# CS340 Machine learning
# Lecture 4
# Learning theory

Some slides are borrowed from Sebastian Thrun and Stuart Russell

# Announcement

- What:  Workshop on applying for NSERC scholarships and for entry to graduate school
  When:  Thursday, Sept 14, 12:30-14:00
  Where:  DMP 110
  Who:  All Computer Science undergraduates expecting to graduate within the next 12 months who are interested in applying  to graduate school

# PAC Learning: intuition

- If we learn hypothesis h on the training data, how can be sure this is close to the true target function f if we don't know what f is?
- Any hypothesis that we learn but which is seriously wrong will almost certainly be "found out" with high probability after a small number of examples, because it will make an incorrect prediction.
- Thus any hypothesis that is consistent with a sufficiently large set of training examples is unlikely to be seriously wrong, i.e., it must be **probably approximately correct**.
- Learning theory is concerned with estimating the sample size needed to ensure good generalization performance.

# PAC Learning

- PAC = Probably approximately correct
- Let f(x) be the true class, h(x) our guess, and $\pi$(x) a distribution of examples. Define the error as

$$error(h) = p(h(x) \neq f(x) | x \text{ drawn from } \pi)$$

- Define h as *approximately correct* if error(h) < $\varepsilon$.
- Goal: find sample size m s.t. for any distribution $\pi$

$$\forall \pi . \forall X \sim \pi : |X| = m. \;\; p(error(h) > \epsilon | X) < \delta$$

- If Ntrain >= m, then with probability 1-$\delta$, the hypothesis will be approximately correct.
- Test examples must be drawn from same distribution as training examples.
- We assume there is no label noise.

# Derivation of PAC bounds for finite H

- Partition H into $H_\varepsilon$, an $\varepsilon$ "ball" around $f^{true}$, and $H_{bad} = H \setminus H_\varepsilon$
- What is the prob. that a "seriously wrong" hypothesis $h_b \in H_{bad}$ is consistent with m examples (so we are fooled)? We can use a union bound

$$\begin{aligned} error(h_b) &> \epsilon \\ p(h_b \text{ agrees with 1 example }) &\leq 1 - \epsilon \\ p(h_b \text{ agrees with m examples }) &\leq (1-\epsilon)^m \end{aligned}$$

The prob of finding such an $h_b$ is bounded by

$$\begin{aligned} p(H_{bad} \text{ contains a consistent hypothesis}) &\leq |H_{bad}|(1-\epsilon)^m \\ &\leq |H|(1-\epsilon)^n \end{aligned}$$

# Derivation of PAC bounds for finite H

- We want to find m s.t. $|H|(1-\epsilon)^m \leq \delta$
- This is called the sample complexity of H
- We use $1 - x \leq e^{-x}$ to derive

$$
\begin{aligned}
|H|e^{-m\epsilon} &\leq \delta \\
\log H - \log \delta &\leq m\epsilon \\
m &\geq \frac{1}{\epsilon}\left(\log \frac{1}{\delta} + \log |H|\right)
\end{aligned}
$$

- If |H| is larger, we need more training data to ensure we can choose the "right" hypothesis.

# PAC Learnability

- Statistical learning theory is concerned with sample complexity.
- Computational learning theory is additionally concerned with computational (time) complexity.
- A concept class C is PAC learnable, if it can be learnt with probability $\delta$ and error $\varepsilon$ in time polynomial in $1/\delta$, $1/\varepsilon$, $n$, and size(c).
- Implies
  - Polynomial sample complexity
  - Polynomial computational time

# H = any boolean function

- Consider all $2^{2^2} = 16$ possible binary functions on k=2 binary inputs

| $x_1$ | $x_2$ | $h_1$ | $h_2$ | $h_3$ | $h_4$ | $h_5$ | $h_6$ | $h_7$ | $h_8$ | $h_9$ | $h_{10}$ | $h_{11}$ | $h_{12}$ | $h_{13}$ | $h_{14}$ | $h_{15}$ | $h_{16}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

- If we observe ($x_1$=0, $x_2$=1, y=0), this removes $h_5$, $h_6$, $h_7$, $h_8$, $h_{13}$, $h_{14}$, $h_{15}$, $h_{16}$
- Each example halves the version space.
- Still leaves exponentially many hypotheses!

