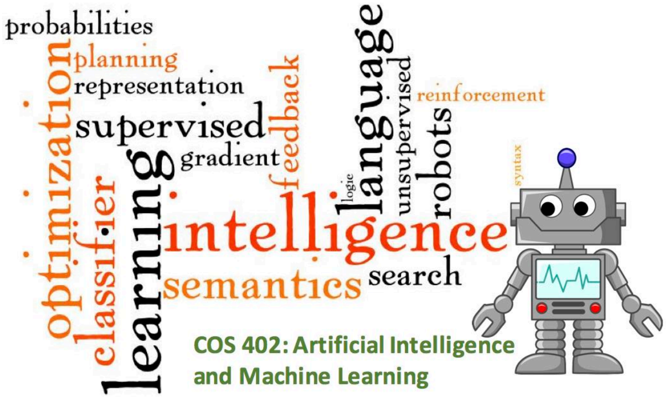


COS 402 – Machine Learning and Artificial Intelligence
Fall 2016

Lecture 4: Learning Theory (cont.) and optimization (start)

Sanjeev Arora

Elad Hazan

PRINCETON UNIVERSITY

Admin

- Exercise 1 – due today
- Exercise 2 (implementation) next Tue, in class
- Enrolment…
- Late policy (exercises)
- Literature (free!)

Recap

Last lecture:

- Shift from AI by introspection (Naïve methods) to statistical/computational learning theory
- Fundamental theorem of statistical learning
- Sample complexity, overfitting, generalization

Agenda

- (Review) statistical & computational learning theories for learning from examples
- (Review) Fundamental theorem of statistical learning for finite hypothesis classes
- The role of optimization in learning from examples
- Linear classification and the perceptron
- SVM and convex relaxations

Definition: learning from examples w.r.t. hypothesis class

A learning problem: $L = (X, Y, c, \ell, H)$

- X = Domain of examples (emails, pictures, documents, …)
- Y = label space (for this talk, binary Y={0,1})
- D = distribution over (X,Y) (the world)
- Data access model: learner can obtain i.i.d samples from D
- Concept = mapping $c: X \mapsto Y$
- Loss function $\ell: (Y, Y) \mapsto R$, such as $\ell(y_1, y_2) = 1_{y_1 \neq y_2}$
- H = class of hypothesis: $H \subseteq \{X \mapsto Y\}$
- Goal: produce hypothesis h$\in H$ with low *generalization error*

$$err(h) = E_{(x,y)\sim D}\,[\ell(h(x), c(x))]$$

agnostic PAC learnability

Learning problem $L = (X, Y, H, c, \ell)$ is agnostically PAC-learnable if there exists a learning algorithm s.t. for every $\delta, \epsilon > 0$, there exists $\mathrm{m} = f(\epsilon, \delta, H) < \infty$, s.t. after observing S examples, for $|S| = m$, returns a hypothesis $h \in H$, such that with probability at least

$$1 - \delta$$

it holds that

$$err(h) \leq \min_{h^* \in H} err(h^*) + \epsilon$$

Equals zero for realizable concepts

The meaning of learning from examples

Theorem:

Every realizable learning problem $L = (X, Y, H, c, \ell)$ for finite H, is PAC-learnable with sample complexity $S = O\left(\frac{\log H + \log\frac{1}{\delta}}{\epsilon}\right)$ using the ERM algorithm.

Noam chomesky, June 2011:

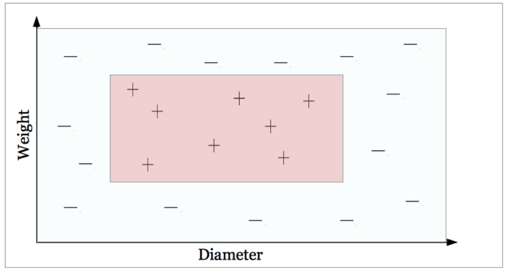
"It's true there's been a lot of work on trying to apply statistical models to various linguistic problems. I think there have been some successes, but a lot of failures. There is a notion of success ... **which I think is novel in the history of science. It interprets success as approximating unanalyzed data**."

Examples – statistical learning theorem

Theorem:

Every realizable learning problem $L = (X, Y, H, c, \ell)$ for finite H, is PAC-learnable with sample complexity $S = O\left(\frac{\log H + \log\frac{1}{\delta}}{\epsilon}\right)$ using the ERM algorithm.

