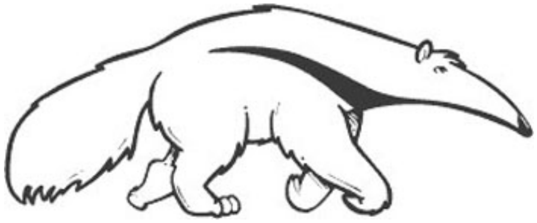
CS178: Machine Learning and Data Mining

VC Dimension

Prof. Erik Sudderth



Some materials courtesy Alex Ihler & Sameer Singh

BREN:ICS
INFORMATION AND COMPUTER SCIENCES

UNIVERSITY of CALIFORNIA IRVINE

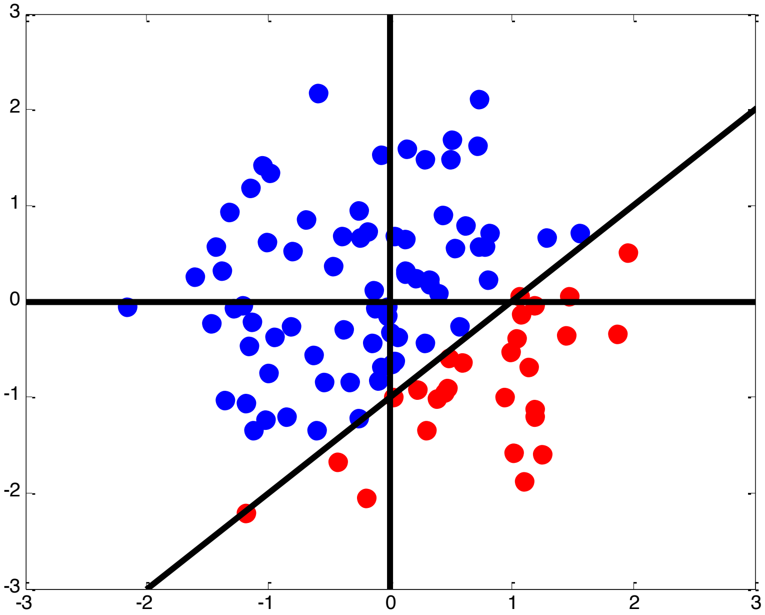
CS178 Zoom Lectures

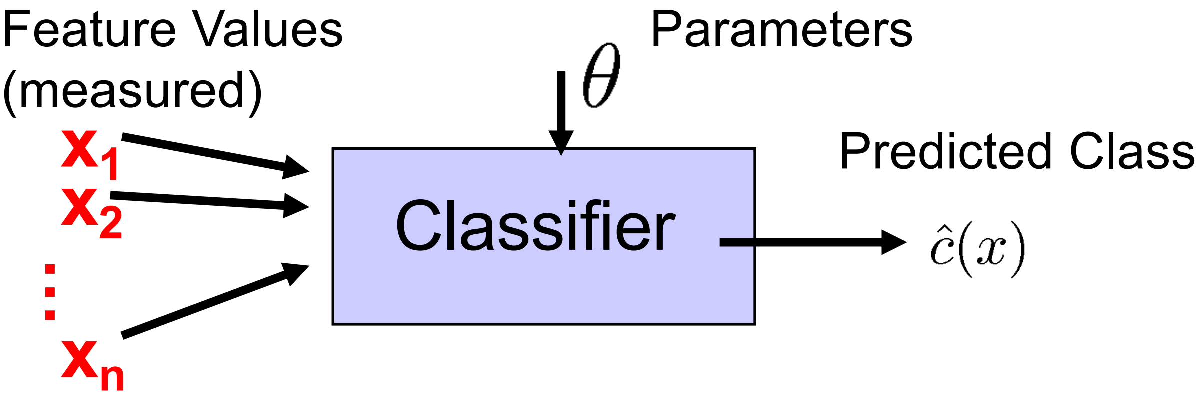
CS178 zoom lectures are recorded by the instructor (the recording feature is disabled for students). Recordings are posted to YuJa, and only available to CS178 students and staff. To ask questions during lecture, you may:

- Use the **Raise Hand** feature. Prof. Sudderth will then call on you by name, unmute your microphone, and let you ask a question. *Your question will be recorded. Please be respectful of your instructor and classmates.*
- Use the **Q&A Window** to type a question. Prof. Sudderth will read your question to the class before answering it, but *will not personally identify you.*

Learners and Complexity

- We’ve seen many versions of underfit/overfit trade-off
  - Complexity of the learner
  - “Representational Power”
- Different learners have different power




