

+

# Machine Learning and Data Mining

## VC Dimension

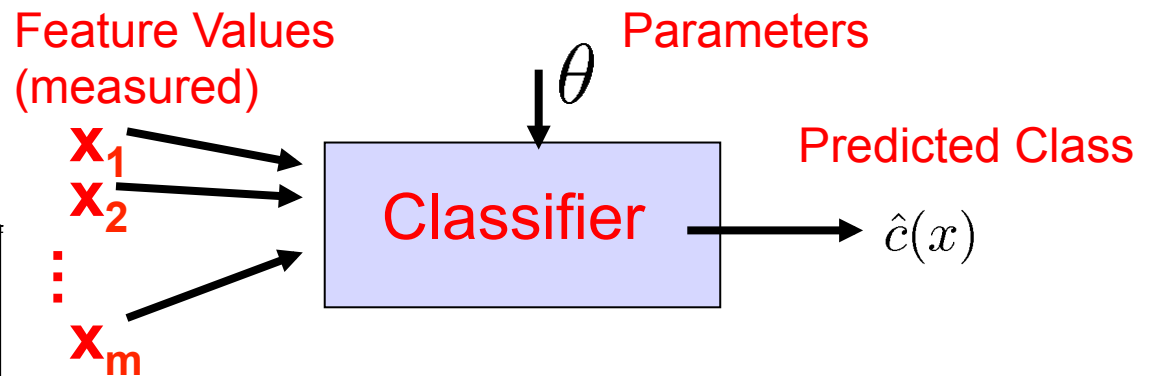
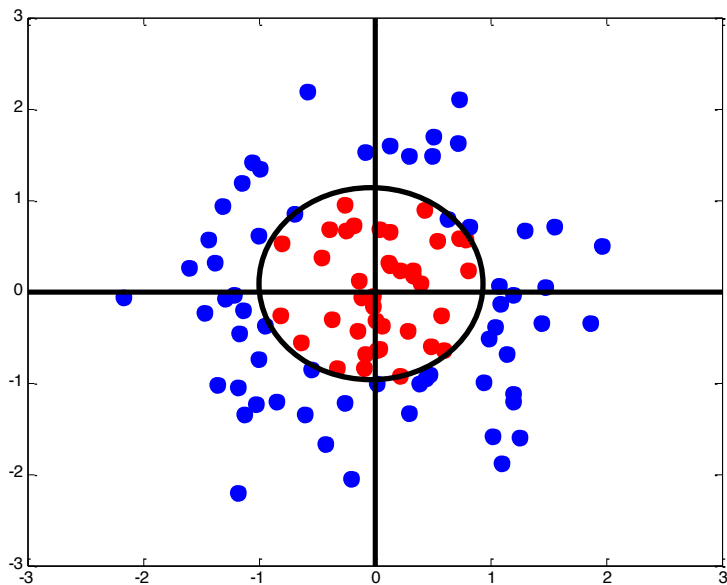
Prof. Alexander Ihler