- Apple factory: Wt. is measured in grams, 100-400 scale.
  Diameter: centimeters, 3-20



- Spam classification using decision trees of size 20 nodes
  - 200K words

Infinite hypothesis classes

VC-dimension: corresponding to "effective size" of hypothesis class (infinite or finite)

- Finite classes, $VCdim(H) = \log H$
- Axis-aligned rectangles in R^d , $VCdim(H) = O(d)$
- Hyperplanes in R^d , $VCdim(H) = d + 1$
- Polygons in the plane, $VCdim(H) = \infty$

Fundamental theorem of statistical learning:

A realizable learning problem $L = (X, Y, H, c, \ell)$ is PAC-learnable if an only if its VC dimension is finite, in which it is learnable with sample complexity $S = O\left(\frac{\mathrm{Vcdim}(H) \log\frac{1}{\delta}}{\epsilon}\right)$ using the ERM algorithm.

Overfitting??

sample complexity $S = O\left(\frac{\log H + \log\frac{1}{\delta}}{\epsilon}\right)$

It is tight!

Reminder: classifying fuel efficiency

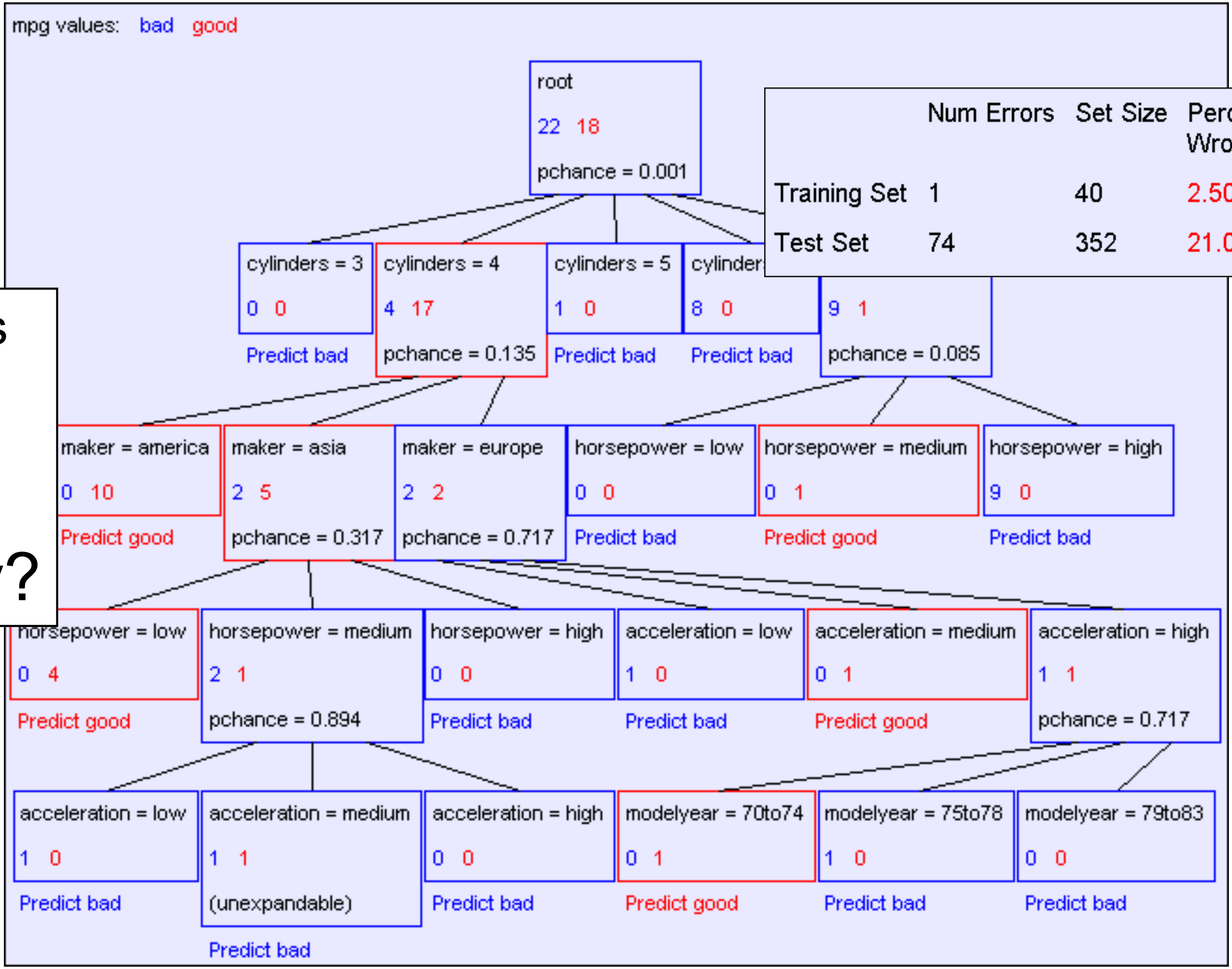
- 40 data points
- Goal: predict MPG

mpg	cylinders	displacement	horsepower	weight	acceleration	modelyear	maker
good	4	low	low	low	high	75to78	asia
bad	6	medium	medium	medium	medium	70to74	america
bad	4	medium	medium	medium	low	75to78	europe
bad	8	high	high	high	low	70to74	america
bad	6	medium	medium	medium	medium	70to74	america
bad	4	low	medium	low	medium	70to74	asia
bad	4	low	medium	low	low	70to74	asia
bad	8	high	high	high	low	75to78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
bad	8	high	high	high	low	70to74	america
good	8	high	medium	high	high	79to83	america
bad	8	high	high	high	low	75to78	america
good	4	low	low	low	low	79to83	america
bad	6	medium	medium	medium	high	75to78	america
good	4	medium	low	low	low	79to83	america
good	4	low	low	medium	high	79to83	america
bad	8	high	high	high	low	70to74	america
good	4	low	medium	low	medium	75to78	europe
bad	5	medium	medium	medium	medium	75to78	europe

Y X

The test set error is much worse than the training set error…

…why?