Feature Values (measured)

x_1

x_2

$\vdots$

x_n

Parameters

θ

Classifier

Predicted Class

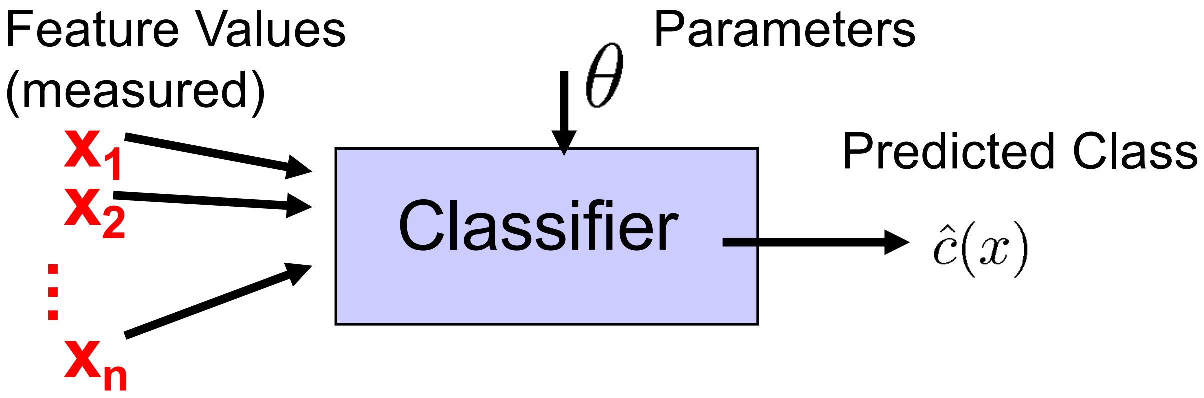
$\hat{c}(x)$

Example:

$$\hat{c}(x) = \text{sign}(\theta_1 x_1 + \theta_2 x_2 + \theta_0)$$

Learners and Complexity

- We've seen many versions of underfit/overfit trade-off
  - Complexity of the learner
  - "Representational Power"
- Different learners have different power



Feature Values (measured)