# H = any boolean function

- Unbiased Learner: $|H|=2^{2^k}$

$$m \geq \frac{1}{\varepsilon}(2^k \ln 2 + \ln(1/\delta))$$

• Needs exponentially large sample size to learn.
• Essentially has to learn whole lookup table, since for any unseen example, H contains as many consistent hypotheses that predict 1 as 0.

# Making learning tractable

- To reduce the sample complexity, and allow generalization from a finite sample, there are two approaches
  - Restrict the hypothesis space to simpler functions
  - Put a prior that encourages simpler functions
- We will consider the latter (Bayesian) approach later

# H = conjunction of boolean literals

- Conjunctions of Boolean literals:

$$h = x_1 \wedge \neg x_3 \wedge \cdots \wedge x_k$$

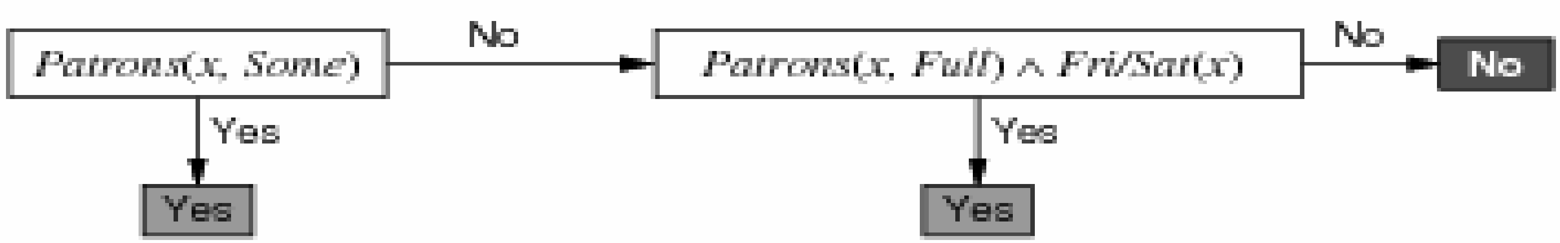
$|H|=3^k$

$$m \geq \frac{1}{\varepsilon}(k \ln 3 + \ln(1/\delta))$$

# H = decision lists

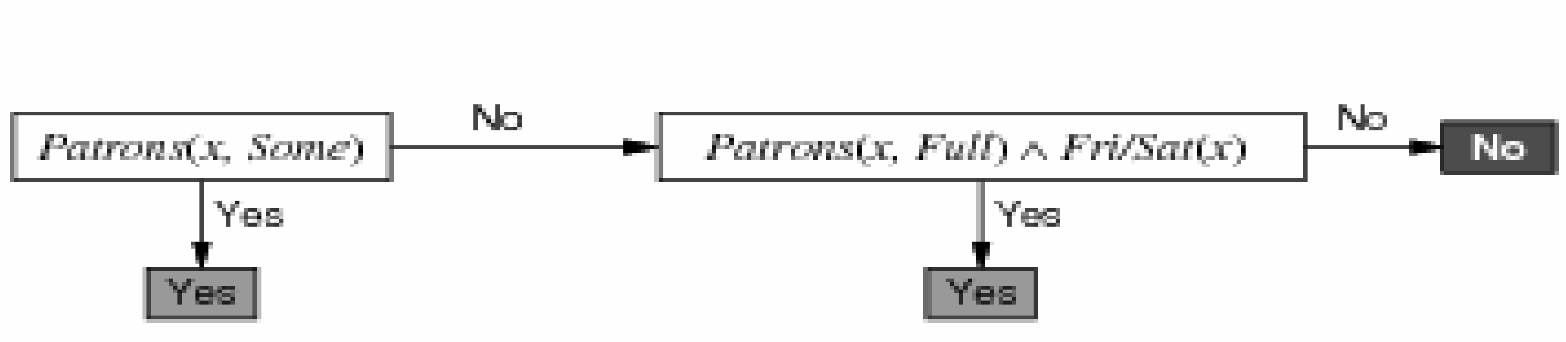
| Example | Attributes | | | | | | | | | | Target |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | *Alt* | *Bar* | *Fri* | *Hun* | *Pat* | *Price* | *Rain* | *Res* | *Type* | *Est* | *Wait* |
| $X_1$ | T | F | F | T | Some | $$$ | F | T | French | 0–10 | T |
| $X_2$ | T | F | F | T | Full | $ | F | F | Thai | 30–60 | F |
| $X_3$ | F | T | F | F | Some | $ | F | F | Burger | 0–10 | T |
| $X_4$ | T | F | T | T | Full | $ | F | F | Thai | 10–30 | T |