	Num Errors	Set Size	Percent Wrong
Training Set	1	40	2.50
Test Set	74	352	21.02

$$t = \sin(2\pi x) + \epsilon$$

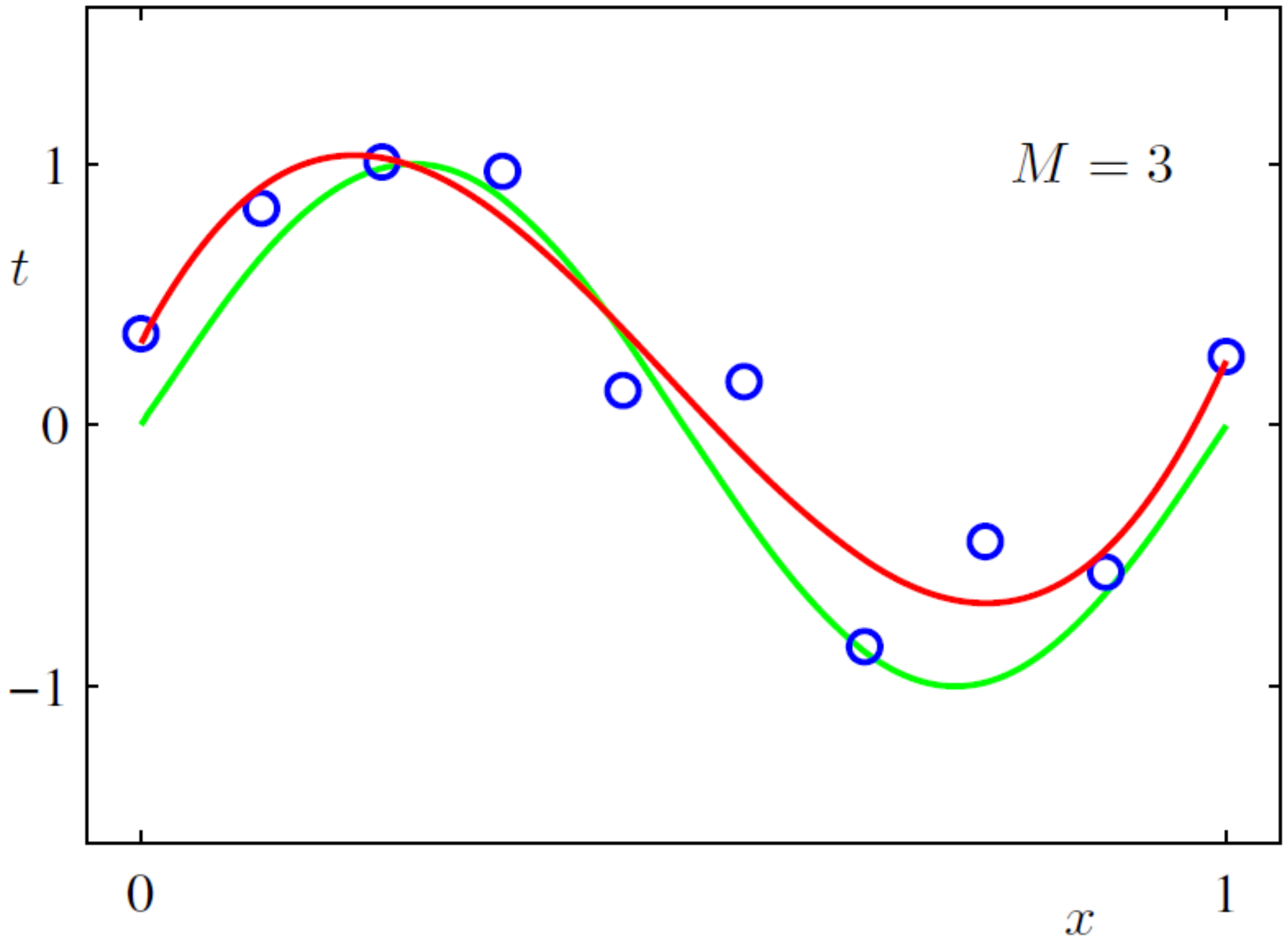


Figure from *Machine Learning and Pattern Recognition*, Bishop