x_1

x_2

$\vdots$

x_n

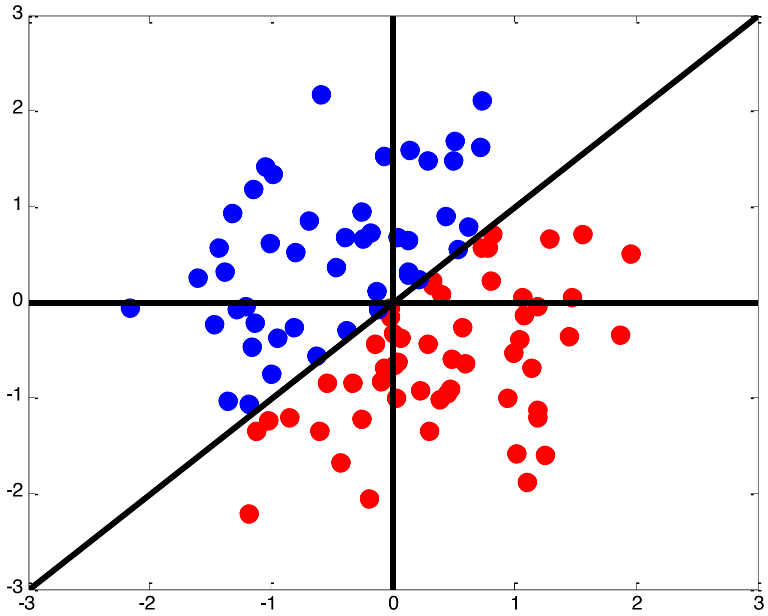
Parameters

θ

Classifier

Predicted Class

$\hat{c}(x)$

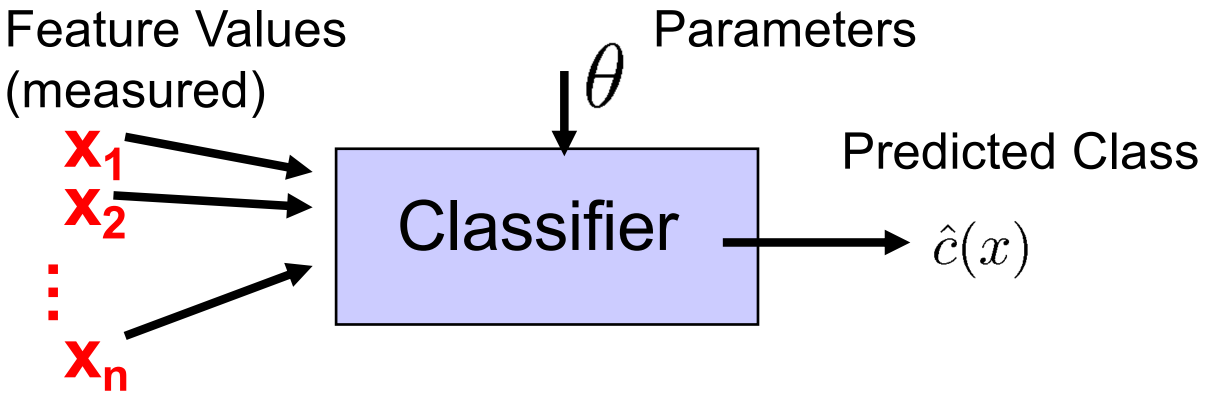


Example:

$$\hat{c}(x) = \text{sign}(\theta_1 x_1 + \theta_2 x_2)$$

Learners and Complexity

- We’ve seen many versions of underfit/overfit trade-off
  - Complexity of the learner
  - “Representational Power”
- Different learners have different power



Feature Values (measured)

x_1

x_2

$\vdots$

x_n

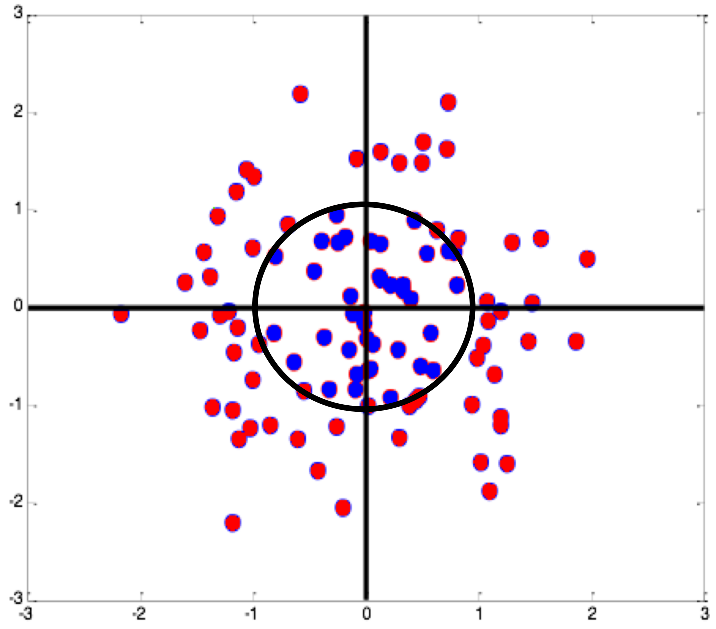
Parameters

θ

Classifier

Predicted Class

$\hat{c}(x)$



Example:

$$\hat{c}(x) = \text{sign}(\,(x_1^2 + x_2^2) - \theta_0)$$

Learners and Complexity

- We’ve seen many versions of underfit/overfit trade-off
  - Complexity of the learner
  - “Representational Power”
- Different learners have different power
- Usual trade-off:
  - More power = represent more complex systems, might overfit
  - Less power = won’t overfit, but may not find “best” learner
- How can we quantify representational power?
  - Not easily…
  - One solution is VC (Vapnik-Chervonenkis) dimension

Some notation

- Assume training data are iid (independent & identically distributed samples) from some distribution p(x,y)

- Define "risk" and "empirical risk"
  - These are just "long term" test and observed training error

$$R(\theta) = \mathrm{TestError} = \mathbb{E}[\mathbb{1}[c \neq \hat{c}(x\,;\,\theta)]]$$

$$R^{\mathrm{emp}}(\theta) = \mathrm{TrainError} = \frac{1}{m}\sum_i \mathbb{1}[c^{(i)} \neq \hat{c}(x^{(i)}\,;\,\theta)]$$

- How are these related? Depends on overfitting…
  - Underfitting domain: pretty similar…
  - Overfitting domain: test error might be lots worse!

