**KNN Decision Rule**

Depends on:
1. Our dataset $D$
2. Our distance $d$
3. Our choice of $k$

**Choosing $k$**

**Special Case: $k=1$**

"Nearest Neighbor"

**Special Case: $k=\infty$**

"Majority Vote"

**Train vs Test Error**

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

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

$D_{\text{train}} = [y^{(1)}]$

$x^{(1)}$

$\cdots$

$x^{(N)}$

$D_{\text{test}} = [y^{(N+1)}]$

$x^{(N+1)}$

$\cdots$

$x^{(N+T)}$

Compact for test error

Choose $k$ based on validation error
Lecture 3

Function Approx. View

1. There exists some unknown distribution \( p^* \) that generates
   or unlabeled data instances, \( x^{(i)} \)
   \[
   x^{(i)} \sim p^*(x) \quad \forall i
   \]
   "denotes is sampled from"

2. Human expert annotated each training instance in \( D \)
   using a fixed unknown function \( h^* \)
   \[
   y^{(i)} = h^*(x^{(i)}) \quad \forall i
   \]

3. Learning algo. takes data \( D \) and outputs a (good) hypothesis \( h \in H \)
   \[
   y^{(i)} = h(x^{(i)})
   \]

4. Our goal: choose \( h \) with low error on data from \( p^* \)
   \[
   \text{error}(h, p^*) = \mathbb{P}_{x \sim p^*}(h(x) \neq h^*(x))
   \]
   "we can't compute this"
   "but validation error gives a good approximation"

---

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