$$t = \sin(2\pi x) + \epsilon$$

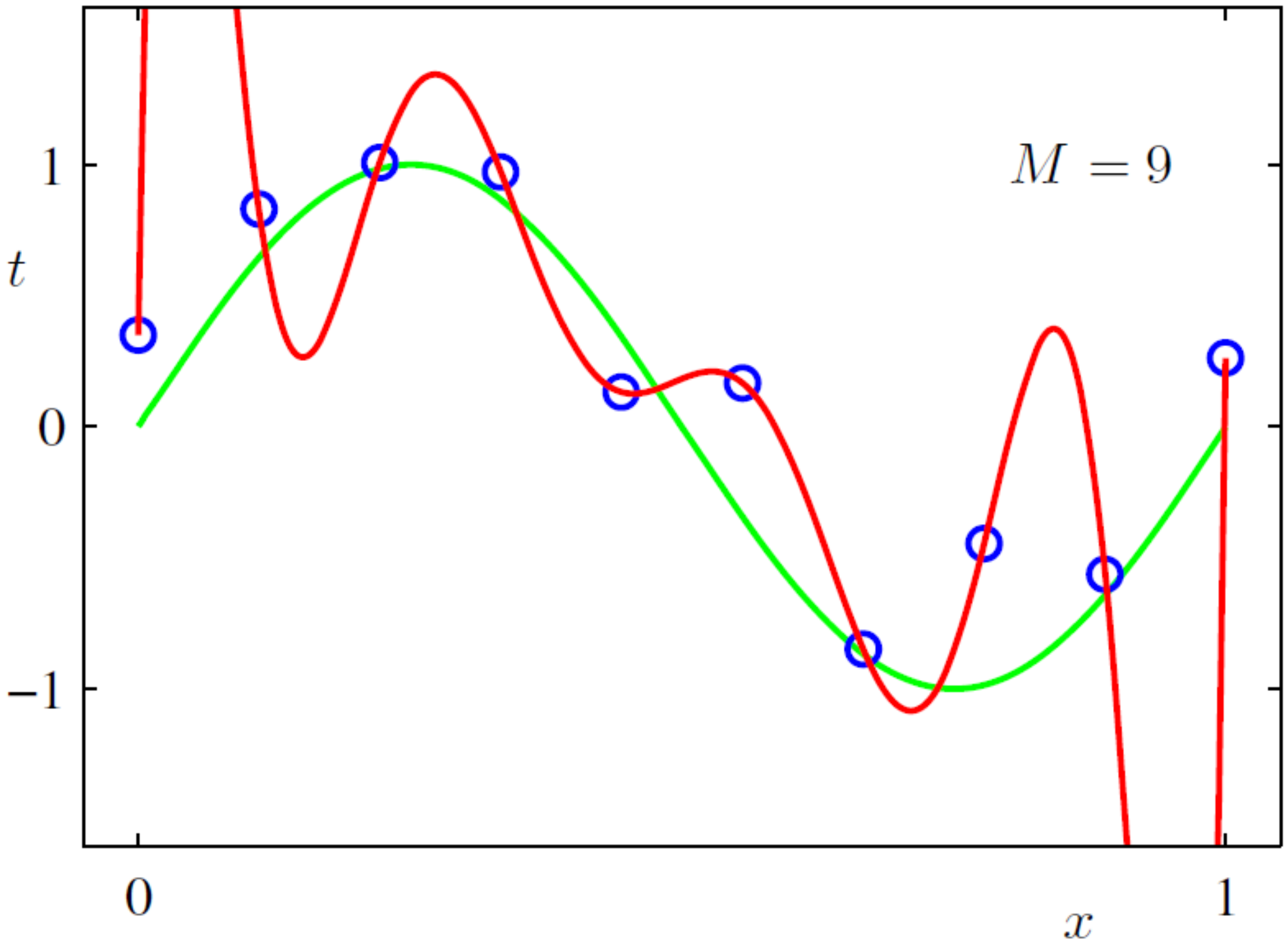


Figure from *Machine Learning and Pattern Recognition*, Bishop

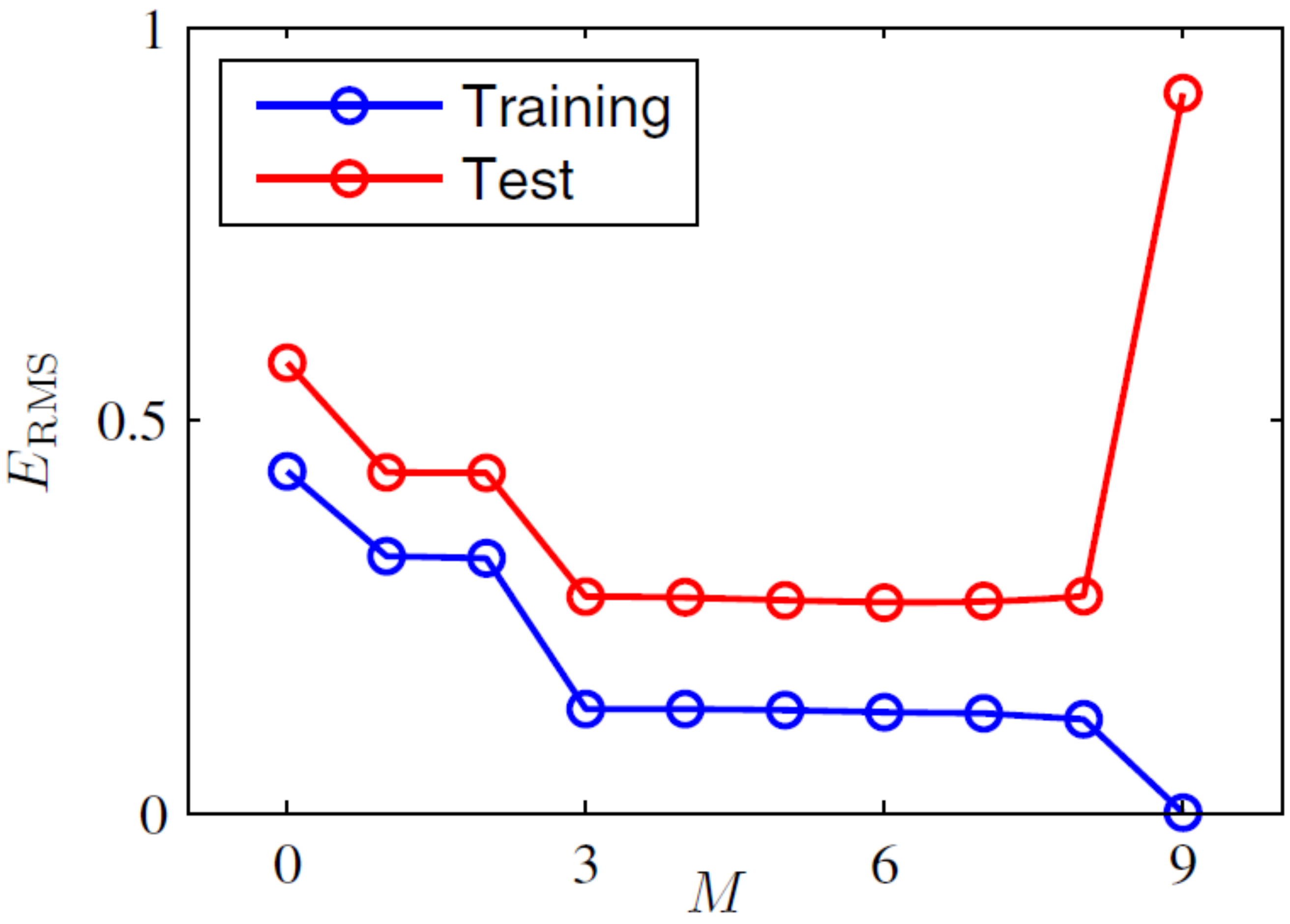


Figure from *Machine Learning and Pattern Recognition*, Bishop

Occam's razor

William of Occam c. 1287 – 1347:

controversial theologian: "plurality should not be posited without necessity",

i.e. "the simplest explanation is best"