VC Dimension and Risk

- Given some classifier, let H be its VC dimension
  - Represents "representational power" of classifier

$$R(\theta) = \text{TestError} = \mathbb{E}[\mathbb{1}[c \neq \hat{c}(x\,;\,\theta)]]$$

$$R^{\text{emp}}(\theta) = \text{TrainError} = \frac{1}{m}\sum_i \mathbb{1}[c^{(i)} \neq \hat{c}(x^{(i)}\,;\,\theta)]$$

- With "high probability" (1-η), Vapnik showed

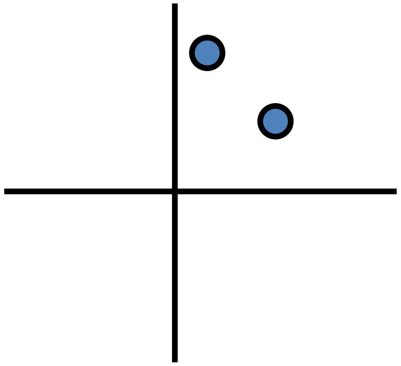
$$\boxed{\text{TestError} \leq \text{TrainError} + \sqrt{\frac{H\log(2m/H) + H - \log(\eta/4)}{m}}}$$

Shattering

- We say a classifier f(x) can *shatter* points $x^{(1)} \ldots x^{(h)}$ iff For *all* $y^{(1)} \ldots y^{(h)}$, f(x) can achieve zero error on training data $(x^{(1)},y^{(1)}), (x^{(2)},y^{(2)}), \ldots (x^{(h)},y^{(h)})$

  (i.e., there exists some θ that gets zero error)

- Can $f(x;\theta) = \text{sign}(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$ shatter these points?

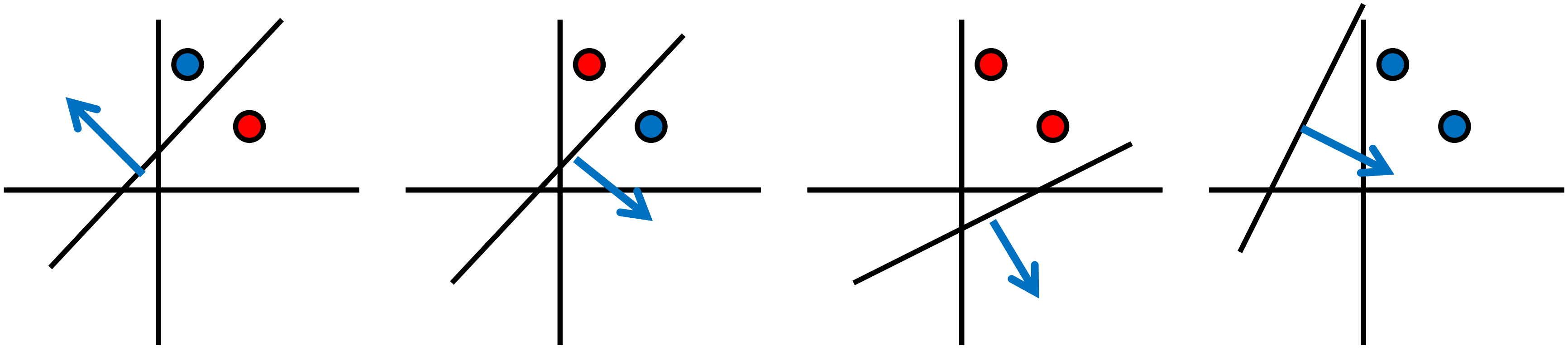


Shattering

- We say a classifier f(x) can *shatter* points $x^{(1)}\ldots x^{(h)}$ iff For *all* $y^{(1)}\ldots y^{(h)}$, f(x) can achieve zero error on training data $(x^{(1)},y^{(1)})$, $(x^{(2)},y^{(2)})$, … $(x^{(h)},y^{(h)})$

  (i.e., there exists some θ that gets zero error)

- Can f(x;θ) = sign($\theta_0 + \theta_1 x_1 + \theta_2$x2) shatter these points?
- Yes: there are 4 possible training sets…

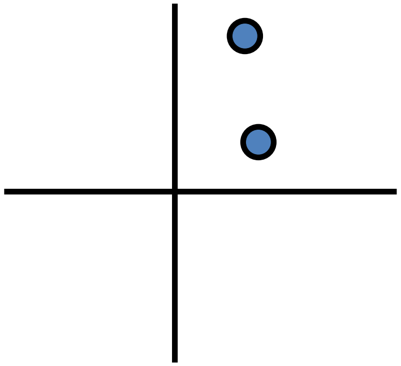


Shattering

- We say a classifier f(x) can *shatter* points $x^{(1)} \ldots x^{(h)}$ iff For *all* $y^{(1)} \ldots y^{(h)}$, f(x) can achieve zero error on training data $(x^{(1)},y^{(1)}), (x^{(2)},y^{(2)}), \ldots (x^{(h)},y^{(h)})$

  (i.e., there exists some θ that gets zero error)

- Can $f(x;\theta) = \text{sign}(x_1^2 + x_2^2 - \theta)$ shatter these points?

