Machine Learning from Data

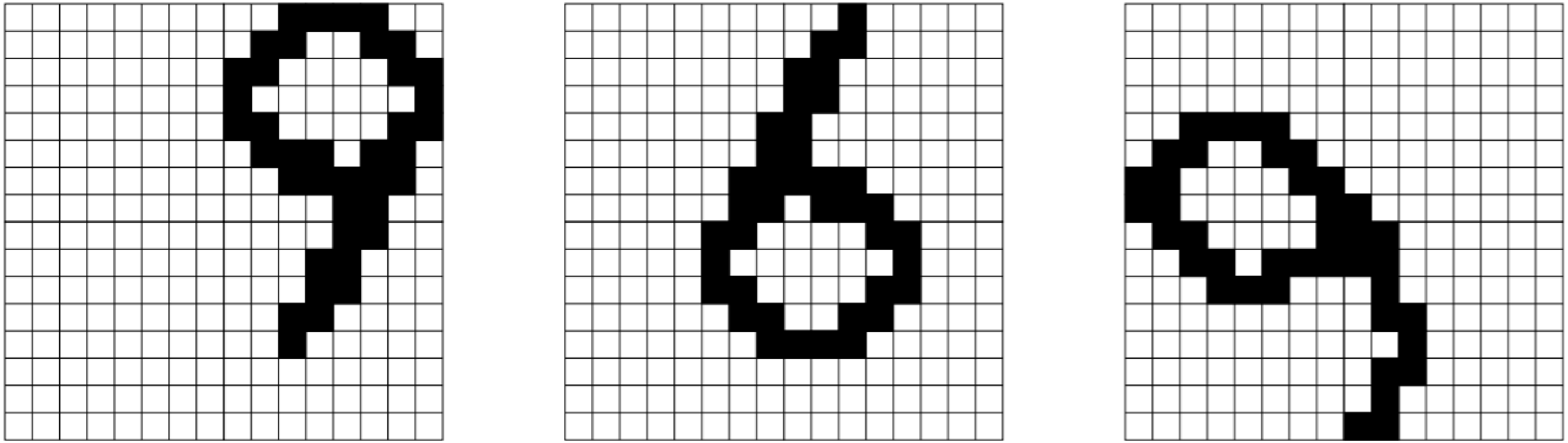
Lecture 16: Spring 2021

Today’s Lecture

- Similarity
- Nearest Neighbor

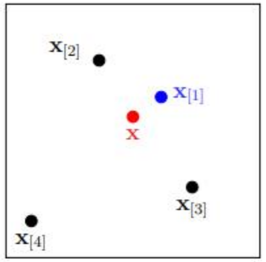
Ask a 5-year-old what is this?

Measuring Similarity



Nearest Neighbor

Test '$\mathbf{x}$' is classified using its nearest neighbor.

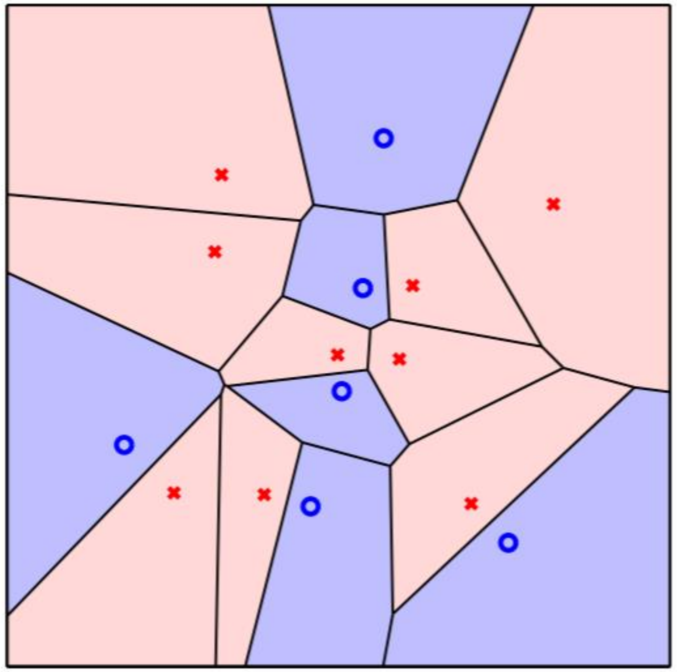


$$d(\mathbf{x}, \mathbf{x}_{[1]}) \leq d(\mathbf{x}, \mathbf{x}_{[2]}) \leq \cdots \leq d(\mathbf{x}, \mathbf{x}_{[N]})$$