Theorem:

Every realizable learning problem $L = (X, Y, H, c, \ell)$ for finite H, is PAC-learnable with sample complexity $S = O\left(\frac{\log H + \log\frac{1}{\delta}}{\epsilon}\right)$ using the ERM algorithm.

Other hypothesis classes?

1. Python programs of <= 10000 words

   $$|\mathrm{H}| \approx 10000^{10000}$$

   → sample complexity is $O\left(\frac{\log H + \log\frac{1}{\delta}}{\epsilon}\right) = O(\frac{50K}{\epsilon})$ - not too bad!

2. Efficient algorithm?
3. (Halting problem…)

4. The MAIN issue with PAC learning is computational efficiency!
5. Next topic: MORE HYPOTHESIS CLASSES that permit efficient OPTIMIZATION

Boolean hypothesis

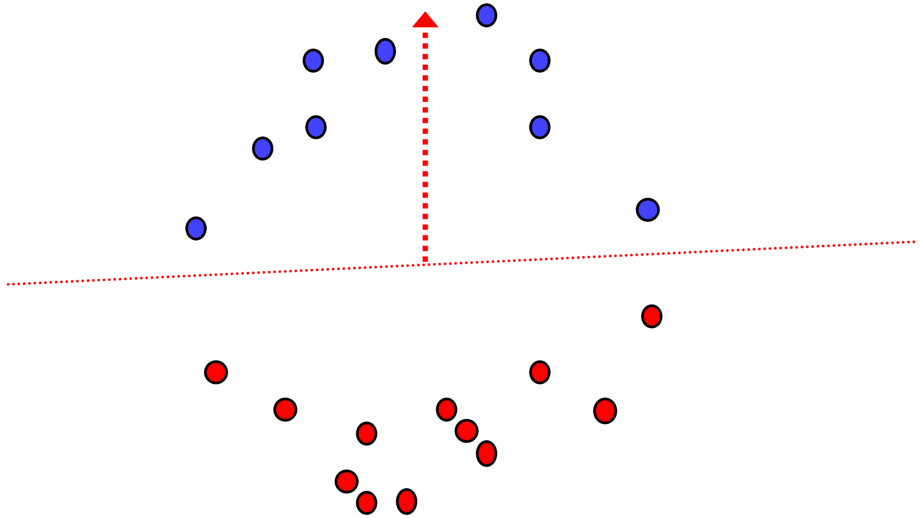
x1	x2	x3	x4	y
1	0	0	1	0
0	1	0	0	1
0	1	1	0	1
1	1	1	0	0

Monomial on a boolean feature vector:

$$M(x_1,..,x_4) = \bar{x}_4 \land x_2 \land \overline{x_1}$$

(homework exercise…)

