Sauer Shelah Lemma

$$S_n(\mathcal{F}(Z^n)) \le \binom{n}{\le d} \le (n+1)^d$$

max # of labelings of n points

$d = V(\mathcal{F})$

Finite class lemma

$$R_n(\mathcal{F}(Z^n)) \le \frac{2L\sqrt{\log N}}{n}$$

$L = \sqrt{n}$ $\quad N = (n+1)^d$

$$= 2\sqrt{\frac{d\log(n+1)}{n}}$$

THEOREM 7.1. *Let $\mathcal{F}$ be a VC class of binary-valued functions $f : \mathsf{Z} \to \{0,1\}$ on some space Z. Let Z^n be an i.i.d. sample of size n drawn according to an arbitrary probability distribution $P \in \mathcal{P}(\mathsf{Z})$. Then, with probability one,*

$$R_n(\mathcal{F}(Z^n)) \le 2\sqrt{\frac{V(\mathcal{F})\log(n+1)}{n}}.$$

A more refined *chaining technique* [**Dud78**] can be used to remove the logarithm in the above bound:

THEOREM 7.2. *There exists an absolute constant $C > 0$, such that under the conditions of the preceding theorem, with probability one,*

$$R_n(\mathcal{F}(Z^n)) \le C\sqrt{\frac{V(\mathcal{F})}{n}}.$$

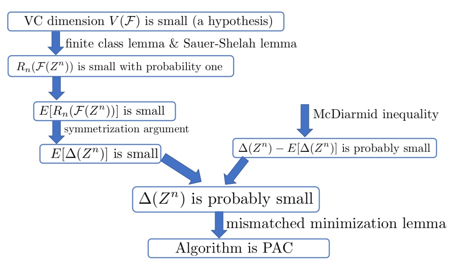


Fig. 3, p. 72

THEOREM 8.1. *(Performance bounds for concept learning by ERM) Consider an agnostic concept learning problem* $(\mathsf{X}, \mathcal{P}, \mathcal{C})$, *and let* $\delta > 0$. *For any* $P \in \mathcal{P}$, *the ERM algorithm satisfies*

$$L_P(\widehat{C}_n) \le L_P^*(\mathcal{C}) + 8\sqrt{\frac{V(\mathcal{C})\log(n+1)}{n}} + \sqrt{\frac{2\log(1/\delta)}{n}} \tag{8.1}$$

with probability at least $1-\delta$. *There is a universal constant* C *so that for any probability distribution* P *on* Z *and* $\delta \in (0,1)$, *the ERM algorithm satisfies*

$$L_P(\widehat{C}_n) \le L_P^*(\mathcal{C}) + C\sqrt{\frac{V(\mathcal{C})}{n}} + \sqrt{\frac{2\log(1/\delta)}{n}} \tag{8.2}$$

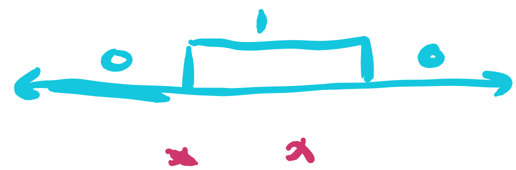
with probability at least $1-\delta$.

Shifting algorithm - used in proof of Sauer-Shelah lemma.

$$\begin{matrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{matrix} \qquad \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{matrix}$$

$$\begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad \begin{matrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

1	1	1	1	1
1	1	1	1	0
1	1	1	0	0
1	1	0	0	0
1	0	0	0	0
0	1	1	1	1
0	1	1	1	0
0	1	1	0	0
0	1	0	0	0
0	0	1	1	1
0	0	1	1	0
0	0	1	0	0
0	0	0	1	1
0	0	0	1	0
0	0	0	0	1
0	0	0	0	0
1	2	3	4	5



$$\binom{5}{\le 2} = \binom{5}{0} + \binom{5}{1} + \binom{5}{2}$$
$$= 1 + 5 + 10 = 16$$

Is $\{1,2\}$ shattered?

$$
\begin{matrix}
1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0
\end{matrix}
$$

- Algorithm terminates when set is downward directed.