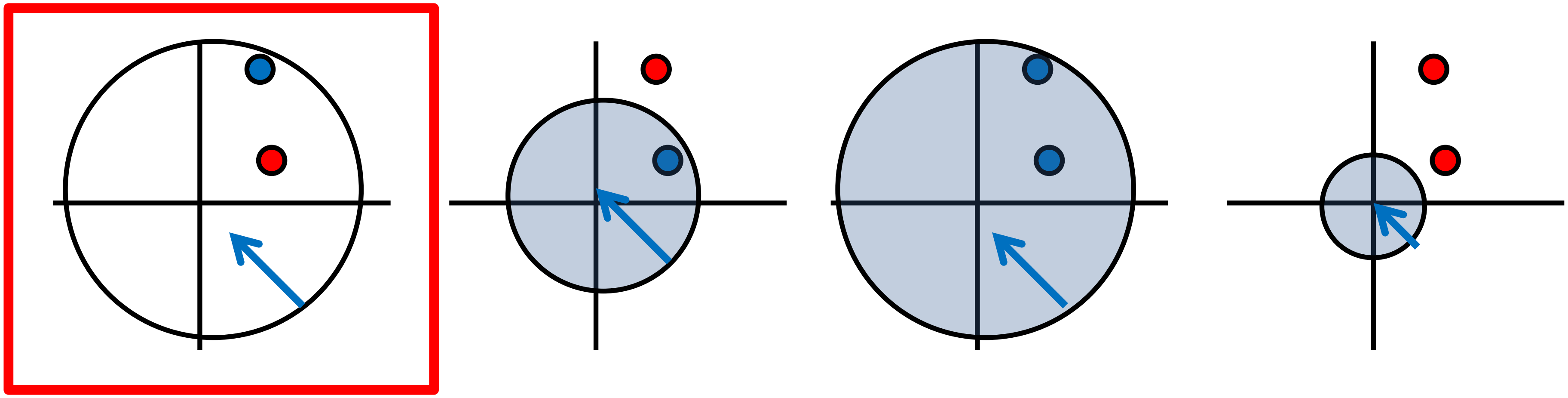


Shattering

- We say a classifier f(x) can *shatter* points $x^{(1)} \ldots x^{(h)}$ iff For *all* $y^{(1)} \ldots y^{(h)}$, f(x) can achieve zero error on training data $(x^{(1)},y^{(1)})$, $(x^{(2)},y^{(2)})$, … $(x^{(h)},y^{(h)})$

  (i.e., there exists some θ that gets zero error)

- Can $f(x;\theta) = \text{sign}(x_1^2 + x_2^2 - \theta)$ shatter these points?
- Nope!



VC Dimension

- The VC dimension H is defined as:

  The maximum number of points h that *can be arranged* so that f(x) can shatter them

- A game:
  - Fix the definition of f(x;θ)
  - Player 1: choose locations $x^{(1)}\ldots x^{(h)}$
  - Player 2: choose target labels $y^{(1)}\ldots y^{(h)}$
  - Player 1: choose value of θ
  - If f(x;θ) can reproduce the target labels, P1 wins

$$\exists\{x^{(1)}\ldots x^{(h)}\}\ s.t.\ \ \forall\{y^{(1)}\ldots y^{(h)}\}\ \ \exists\theta\ s.t.\ \ \forall i\ f(x^{(i)};\theta)=y^{(i)}$$

VC Dimension

- The VC dimension H is defined as:

  The maximum number of points h that *can be arranged* so that f(x) can shatter them

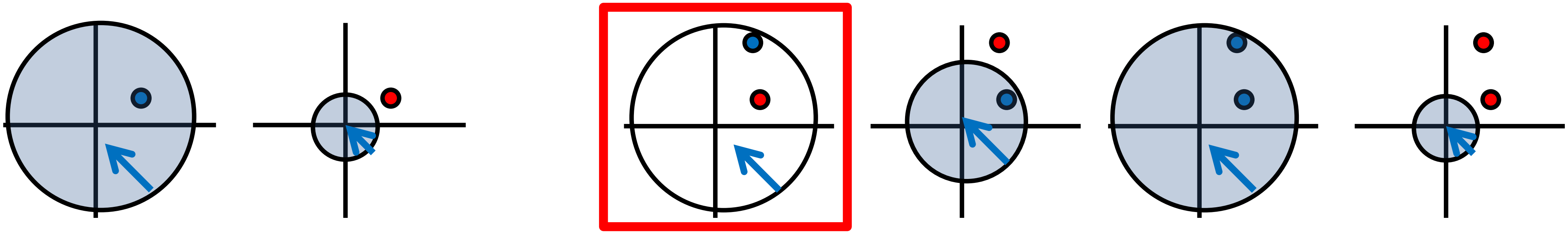
- Example: what's the VC dimension of the (zero-centered) circle, $f(x;\theta) = \text{sign}(x_1^2 + x_2^2 - \theta)$?