| $X_5$ | T | F | T | F | Full | $$$ | F | T | French | >60 | F |
| $X_6$ | F | T | F | T | Some | $$ | T | T | Italian | 0–10 | T |
| $X_7$ | F | T | F | F | None | $ | T | F | Burger | 0–10 | F |
| $X_8$ | F | F | F | T | Some | $$ | T | T | Thai | 0–10 | T |
| $X_9$ | F | T | T | F | Full | $ | T | F | Burger | >60 | F |
| $X_{10}$ | T | T | T | T | Full | $$$ | F | T | Italian | 10–30 | F |
| $X_{11}$ | F | F | F | F | None | $ | F | F | Thai | 0–10 | F |
| $X_{12}$ | T | T | T | T | Full | $ | F | F | Burger | 30–60 | T |

$$\forall x.\ \ WillWait(x) \Leftrightarrow Patrons(x, Some) \vee (Patrons(x, Full) \wedge Fri/Sat(x))$$



# H = decision lists

$$\forall x.\ WillWait(x) \Leftrightarrow Patrons(x, Some) \vee (Patrons(x, Full) \wedge Fri/Sat(x))$$




k-DL(n) restricts each test to contain at most k literals chosen from n attributes
k-DL(n) includes the set of all decision trees of depth at most k

$$|k - DL(n)| \leq 3^{|Conj(n,k)|}\ |Conj(n,k)|!$$

$$|Conj(n,k)| = \sum_{i=0}^{k} C_i^{2n} = O(n^k)$$

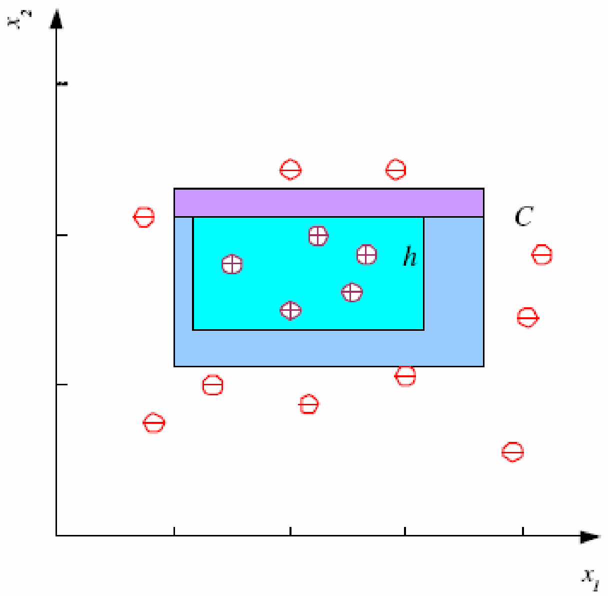
$$|k - DL(n)| = 2^{O(n^k \log_2(n^k))}$$

$$m \geq \frac{1}{\epsilon}\left(\log\frac{1}{\delta} + O(n^k \log_2 n^k)\right)$$

# PAC bounds for rectangles

- Let us consider an infinite hypothesis space, for which |H| is not defined.
- Let h be the most specific hypothesis, so errors occur in the purple strips.