Linear classifiers



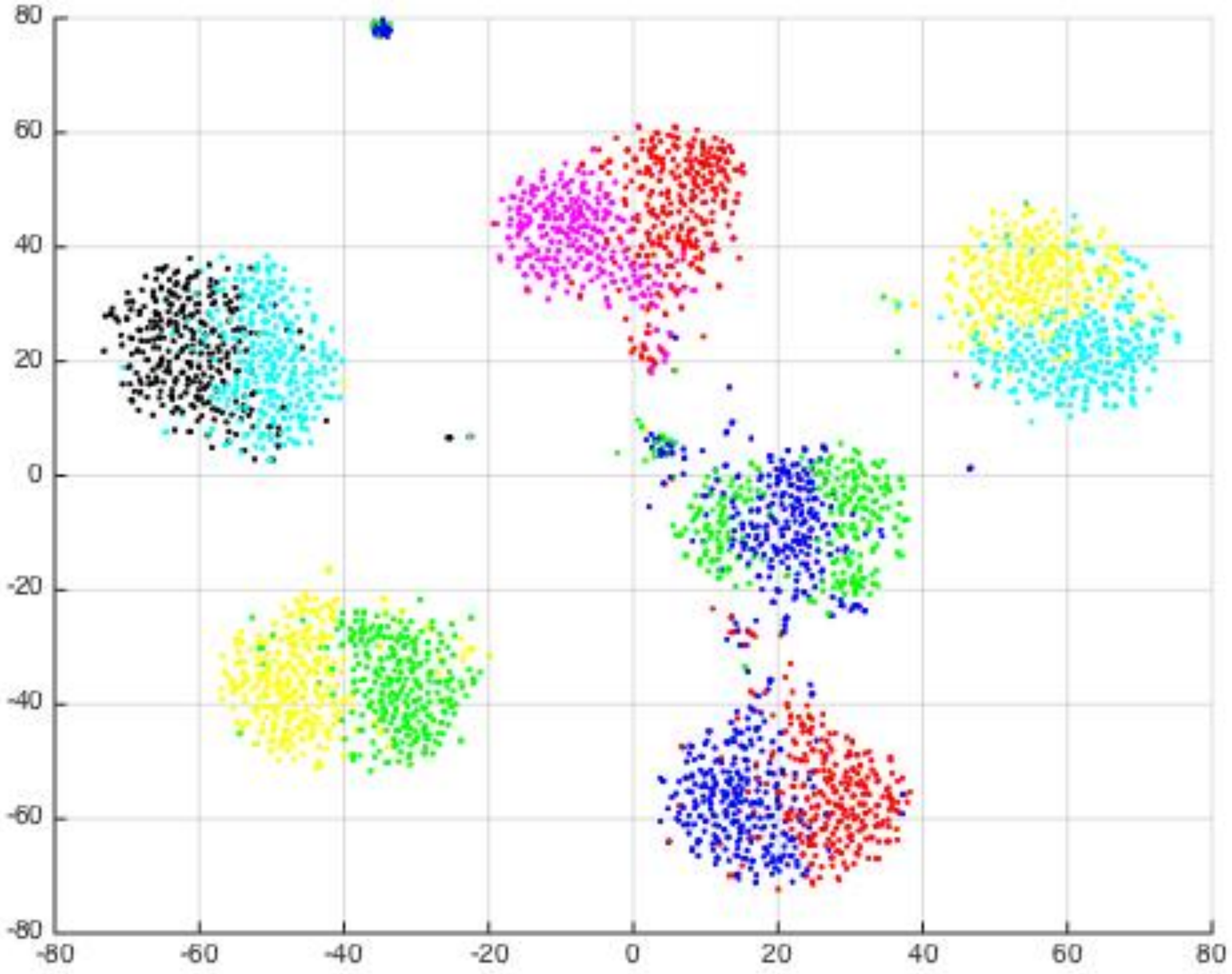
Domain = vectors over Euclidean space R^d

Hypothesis class: all hyperplanes that classify according to:

$$h(x) = sign(w^\top x - b)$$

(we usually ignore b – the bias, it is 0 almost w.l.o.g.)

Empirically: the world is many times linearly-separable



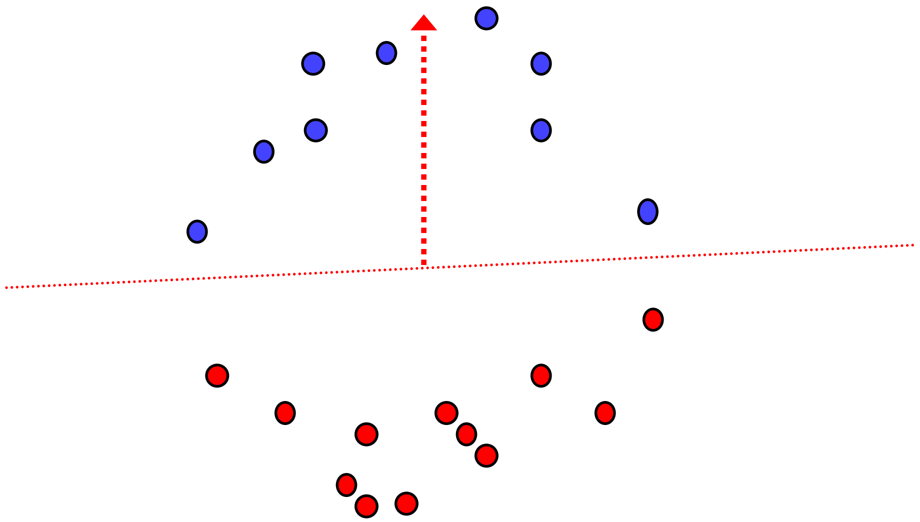
The statistics of linear separators

Hyperplanes in d dimensions with norm at most 1, and accuracy of ϵ:

$$|H| \approx \left(\frac{1}{\epsilon}\right)^d$$

VC dimension is d+1

➔ Sample complexity $O(\frac{d+\log\frac{1}{\delta}}{\epsilon})$



Finding the best linear classifier: Linear Classification

The ERM algorithm reduces to:

n vectors in d dimensions: $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{n} \in R^d$