**Slides based on Andrew Moore's**

# Learners and Complexity

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

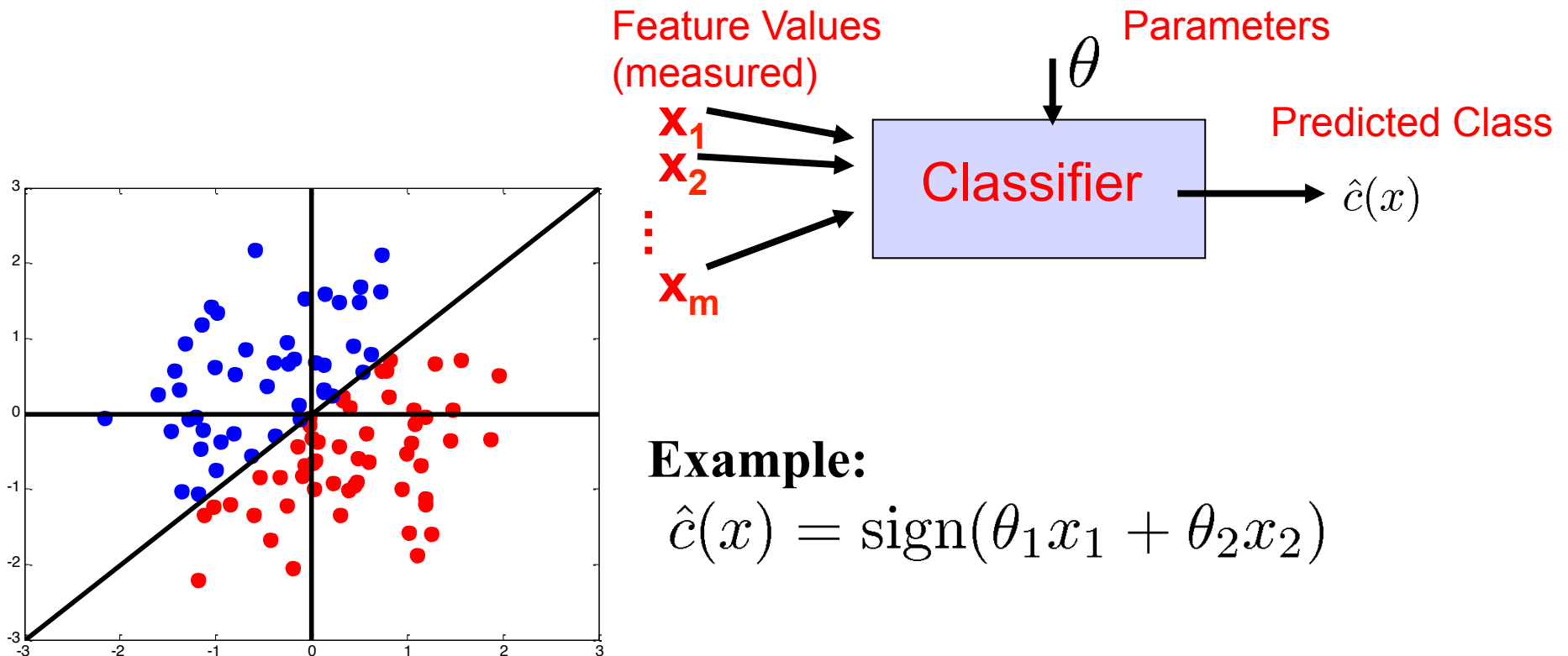


**Example:**

$$\hat{c}(x) = \text{sign}(x^T x - \theta_0)$$

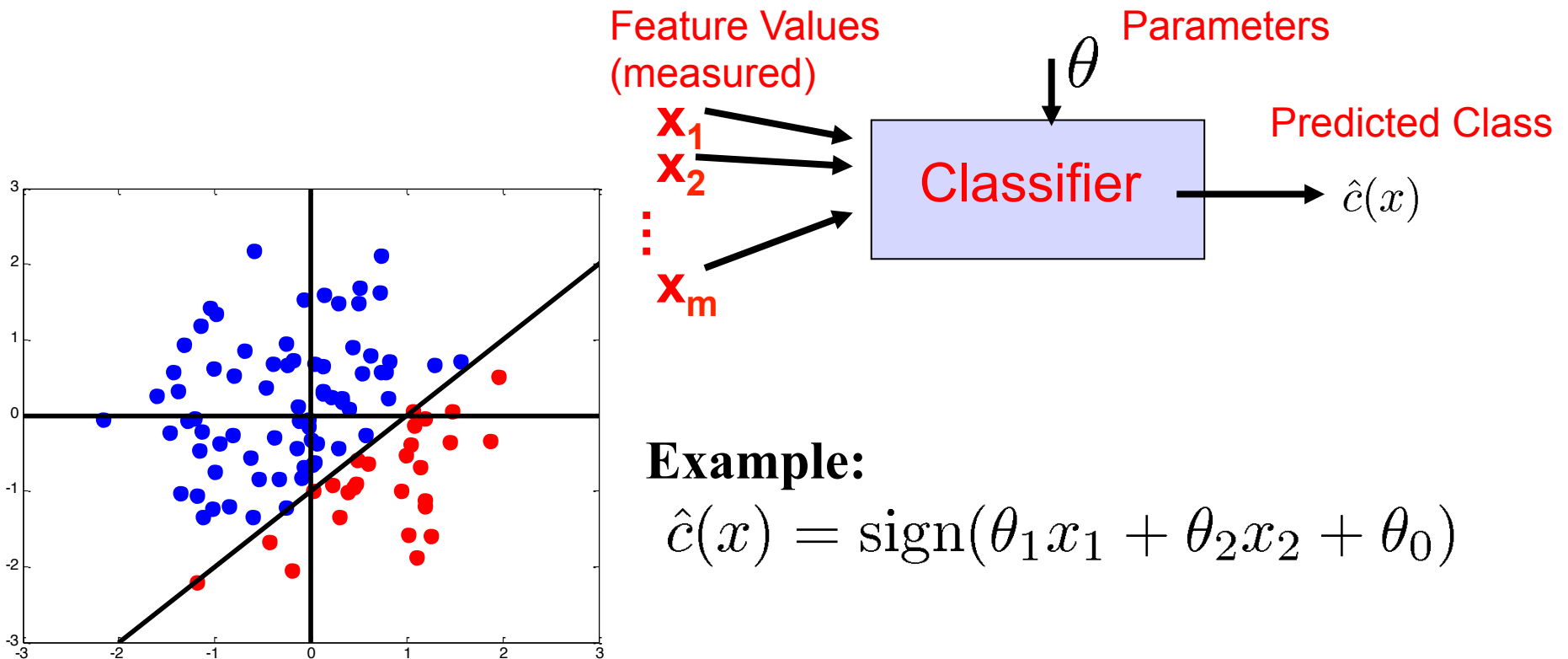
# Learners and Complexity

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



# Learners and Complexity

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



# 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

- Let's assume our training data are iid from some distribution  $p(x)$
- Define “risk” and “empirical risk”
  - These are just “long term” test and observed training error

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

$$R^{\text{emp}}(\theta) = \text{TrainError} = \frac{1}{N} \sum_i \delta(c^{(i)} \neq \hat{c}(x^{(i)}; \theta))$$