VC Dimension

- The VC dimension H is defined as

  The maximum number of points h that *can be arranged* so that f(x) can shatter them

- Example: what’ s the VC dimension of the (zero-centered) circle, $f(x;\theta) = \mathrm{sign}(x_1^2 + x_2^2 - \theta)$?
- VCdim = 1 : can arrange one point, cannot arrange two (previous example was general)



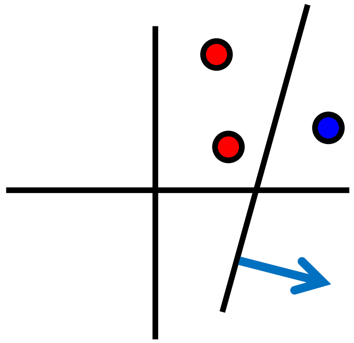
VC Dimension

- Example: what's the VC dimension of the two-dimensional line, $f(x;\theta) = \mathrm{sign}(\theta_1 x_1 + \theta_2 x_2 + \theta_0)$?

VC Dimension

- Example: what's the VC dimension of the two-dimensional line, $f(x;\theta) = \text{sign}(\theta_1 x_1 + \theta_2 x_2 + \theta_0)$?

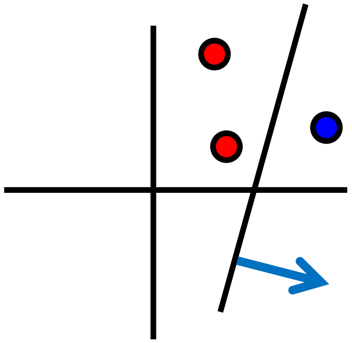
- VC dim >= 3? Yes



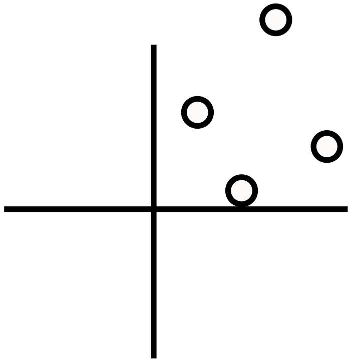
VC Dimension

- Example: what's the VC dimension of the two-dimensional line, f(x;θ) = sign(θ_1 x$_1$ + θ_2 x$_2$ + θ_0)?

- VC dim >= 3? Yes



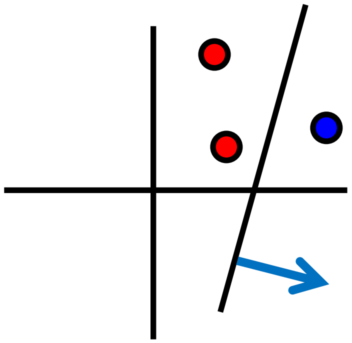
- VC dim >= 4?



VC Dimension

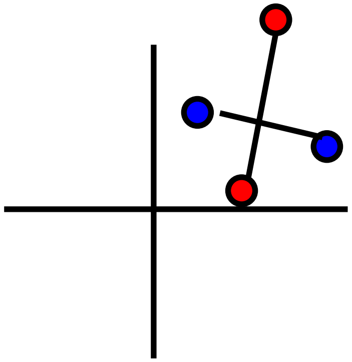
- Example: what's the VC dimension of the two-dimensional line, $f(x;\theta) = \text{sign}(\theta_1 x_1 + \theta_2 x_2 + \theta_0)$?

- VC dim >= 3? Yes



- VC dim >= 4? No…

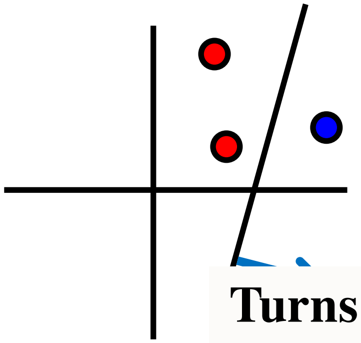
  Any line through these points must split one pair (by crossing one of the lines)



VC Dimension

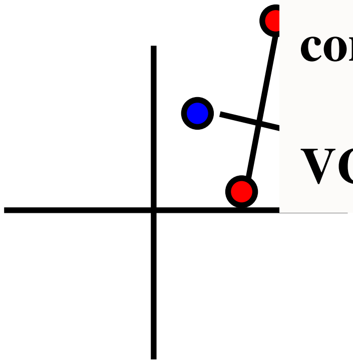
- Example: what’s the VC dimension of the two-dimensional line, $f(x;\theta) = \text{sign}(\theta_1 x_1 + \theta_2 x_2 + \theta_0)$?