$$g(\mathbf{x}) = y_{[1]}(\mathbf{x})$$

No training needed!

$$E_{\text{in}} = 0$$



Nearest neighbor Voronoi tesselation

Proving $E_{\text{out}} \leq 2E^*_{\text{out}}$

$$\pi(\mathbf{x}) = \mathbb{P}[y = +1|\mathbf{x}].$$ ← the target in logistic regression

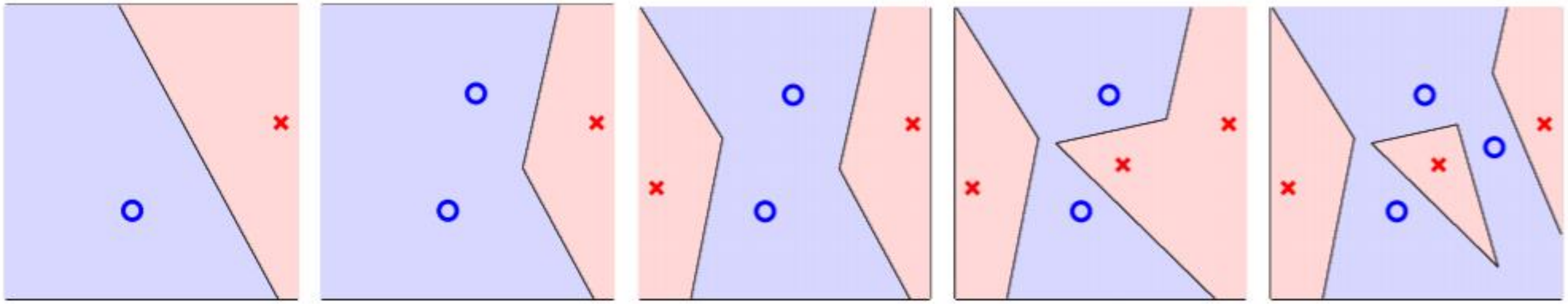
Assume $\pi(\mathbf{x})$ is continuous and $\mathbf{x}_{[1]} \xrightarrow{N\to\infty} \mathbf{x}$. Then $\pi(\mathbf{x}_{[1]}) \xrightarrow{N\to\infty} \pi(\mathbf{x})$.

$$\begin{aligned}
\mathbb{P}[g_N(\mathbf{x}) \neq y] &= \mathbb{P}[y = +1, y_{[1]} = -1] + \mathbb{P}[y = -1, y_{[1]} = +1], \\
&= \pi(\mathbf{x}) \cdot (1 - \pi(\mathbf{x}_{[1]})) + (1 - \pi(\mathbf{x})) \cdot \pi(\mathbf{x}_{[1]}), \\
&\to \pi(\mathbf{x}) \cdot (1 - \pi(\mathbf{x})) + (1 - \pi(\mathbf{x})) \cdot \pi(\mathbf{x}), \\
&= 2\pi(\mathbf{x}) \cdot (1 - \pi(\mathbf{x})), \\
&\leq 2\min\{\pi(\mathbf{x}), 1 - \pi(\mathbf{x})\}.
\end{aligned}$$

The best you can do is

$$E^*_{\text{out}}(\mathbf{x}) = \min\{\pi(\mathbf{x}), 1 - \pi(\mathbf{x})\}.$$

Nearest Neighbor ‘Self-Regularizes’



$N = 2$ $N = 3$ $N = 4$ $N = 5$ $N = 6$

A simple boundary is used with few data points.