- 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

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

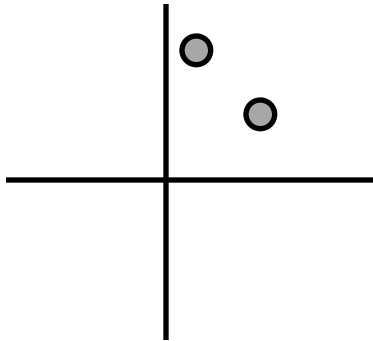
$$R^{\text{emp}}(\theta) = \text{TrainError} = \frac{1}{N} \sum_i \delta(c^{(i)} \neq \hat{c}(x^{(i)}; \theta))$$

- With “high probability”  $(1-\eta)$ , Vapnik showed

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

# Shattering

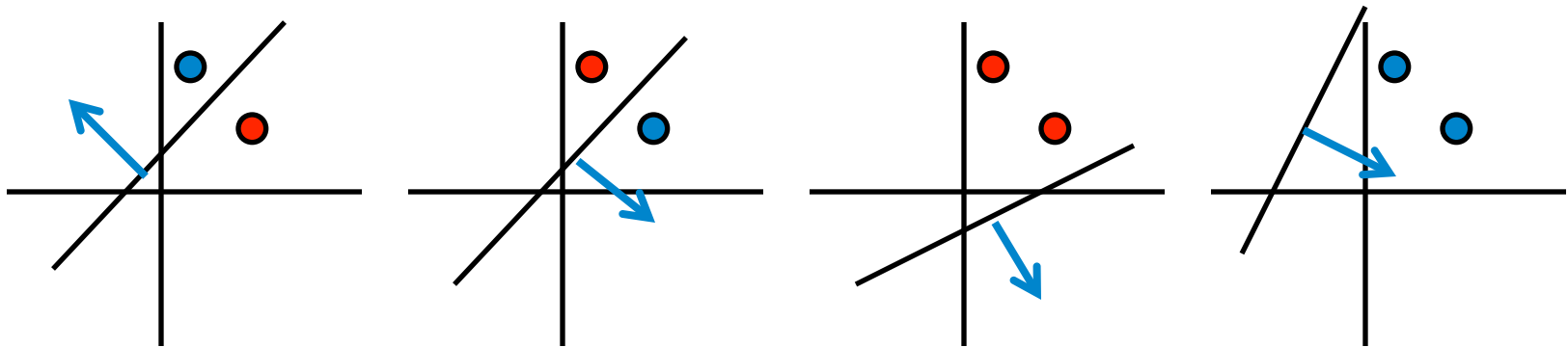
- We say a classifier  $f(x)$  can shatter points  $x_1 \dots x_N$  iff For *all*  $y_1 \dots y_N$ ,  $f(x)$  can achieve zero error on training data  $(x_1, y_1), (x_2, y_2), \dots (x_N, y_N)$  (i.e., there exists some  $\theta$  that gets zero error)
- Can  $f(x; \theta) = \text{sign}(\theta x^T)$  shatter these points?





