Today

1. VC Dimension & PAC Bounds
2. No Free Lunch
3. Bias & Consistency
4. Bayesian Model Selection & MDL

**VC Dimension**

**Example:** 2D Linear Classifier

\[ x(x) = \text{sign}(\alpha^T x + a_0) \]

How many different class boundaries can it represent?

**Definition of "Different"**

\( n = 1 : \mathcal{D} = \{ x_1, x_2 \} \Rightarrow 2 \) different boundaries

\[ \alpha^T x_1 + a_0 < 0 \]

\[ \alpha^T x_1 + a_0 > 0 \]

\( n = 2 : \mathcal{D} = \{ x_1, x_2, x_3 \} \Rightarrow 4 \) different classifiers

\( n = 3 : 8 \) different classifier

\( n = 4 : ? \)

Possible decision regions \( R_{+1} : \)

\[ R_1 = \emptyset, \{ x_3 \}, \{ x_2, x_3 \}, \{ x_1, x_2, x_3 \} \]

\[ \{ x_1, x_2, x_3, x_4 \} \]

but not \( \{ x_3, x_4 \} !! \)

\( N_p(n) = \# \) possible partitions of n-sample training set
**Linear 2-D classifier:**

1. Rotating \( z \) gives \( n \) data orderings
2. Threshold each ordering \( \Rightarrow n \) partitions/ordering
   \( \Rightarrow N_p(n) \leq n^2 \)

**Linear d-Dimensional classifier:** \( N_p(n) \leq n^d \)

**Vapnik-Chervonenkis Dimension of a classifier:**

\[
d_{vc} = \lim_{n \to \infty} \frac{\log N_p(n)}{\log n}\]

**Probably Approximately Correct Bounds (PAC Bounds)**

Define \( L_{\alpha}(x_i, y_i) = \text{loss function}, \ \text{e.g.,} \ L = [\alpha(x_i) \neq y_i] \)

\[
R(\alpha) = E[L_{\alpha}] = \int \int L_{\alpha}(x, y) p(x, y) \, dx \, dy = \text{Risk}
\]

\[
\text{Rem}_n(\alpha, D) = \frac{1}{n} \sum_{i=1}^{n} L_{\alpha}(x_i, y_i) = \text{"Empirical Risk"}
\]

Assume \((x_i, y_i)\) i.i.d. selections from \( p(x, y) \)

Then

\[
P \left( |R(\alpha) - \text{Rem}_n(\alpha)| < G \left( \frac{d_{vc}}{n}, \delta \right) \right) > 1 - \delta
\]

\[
G \left( \frac{d_{vc}}{n}, \delta \right) = \text{"Generalization Error"} \quad \text{"confidence"}
\]

\[
G \left( \frac{d_{vc}}{n}, \delta \right) \approx \frac{d_{vc}}{n} \quad \text{(only weak dependence on } \delta \text{)}
\]

\[
\Rightarrow R(\alpha) < \text{Rem}_n(\alpha) + G \left( \frac{d_{vc}}{n} \right) \quad \text{w/ prob } 1 - \delta
\]
**Instructive Examples**

\[ x = \text{scalar}, \quad y \in \{-1, 1\} \]

**Classifier 1:**  \( \alpha(x) = \text{sign}(x + b) \)

\[ d_{vc} = 1 \]

**Classifier 2:**  \( \alpha(x) = \text{sign}(\sin(10\pi x) + b) \)

\[ d_{vc} = 1 \]

Which is better?

A: Whichever produces lower \( R_{\text{emp}}(\alpha) \)

\( \approx \) whichever better matches reality!

**No Free Lunch** Define \( \text{Ave}(f) = \text{average over all } p(x,y) \)

**Version 1** (Textbook): \( \mathcal{D} \) not i.i.d.

For any particular \( p(x,y) \),

I as many "good" \( \mathcal{D} \) as "bad" \( \mathcal{D} \),

thus \( \# \mathcal{D} \) selected from \( p(x,y) \) without i.i.d. assumption,

\[ \text{Ave}(R(\alpha_1)) - \text{Ave}(R(\alpha_2)) = 0 \]

regardless of \( R_{\text{emp}}(\alpha_1), R_{\text{emp}}(\alpha_2) \)

for any classifiers \( \alpha_1, \alpha_2 \)

**Version 2** (Not in textbook): \( \mathcal{D} \) i.i.d.

Given \( R_{\text{emp}}(\alpha_1) = R_{\text{emp}}(\alpha_2) \)

\[ P(R(\alpha_1) < R(\alpha_2)) = \frac{1}{2} \quad \text{independent of } d_{vc}(\alpha_1), d_{vc}(\alpha_2) \]
Reason: PAC bounds tell us that, w/ prob 1 - \delta

\[ R_{emp} - \frac{dvc(\alpha_1)}{n} < R(\alpha_1) < R_{emp} + \frac{dvc(\alpha_1)}{n} \]

\[ R_{emp} - \frac{dvc(\alpha_2)}{n} < R(\alpha_2) < R_{emp} + \frac{dvc(\alpha_2)}{n} \]

\[ dvc(\alpha_1) < dvc(\alpha_2) \Rightarrow R(\alpha_1) \text{ has low variance} \]

but mean \( \bar{y} \) same

**Bias \& Consistency**

Let \( g(\mathbf{x}_0 | \mathbf{D}) = \text{function of } \mathbf{x}_0 \), trained on \( \mathbf{D} = \{ \mathbf{x}_1, \ldots, \mathbf{x}_n \} \)