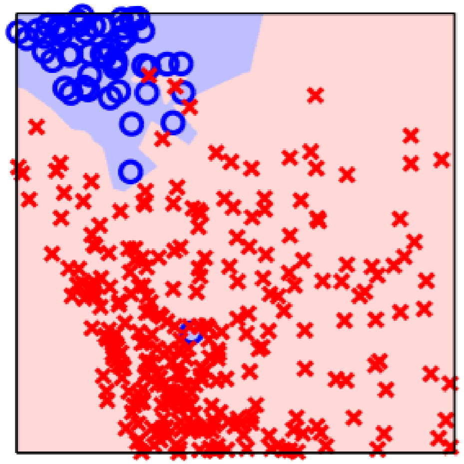
A more complicated boundary is possible *only* when you have more data points.

regularization guides you to simpler hypotheses when data quality/quantity is lower.

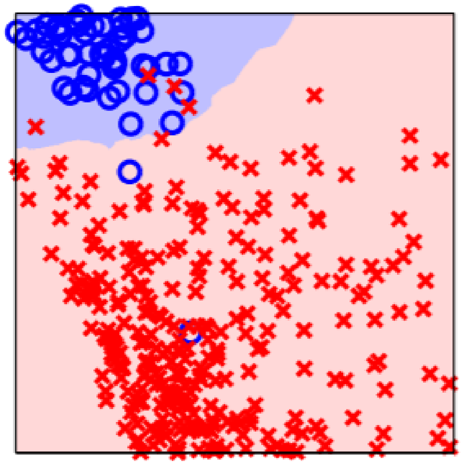
k-Nearest Neighbor

$$g(\mathbf{x}) = \mathrm{sign}\left(\sum_{i=1}^{k} y_{[i]}(\mathbf{x})\right).$$

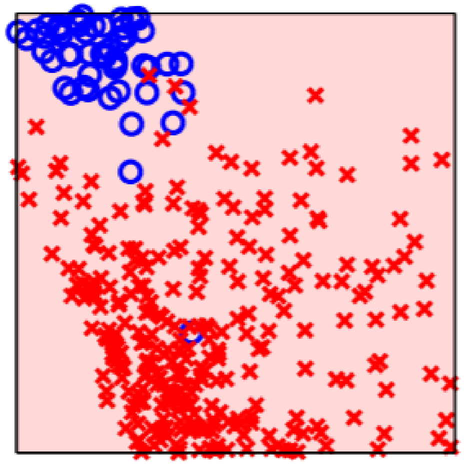
(k is odd and $y_n = \pm 1$).



1-NN rule



21-NN rule



127-NN rule

The Role of k

k determines the tradeoff between fitting the data and overfitting the data.

Theorem. For $N \to \infty$, if $k(N) \to \infty$ and $k(N)/N \to 0$ then,

$$E_{\text{in}}(g) \to E_{\text{out}}(g) \qquad \text{and} \qquad E_{\text{out}}(g) \to E^*_{\text{out}}.$$

For example $k = \left\lceil \sqrt{N} \right\rceil$.