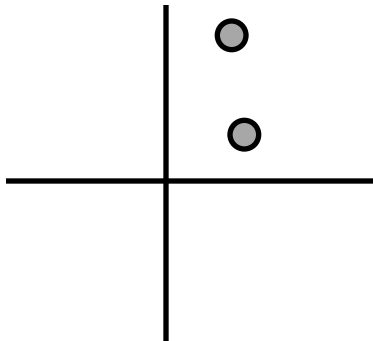
# Shattering

- We say a classifier  $f(x)$  can shatter points  $x_1 \dots x_N$  iff For *all*  $y_1 \dots y_N$ ,  $f(x)$  can achieve zero error on training data  $(x_1, y_1), (x_2, y_2), \dots (x_N, y_N)$  (i.e., there exists some  $\theta$  that gets zero error)
- Can  $f(x; \theta) = \text{sign}(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$  shatter these points?
- Yes: there are 4 possible training sets...



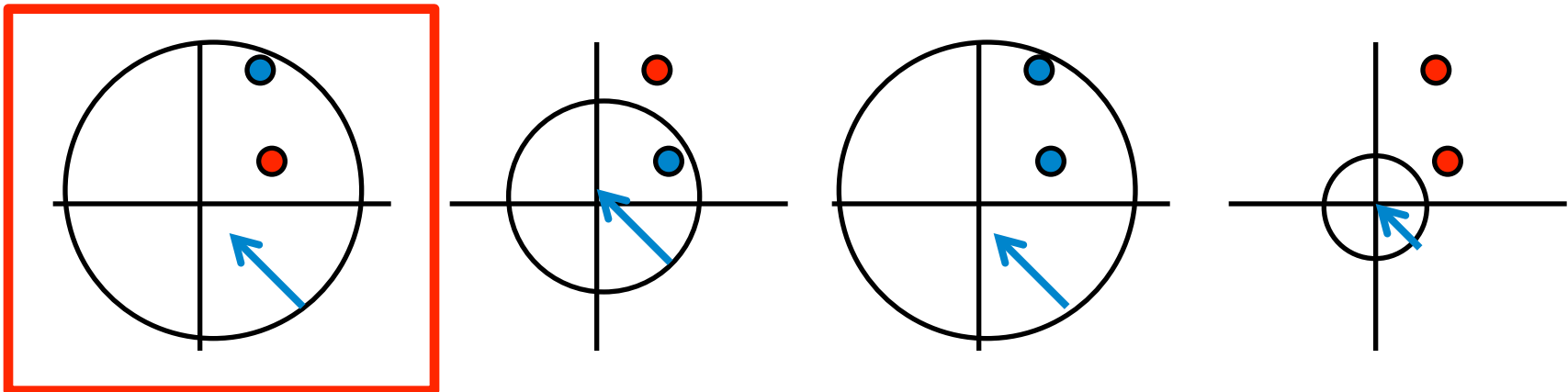
# Shattering

- We say a classifier  $f(x)$  can shatter points  $x_1 \dots x_N$  iff For *all*  $y_1 \dots y_N$ ,  $f(x)$  can achieve zero error on training data  $(x_1, y_1)$ ,  $(x_2, y_2)$ , ...  $(x_N, y_N)$  (i.e., there exists some  $\theta$  that gets zero error)
- Can  $f(x; \theta) = \text{sign}(x^T x + \theta)$  shatter these points?



