Learning From Data
Lecture 16
Similarity and Nearest Neighbor

The simplest method of learning that we know. Classify according to similar objects you have seen.

Measuring similarity:

\[ d(x, x') = ||x - x'|| \]

Nearest neighbor: My 5-Year-Old Called It “A ManoHorse”
Nearest Neighbor

Test 'x' is classified using its nearest neighbor.

\[ d(x, x_{(j)}) \leq d(x, x_{(k)}) \leq \ldots \leq d(x, x_{(N)}) \]

\[ g(x) = y_{(j)}(x) \]

No training needed!

\[ E_{in} = 0 \]

Nearest neighbor Voronoi tessellation

No training needed!

\[ E_{in} = 0 \]

Nearest neighbor Voronoi tessellation

What about \( E_{out} \)?

Theorem: \( E_{out} \leq 2E_{out}^* \)

(with high probability as \( N \to \infty \))

VC analysis: \( E_{in} \) is an estimate for \( E_{out} \).

Nearest neighbor analysis: \( E_{in} = 0, E_{out} \) is small.

So we will never know what \( E_{out} \) is, but it cannot be much worse than the best anyone can do.

Half the classification power of the data is in the nearest neighbor
Proving $E_{\text{out}} \leq 2E_{\text{out}}^*$

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

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

$P[g_N(x) = y] = P[y = 1, y_{[1]} = -1] + P[y = -1, y_{[1]} = +1]$, \[
= \pi(x) \cdot (1 - \pi(x_{[1]})) + (1 - \pi(x)) \cdot \pi(x_{[1]}),
\]

$\rightarrow \pi(x) \cdot (1 - \pi(x)) + (1 - \pi(x)) \cdot \pi(x)$, \[
= 2\pi(x) \cdot (1 - \pi(x)),
\]

$\leq 2 \min\{\pi(x), 1 - \pi(x)\}$.

The best you can do is $E_{\text{out}}^* = \min\{\pi(x), 1 - \pi(x)\}$.

$k$-Nearest Neighbor

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

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

$k$ 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, $E_{\text{in}}(g) \to E_{\text{out}}(g)$ and $E_{\text{out}}(g) \to E_{\text{out}}^*$.

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

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.
3 Ways To Choose $k$

1. $k = 3$.

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

3. Validation or cross validation: $k$-NN rule hypotheses $g_k$ constructed on training set, tested on validation set, and best $k$ is picked.

### Similarity and Nearest Neighbor

Nearest Neighbor is Nonparametric

**NN-rule**

- No parameters
- Expressive/ Flexible

**Linear Model**

- $(d+1)$ parameters
- Rigid, always linear

### Nearest Neighbor Easily Extends to Multiclass

**Average Intensity**

$g(x) = \frac{1}{k} \sum_{i=1}^{k} y_i(x)$

**Symmetry**

$g(x) = \frac{1}{k} \sum_{i=1}^{k} [y_i(x) = +1]$  

41% accuracy!

### Highlights of $k$-Nearest Neighbor

1. Simple.

2. No training.

3. Near optimal $E_{\text{out}}$.

4. Easy to justify classification to customer.

5. Can easily do multi-class.

6. Can easily adapt to regression or logistic regression

7. Computationally demanding → we will address this next