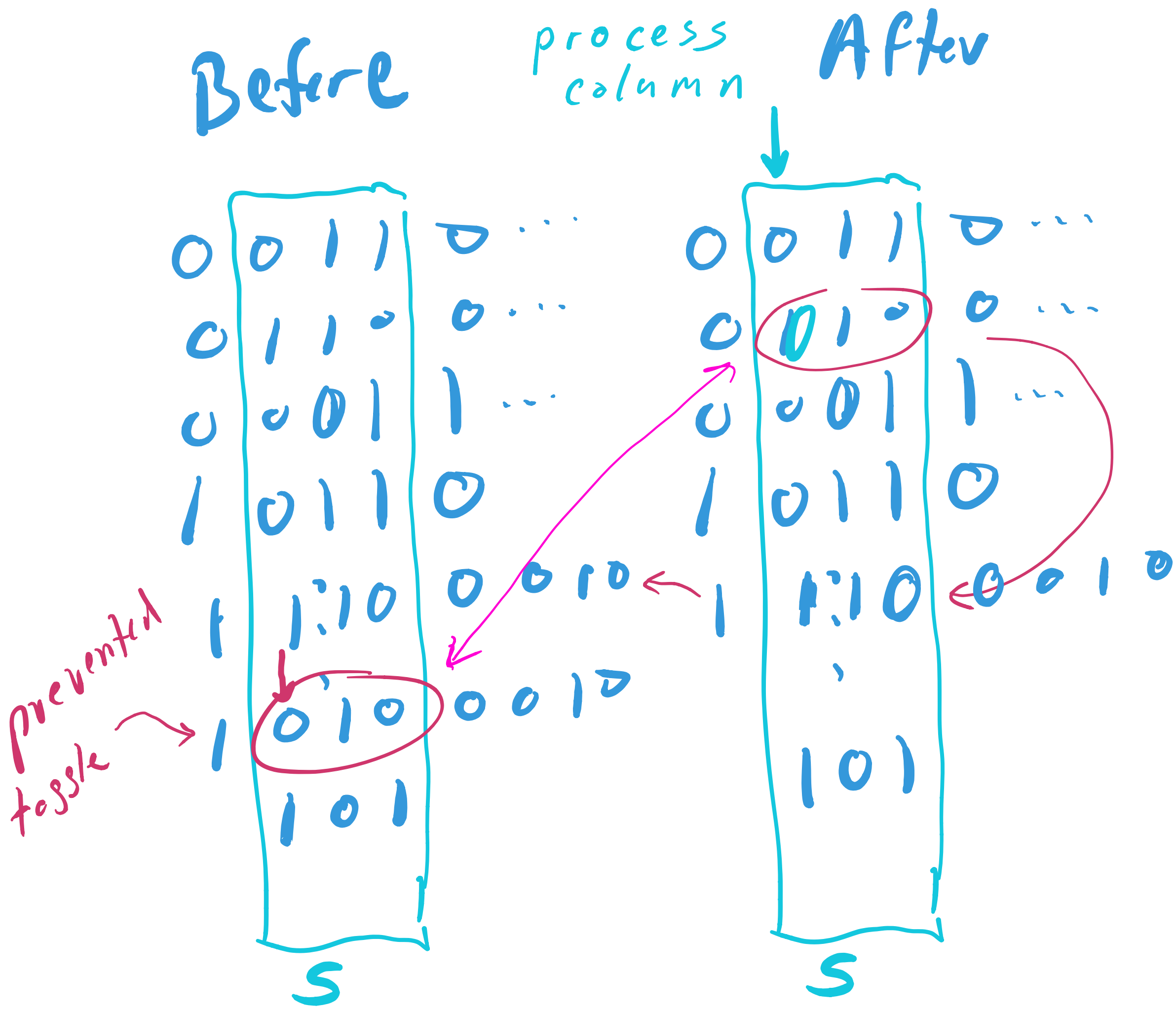
- # of rows is same as at beginning.

- If a set is shattered after alg. run, then it was shattered originally.

$\therefore$ # rows $\leq \binom{n}{\leq d}$

proof next that

" shifting does not create new shattered sets "


Before
process column
After
prevented toggle
S
S

Dudley class.

(Take $h=0$ for simplicity)

Z $\quad \psi_1, \dots, \psi_m$ - linearly independent functions on Z.

$$
\begin{aligned}
C_g &= \mathrm{pos}(g) \\
&= \mathrm{pos}\left(\sum_{i=1}^{m} c_i \psi_i(z)\right) \\
&= \left\{ z : \sum_{i=1}^{m} c_i \psi_i(z) \geq 0 \right\}
\end{aligned}
$$

Claim $V(\mathcal{G}) = m$

Converse select $z_1, \dots, z_{m+1} \in Z$

$$
\begin{pmatrix} b_1 \\ \vdots \\ \\ b_{m+1} \end{pmatrix}
=
\begin{array}{cc}
 & m \\
m+1 & \underbrace{\begin{pmatrix} \psi_1(z_1) & \cdots & \psi_m(z_1) \\ & & \\ & & \\ \psi_1(z_{m+1}) & \cdots & \psi_m(z_{m+1}) \end{pmatrix}}
\end{array}
\begin{pmatrix} c_1 \\ \vdots \\ \\ c_m \end{pmatrix}
$$

Range is a linear subspace of dimension $\leq m$ in $\mathbb{R}^{m+1}$