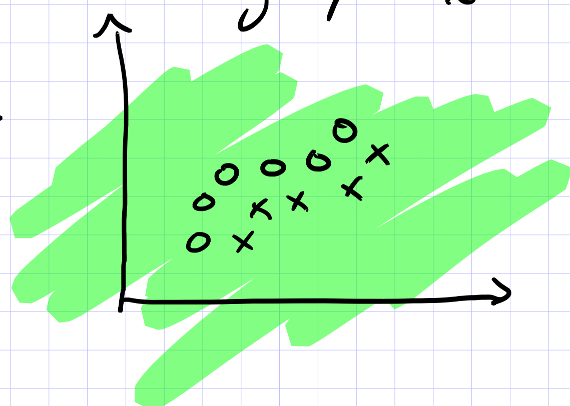
Special Case: $k=1$

"Nearest Neighbor"

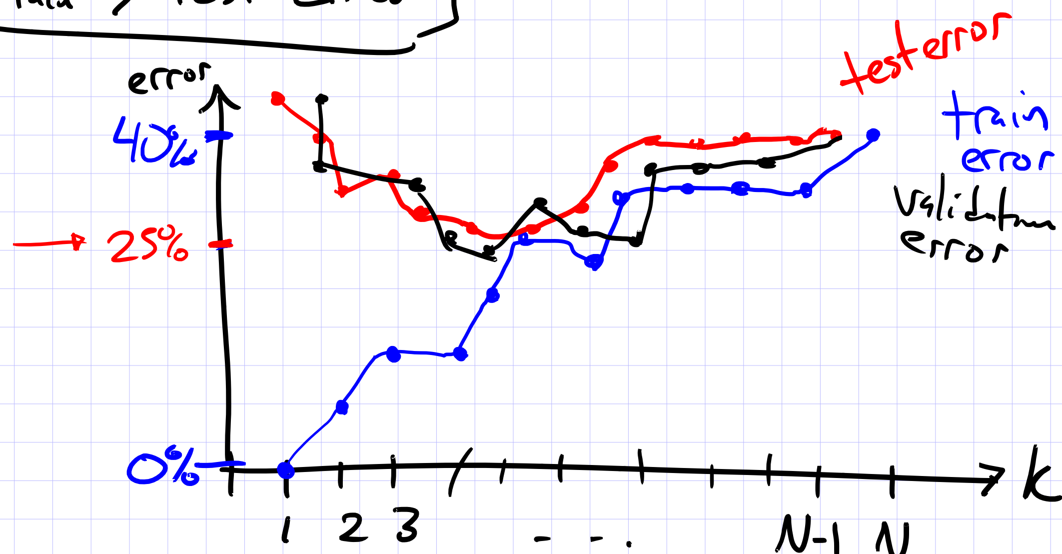


Special Case: $k=N$

"Majority Vote"



Train & Test Error



D is 40% $y^{(i)}=0$

60% $y^{(i)}=1$

$$D^{train} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix} \begin{bmatrix} x^{(1)} \\ \vdots \\ x^{(N)} \end{bmatrix} \quad 90\%$$

$$\hat{y}^{(test)} = \begin{bmatrix} h(x^{(t1)}) \\ \vdots \\ h(x^{(tT)}) \end{bmatrix} \qquad D^{test} = \begin{bmatrix} y^{(t1)} \\ \vdots \\ y^{(tT)} \end{bmatrix} \begin{bmatrix} x^{()} \\ \vdots \\ x^{()} \end{bmatrix} \quad 10\%$$

compare for test error

80% train

10% validation

10% test

100% $N+T$

★ Choose k based on validation error

Lecture 3

Function Approx. View

① There exists some unknown distribution p^* that generates our unlabeled data instances, $x^{(i)}$

$$x^{(i)} \sim p^*(x) \quad \forall i$$

$\sim$ denotes "is sampled from"

② Human expert annoted each training instance in D using a fixed unknown function h^*

$$y^{(i)} = h^*(x^{(i)}) \quad \forall i$$

where D comes from

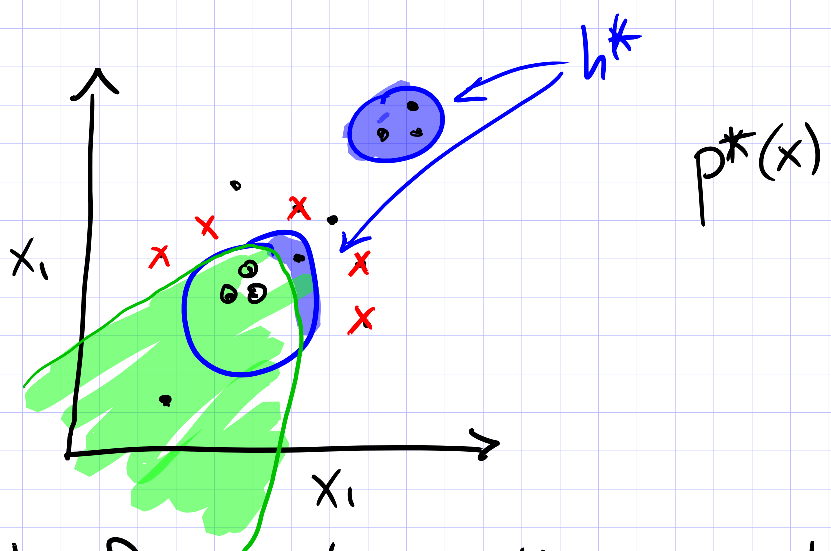
③ Learning algo. takes data D and output a (good) hypothesis $h \in H$

$$y^{(ti)} = h(\vec{x}^{(ti)})$$

④ Our goal: choose h with low error on unknown data from p^*

$$\text{error}(h, p^*) = P_{x \sim p^*}\left(h(x) \neq h^*(x)\right)$$

we can't compute this but validation error gives a good approximation



Question: What if we know that our data was not created as described above?