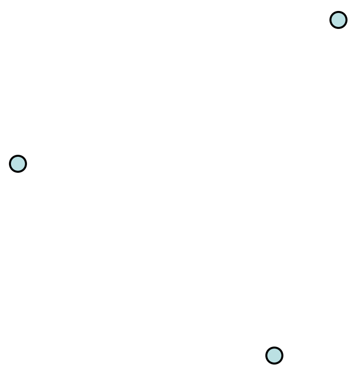
- Each strip is at most $\varepsilon/4$
- Pr that we miss a strip $1-\varepsilon/4$
- Pr that $N$ instances miss a strip $(1-\varepsilon/4)^N$
- Pr that $N$ instances miss 4 strips $4(1-\varepsilon/4)^N$
- $4(1-\varepsilon/4)^N \leq \delta$ and $(1-x) \leq \exp(-x)$
- $4\exp(-\varepsilon N/4) \leq \delta$ and $N \geq (4/\varepsilon)\log(4/\delta)$



# VC Dimension

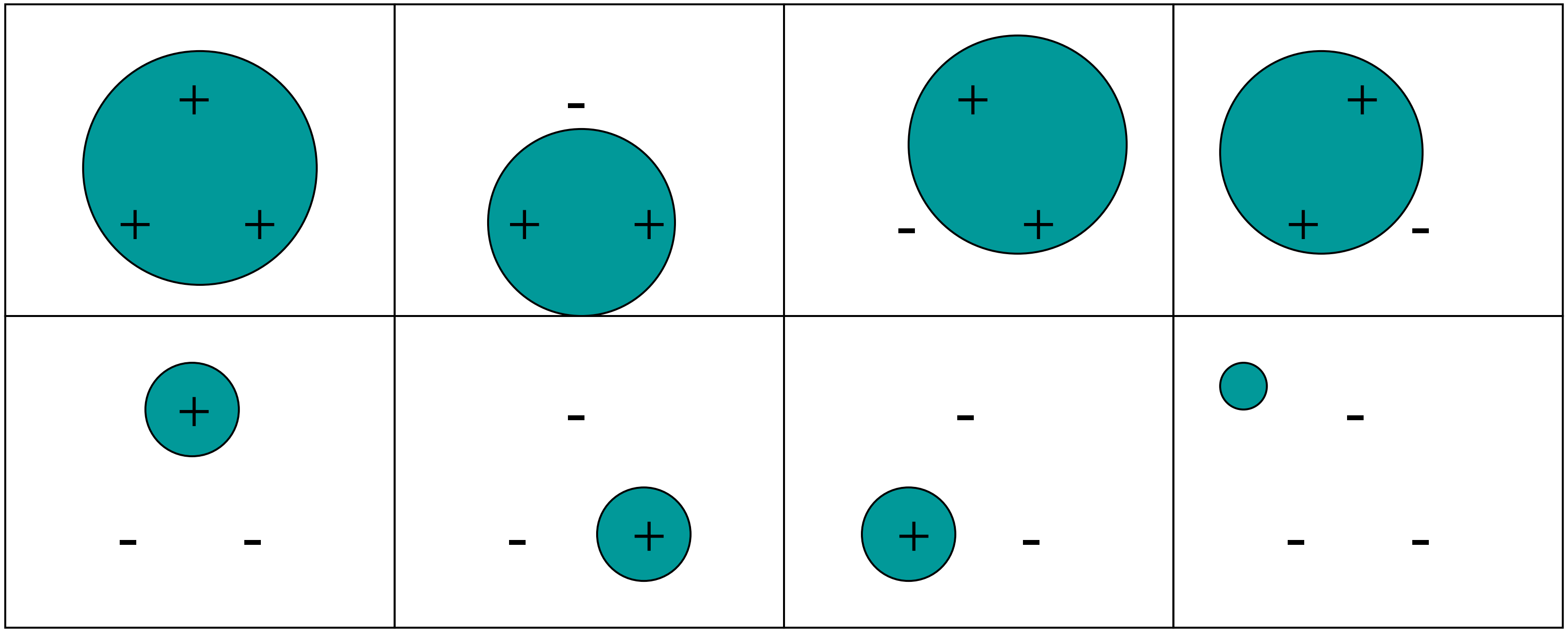
- We can generalize the rectangle example using the Vapnik-Chervonenkis dimension.
- VC(H) is the maximum number of points that can be shattered by H.
- A set of instances S is <u>shattered</u> by H if for every dichotomy (binary labeling) of S there is a consistent hypothesis in H.
- This is best explained by examples.