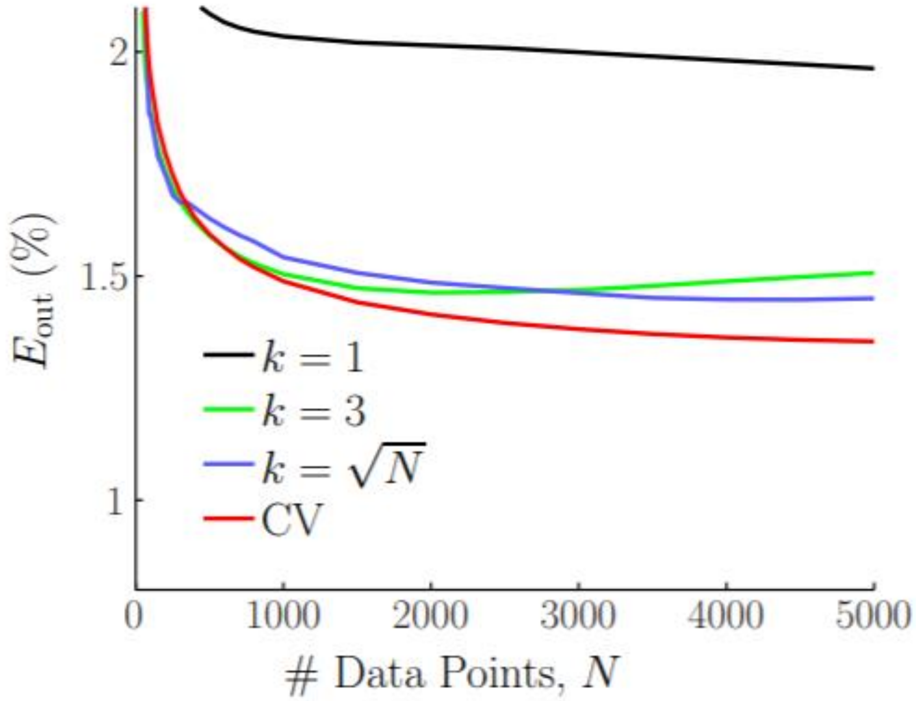
3 Ways To Choose k

1. $k = 3$.

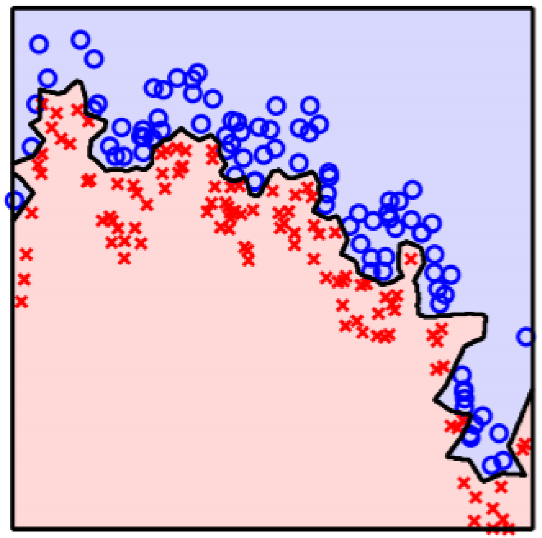
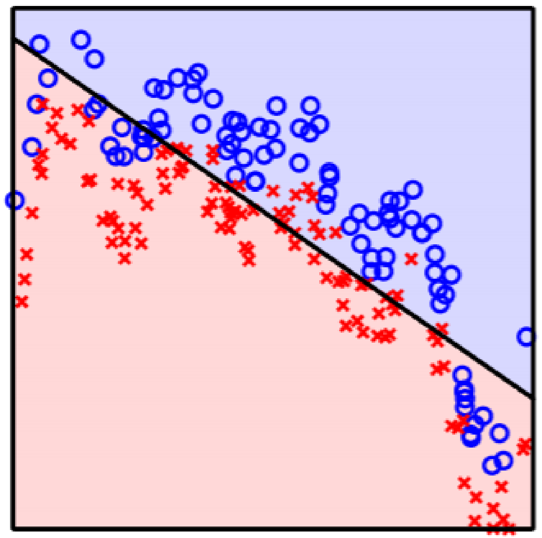
2. $k = \left\lceil \sqrt{N} \right\rceil$.