Examples:

1. \( g(\mathbf{x}_0 | \mathbf{D}) = \frac{1}{(2\pi)^{d/2} |\Sigma(\mathbf{D})|^{1/2}} e^{-\frac{1}{2} (\mathbf{x}_0 - \mu(\mathbf{D}))^T \Sigma(\mathbf{D})^{-1} (\mathbf{x}_0 - \mu(\mathbf{D}))} \)

   Gaussian trained on \( \mathbf{D} \)

2. \( g(\mathbf{x}_0 | \mathbf{D}) = \frac{1}{n} \sum_{i=1}^{n} C_i (\mathbf{x}_0 - \mathbf{x}_i) \)

   Parzen window estimate

3. \( g(\mathbf{x}_0 | \mathbf{D}) = \text{neural net trained on } \mathbf{D} \)

   starting from known \( \widehat{\mathbf{w}}_0 \)

\[ \text{MSE}(\mathbf{x}_0) = E \left[ (g(\mathbf{x}_0 | \mathbf{D}) - F(\mathbf{x}_0))^2 \right] \]

\( \uparrow \text{Target function} \)

\[ = \int \int \ldots \int (g(\mathbf{x}_0 | \mathbf{D}) - F(\mathbf{x}))^2 p(\mathbf{x}_1) d\mathbf{x}_1 \ldots p(\mathbf{x}_n) d\mathbf{x}_n \]
Then
\[ \text{MSE}(\hat{x}_0) = \text{Bias}^2 = \text{Var}(g) \]

\[ \text{Bias} \equiv E_D \left[ g(x_0 | D) \right] - F(\hat{x}_0) \]

\[ \text{Var}(g) \equiv E_D \left[ (g(x_0 | D) - E[g(x_0 | D)])^2 \right] \]

Low bias \iff \text{g}(x_0 | D) \text{ flexible (high dvc)}
Low variance \iff \text{g}(x_0 | D) \text{ not too dependent on D (low dvc)}

**Example: Classification**

Suppose \( g(x_0 | D) = \alpha(x_0) \in \{ w_1 \text{ class } 1 \}
\]

Define \( F(\hat{x}_0) = P(y_0 = w_1 | \hat{x}_0) \)

**Zero-Bias Classifier**

Suppose \( \alpha(x_0) \) good enough so \( E_D \left[ g(x_0 | D) \right] = F(\hat{x}_0) \)

Then \( \text{Var}(g) = P(w_1 | \hat{x}_0) \left( 1 - P(w_1 | \hat{x}_0) \right)^2 \)
\[ + \left( 1 - P(w_1 | \hat{x}_0) \right) \left( 0 - P(w_1 | \hat{x}_0) \right)^2 \]
\[ = P(w_1 | \hat{x}_0) \left( 1 - P(w_1 | \hat{x}_0) \right) \]

**Zero-Variance Case**

Suppose \( g(x_0 | D) = 1 \) everywhere!

Then \( \text{Bias}^2 = (1 - F(\hat{x}_0))^2 = (1 - P(w_1 | \hat{x}_0))^2 \)
\[ = \text{MSE}(\hat{x}_0) \]
4. BAYESIAN CLASSIFIER SELECTION &
MINIMUM DESCRIPTION LENGTH

Goal: Choose \( \alpha(x_0 | D) \) to minimize
\[
\log P(\alpha, D) = \log P(\alpha) + \log P(D | \alpha)
\]
\[
\log P(D | \alpha) = \text{log likelihood of data}
\]
\[
\log P(\alpha) = \text{prior over "reasonable" hypotheses}
\]
Suppose \( \alpha \in \mathcal{A} \), the set of hypotheses

Consider a Huffman code \( C(\alpha) = [c_1, c_2, ...] \)
\( c_i \) = \( i^{th} \) bit
\[
c_1 = 0 \quad \text{if} \quad \alpha \in A_0, \quad P(\alpha \in A_0) = \frac{1}{2}
\]
\[
c_1 = 1 \quad \text{if} \quad \alpha \in A_1, \quad P(\alpha \in A_1) = \frac{1}{2}
\]
\[
[c_1, c_2] = [0, 1] \quad \text{if} \quad \alpha \in A_{01}, \quad P(\alpha \in A_{01}) = \frac{1}{4}
\]

Then \( \text{length}(C(\alpha)) = \log_2 P(\alpha) \) bits

Similarly let \( C(D | \alpha) = [d_1, d_2, ...] \)
\( [d_1, d_2] = [0, 1] \quad \text{if} \quad D \in R_{01}, \quad P(D \in R_{01} | \alpha) = \frac{1}{4} \)

Then \( \text{length}(C(D | \alpha)) = \log_2 P(D | \alpha) \)

\( \Rightarrow \) Choose \( \alpha \) to minimize description length
\[
\text{length}[C(\alpha)] + \text{length}[C(D | \alpha)]
\]
Example, d-dimensional hyperplane: \( \text{length}[C(\alpha)] \propto d + 1 \).