# Shattering 3 points in $R^2$ with circles



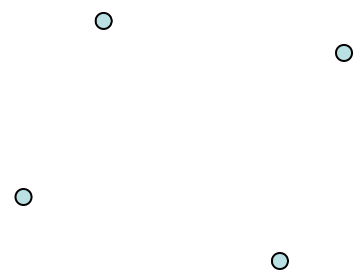
Is this set of points shattered by the hypothesis space H of all circles?

# Shattering 3 points in $R^2$ with circles

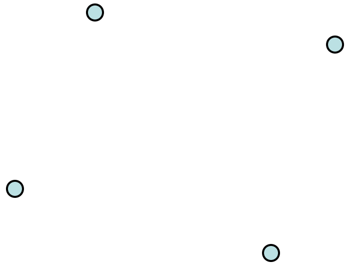


Every possible labeling can be covered by a circle, so we can shatter 3 points.

# Is this set of points shattered by circles?



# Is this set of points shattered by circles?



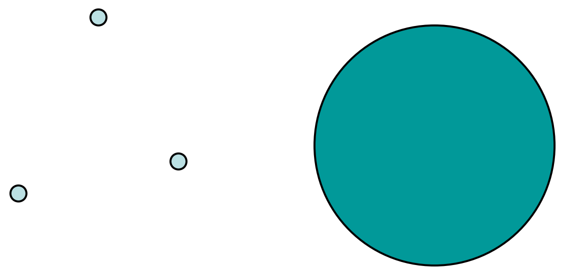
No,  we cannot shatter *any* set of 4 points.

# How About This One?

# How About This One?

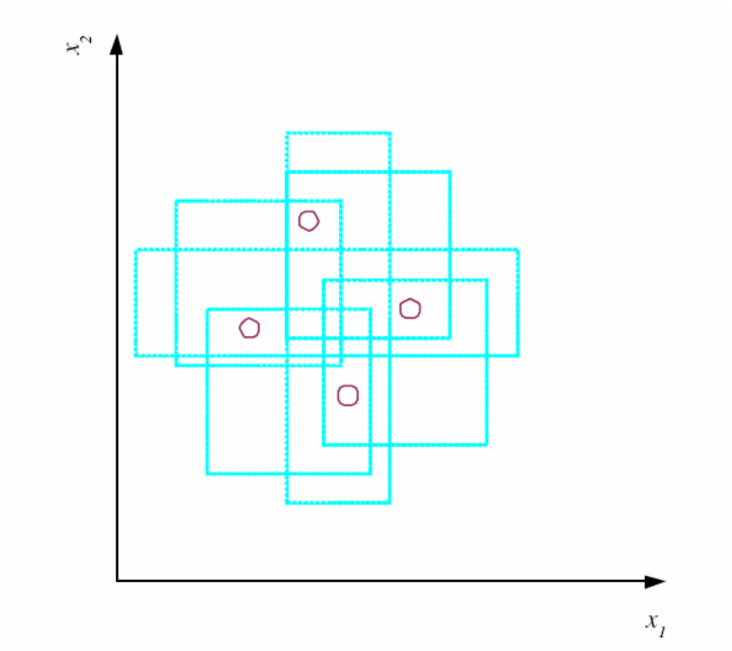
We cannot shatter this set of 3 points,
but we *can* find *some* set of 3 points which we can shatter