3. Validation or cross validation:

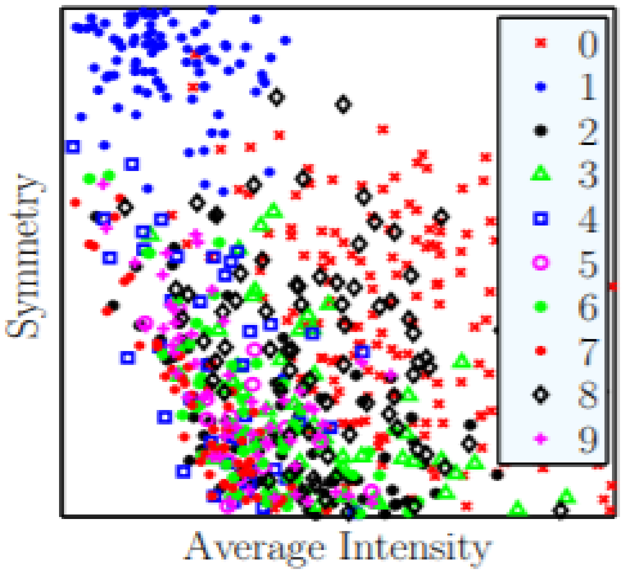
   k-NN rule hypotheses g_k^- constructed on training set, tested on validation set, and best k is picked.

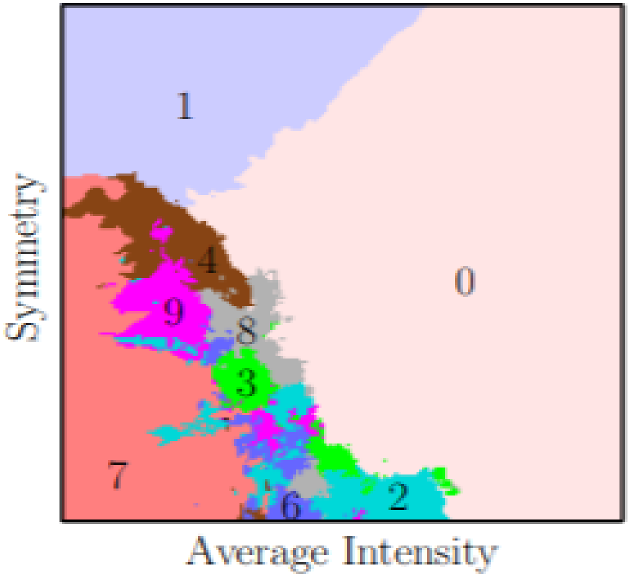


Nearest Neighbor is Nonparametric

NN-rule	Linear Model
	
no parameters	$(d+1)$ parameters
expressive/flexible	rigid, always linear
$g(\mathbf{x})$ needs data	$g(\mathbf{x})$ needs only weights
generic, can model anything	specialized

Nearest Neighbor Easily Extends to Multiclass





True	Predicted										
	0	1	2	3	4	5	6	7	8	9	
0	**13.5**	0.5	0.5	1	0	0.5	0	0	0.5	0	16.5
1	0.5	**13.5**	0	0	0	0	0	0	0	0	14
2	0.5	0	**3.5**	1	1	1.5	1	1	0	0.5	10
3	2.5	0	1.5	**2**	0.5	0.5	0.5	0.5	0.5	1	9.5
4	0.5	0	1	0.5	**1.5**	0.5	1	2	0	1.5	8.5
5	0.5	0	2.5	1	0.5	**1.5**	1	1	0	0.5	7.5
6	0.5	0	2	1	1	1	**1**	1	0	1	8.5
7	0	0	1.5	0.5	1.5	0.5	1	**3**	0	1	9
8	3.5	0	0.5	1	0.5	0.5	0.5	0	**0.5**	1	8
9	0.5	0	1	1	1	0.5	1	1	0.5	**2**	8.5
	22.5	14	14	9	7.5	7	7	9.5	2	8.5	100

41% accuracy!

Highlights of k-Nearest Neighbor

1. Simple.
2. No training.
3. Near optimal E_{out}.
4. Easy to justify classification to customer.

A **good!** method

5. Can easily do multi-class.
6. Can easily adapt to regression or logistic regression

$$g(\mathbf{x}) = \frac{1}{k}\sum_{i=1}^{k} y_{[i]}(\mathbf{x}) \qquad\qquad g(\mathbf{x}) = \frac{1}{k}\sum_{i=1}^{k} [\![y_{[i]}(\mathbf{x}) = +1]\!]$$

7. **Computationally demanding**. $\longleftarrow$ we will address this next

Thanks!