- VC dim >= 3? Yes



- VC dim >= 4? No…

  Any line through these points must split one pair (by crossing one of the lines)



Turns out:
For a general , linear classifier (perceptron) in d dimensions with a constant term:

VC dim = d+1

VC Dimension

- VC dimension measures the “power” of the learner
- Does *not* necessarily equal the # of parameters!

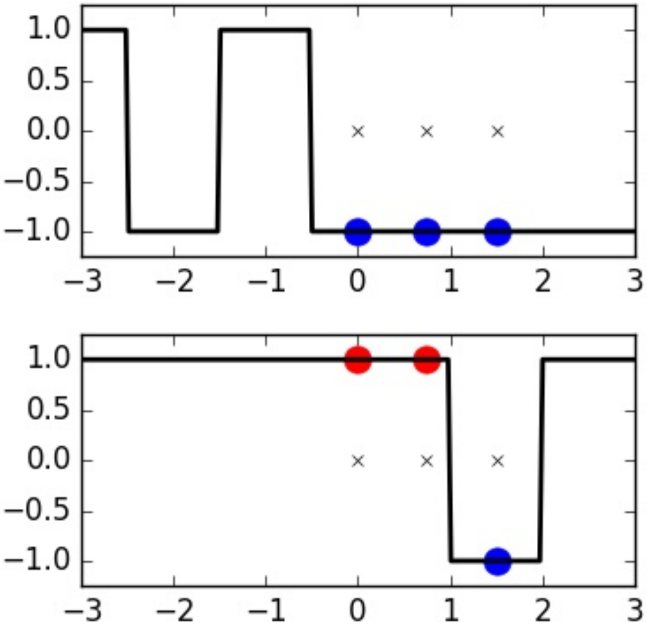
- Number of parameters does not necessarily equal complexity
  - Can define a classifier with a lot of parameters but not much power (how?)
  - Can define a classifier with one parameter but lots of power (how?)

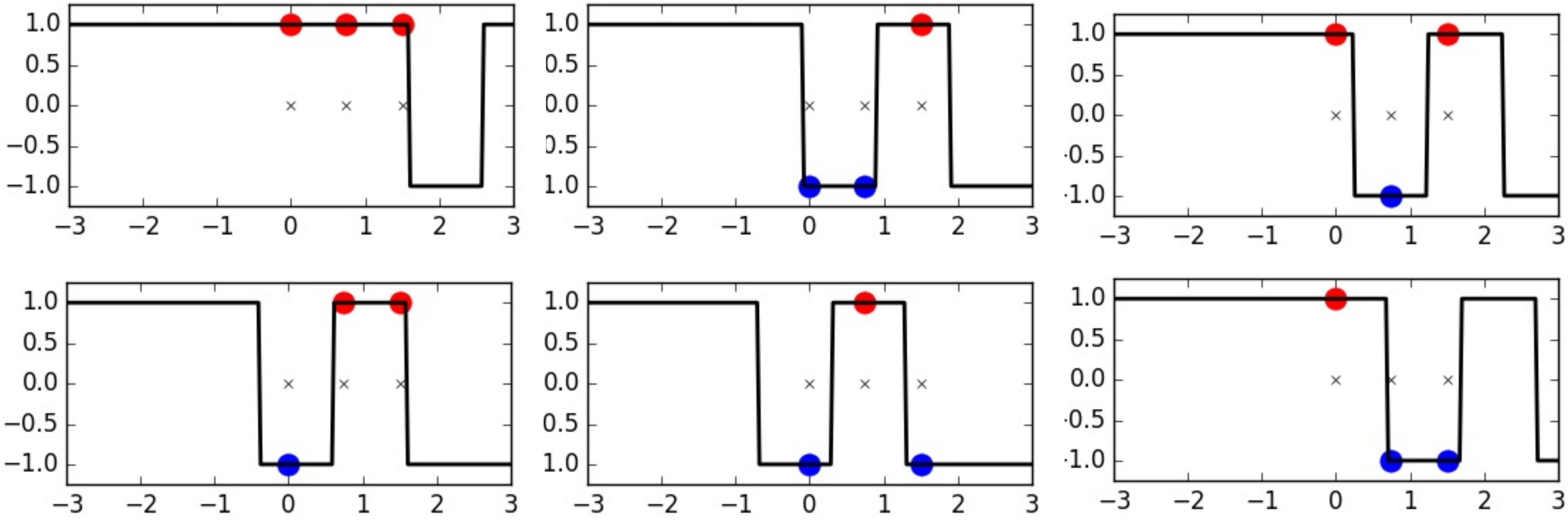
- Lots of work to determine what the VC dimension of various learners is…