# VCD(Circles)=3



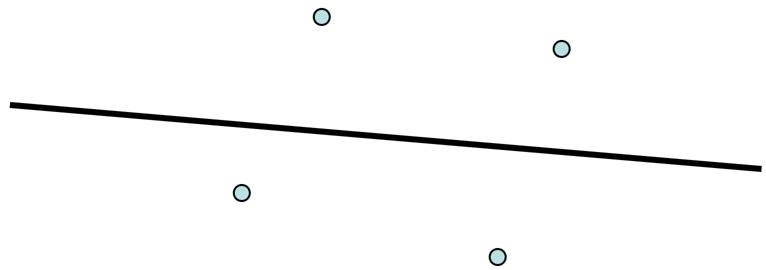
- VC(H) = 3, since 3 points can be shattered but not 4

# VCD(Axes-Parallel Rectangles) = 4



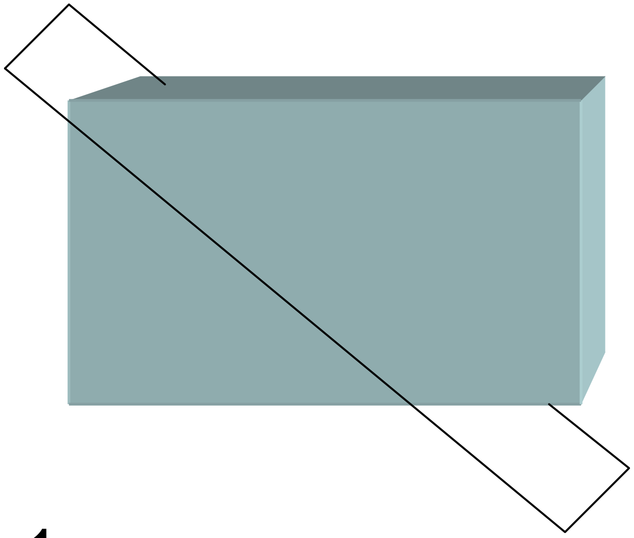

Can shatter at most 4 points in $R^2$ with a rectangle

# Linear decision surface in 2D



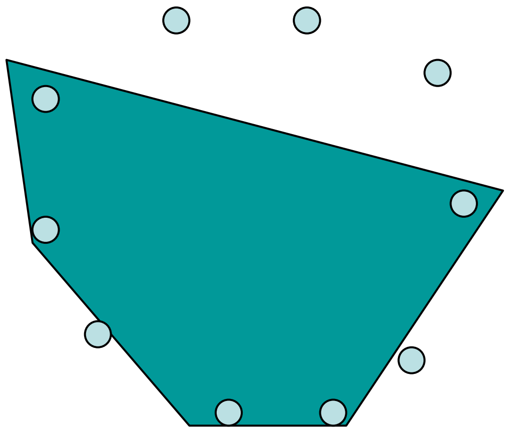
VC(H) = 3, so xor problem is not linearly separable

# Linear decision surface in n-d



VC(H) = n+1

# Is there an H with VC(H)=$\infty$ ?



Yes! The space of all convex polygons

# PAC-Learning with VC-dim.

- Theorem: After seeing

$$m \geq \frac{1}{\varepsilon}(4\log_2(2/\delta) + 8VC(H)\log_2(13/\varepsilon))$$

random training examples the learner will with probability 1-$\delta$ generate a hypothesis with error at most $\varepsilon$.

# Criticisms of PAC learning

- The bounds on the generalization error are very loose, because
  - they are distribution free/ worst case bounds, and do not depend on the actual observed data
  - they make various approximations
- Consequently the bounds are not very useful in practice.