# Shattering

- We say a classifier  $f(x)$  can shatter points  $x_1 \dots x_N$  iff For *all*  $y_1 \dots y_N$ ,  $f(x)$  can achieve zero error on training data  $(x_1, y_1), (x_2, y_2), \dots (x_N, y_N)$  (i.e., there exists some  $\theta$  that gets zero error)
- Can  $f(x; \theta) = \text{sign}(x^T x + \theta)$  shatter these points?
- Nope!

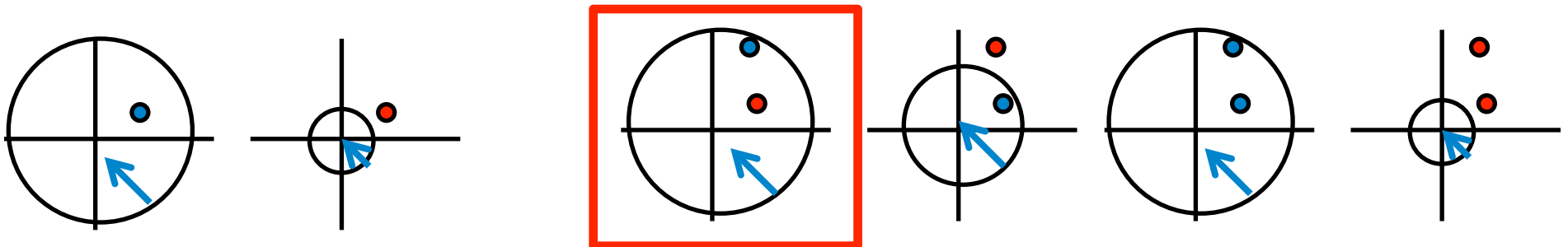


# VC Dimension

- The VC dimension is defined as  
The maximum number of points that *can be arranged*  
so that  $f(x)$  can shatter them
- Example: what's the VC dimension of the (zero-centered) circle,  $f(x; \theta) = \text{sign}(x^T x + \theta)$  ?

# VC Dimension

- The VC dimension is defined as  
The maximum number of points that *can be arranged* so that  $f(x)$  can shatter them
- Example: what's the VC dimension of the (zero-centered) circle,  $f(x;\theta) = \text{sign}(x^T x + \theta)$  ?
- VCdim = 1 : can arrange one point, cannot arrange two (previous example was general)



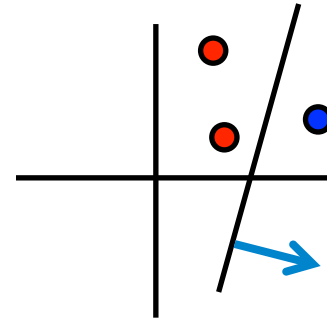
# VC Dimension

---

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

# VC Dimension

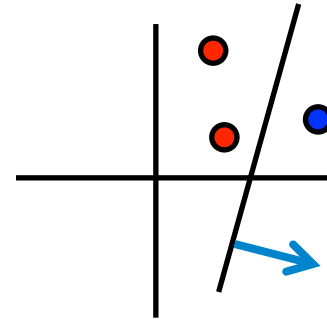
- Example: what's the VC dimension of the two-dimensional line,  $f(x;\theta) = \text{sign}(\theta_1 x_1 + \theta_2 x_2 + \theta_0)$ ?
- VC dim  $\geq 3$ ? Yes



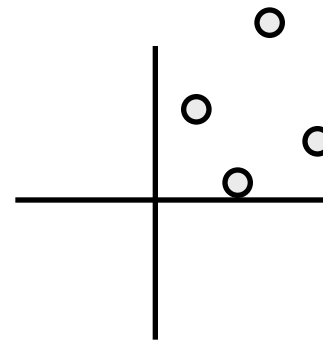
# VC Dimension

- 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  $\geq 3$ ? Yes



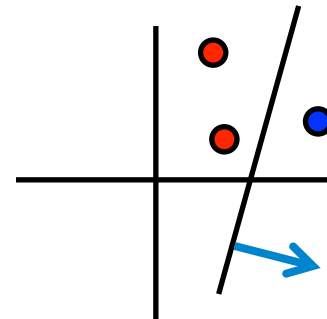
- VC dim  $\geq 4$ ?