Example

$$f(x;t) = \begin{cases} +1 & x \in [-\inf, t] \cup [t+1, t+2] \\ -1 & \text{otherwise} \end{cases}$$

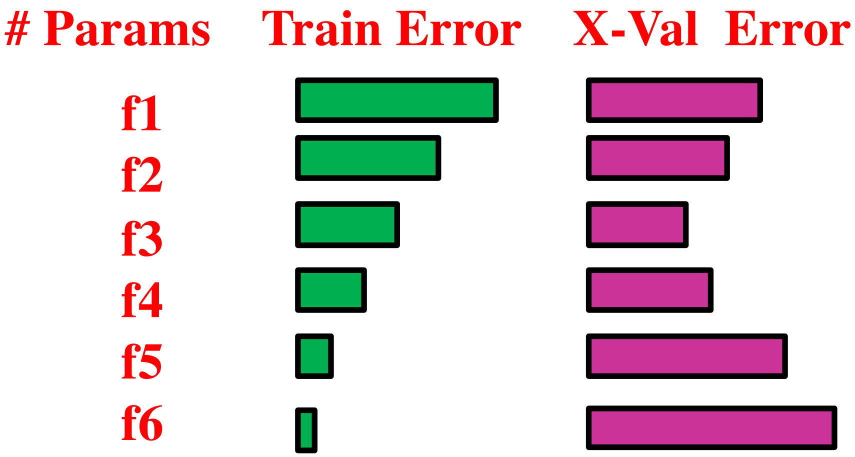
- VC Dim >= 3?
- VC Dim >= 4 ?





Using VC dimension

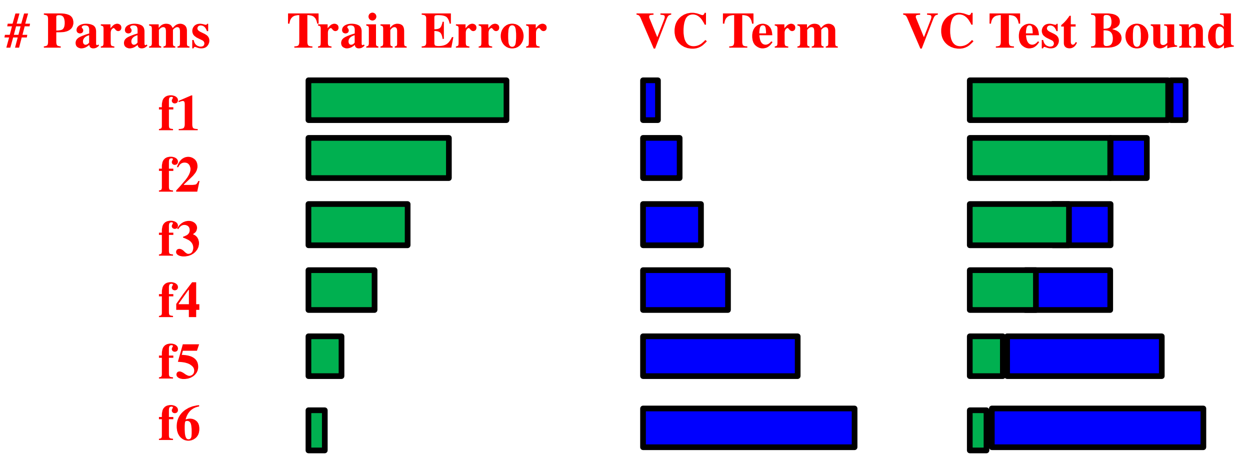
- Used validation / cross-validation to select complexity



Using VC dimension

- Used validation / cross-validation to select complexity
- Use VC dimension based bound on test error similarly

- “Structural Risk Minimization” (SRM)



Using VC dimension

- Used validation / cross-validation to select complexity
- Use VC dimension based bound on test error similarly

- Other Alternatives
  - Probabilistic models: likelihood under model (rather than classification error)
  - AIC (Aikike Information Criterion)
    - Log-likelihood of training data - # of parameters
  - BIC (Bayesian Information Criterion)
    - Log-likelihood of training data - (# of parameters)*log(m)
- Similar to VC dimension: performance + penalty

- BIC conservative; SRM very conservative
- Also, “true Bayesian” methods (take prob. learning…)