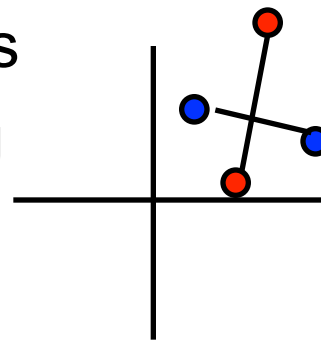


# VC Dimension

- 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  $\geq 3$ ? Yes

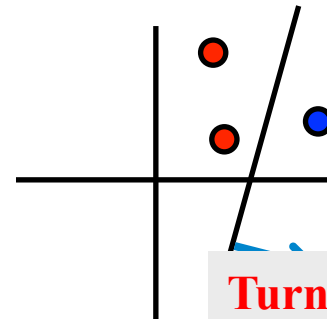


- VC dim  $\geq 4$ ? No...  
Any line through these points must split one pair (by crossing one of the lines)

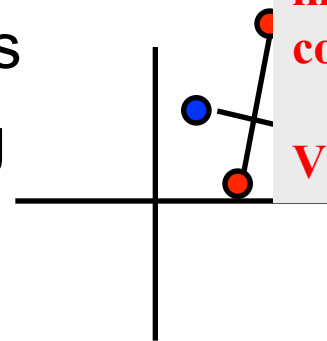


# VC Dimension

- 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  $\geq 3$ ? Yes



- VC dim  $\geq 4$ ? 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...

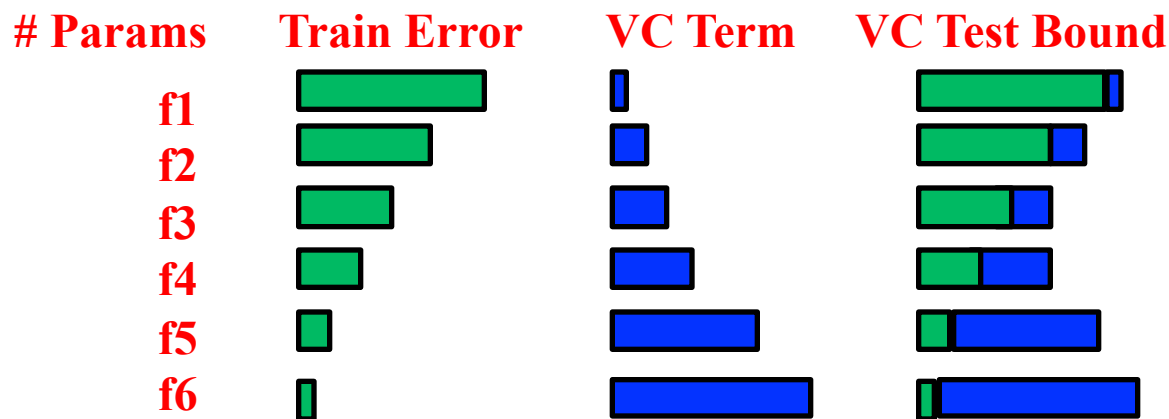
# Using VC dimension

- Recall how we used validation data, or cross-validation error rates to select a complexity



# Using VC dimension

- Recall how we used validation data, or cross-validation error rates to select a complexity
- Use VC dimension based bound on test error similarly
- “Structural Risk Minimization” (SRM)



# Using VC dimension

- Recall how we used validation data, or cross-validation error rates to select a 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(N)
- Similar to VC dimension: performance + penalty
- BIC conservative; SRM very conservative
- Also, “true Bayesian” methods (take prob. learning...)