Labels $\mathrm{y}_1, y_2, \ldots, y_n \in \{-1, 1\}$

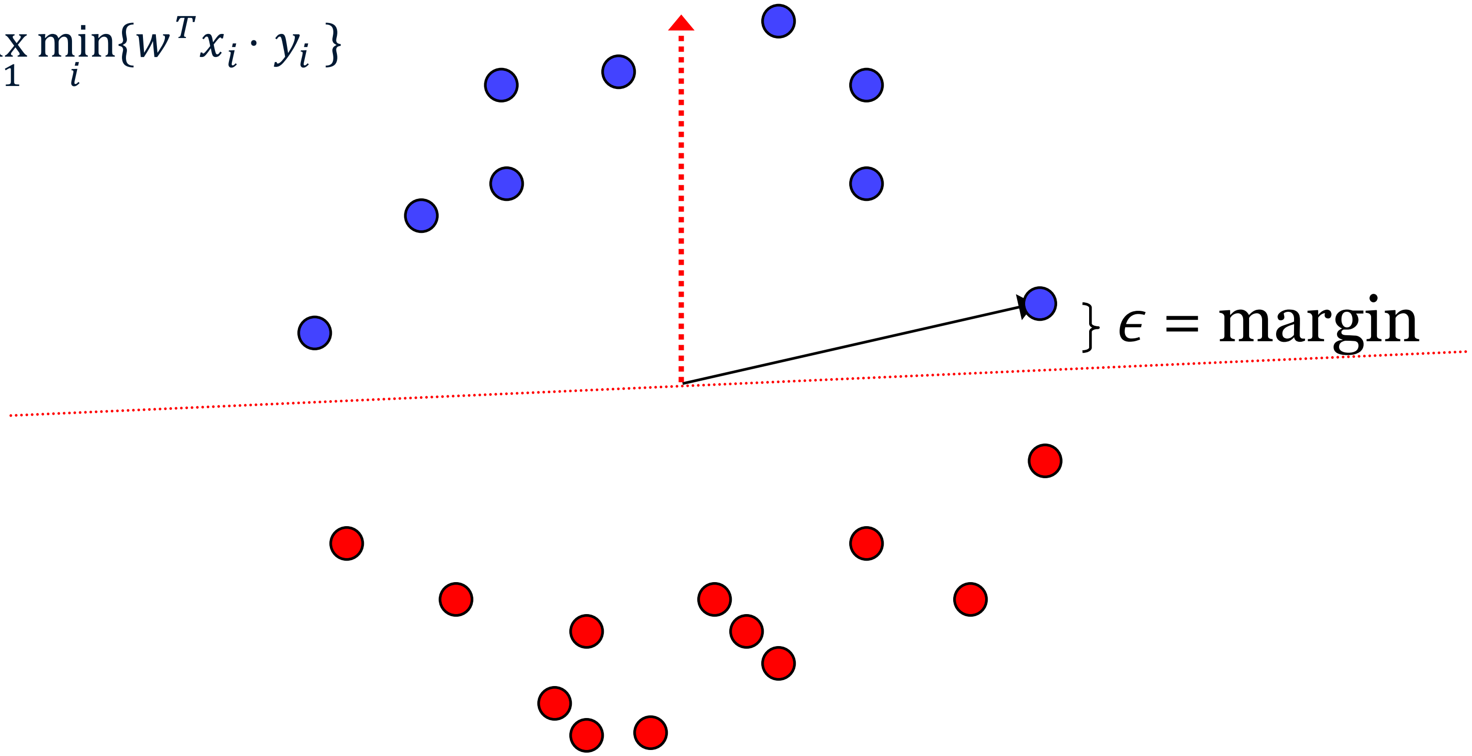
Find vector w such that:

$$\forall i \;.\quad sign(w^T x_i) = y_i$$

Assume: $|\mathrm{w}| \leq 1\,,\ |x_i| \leq 1$

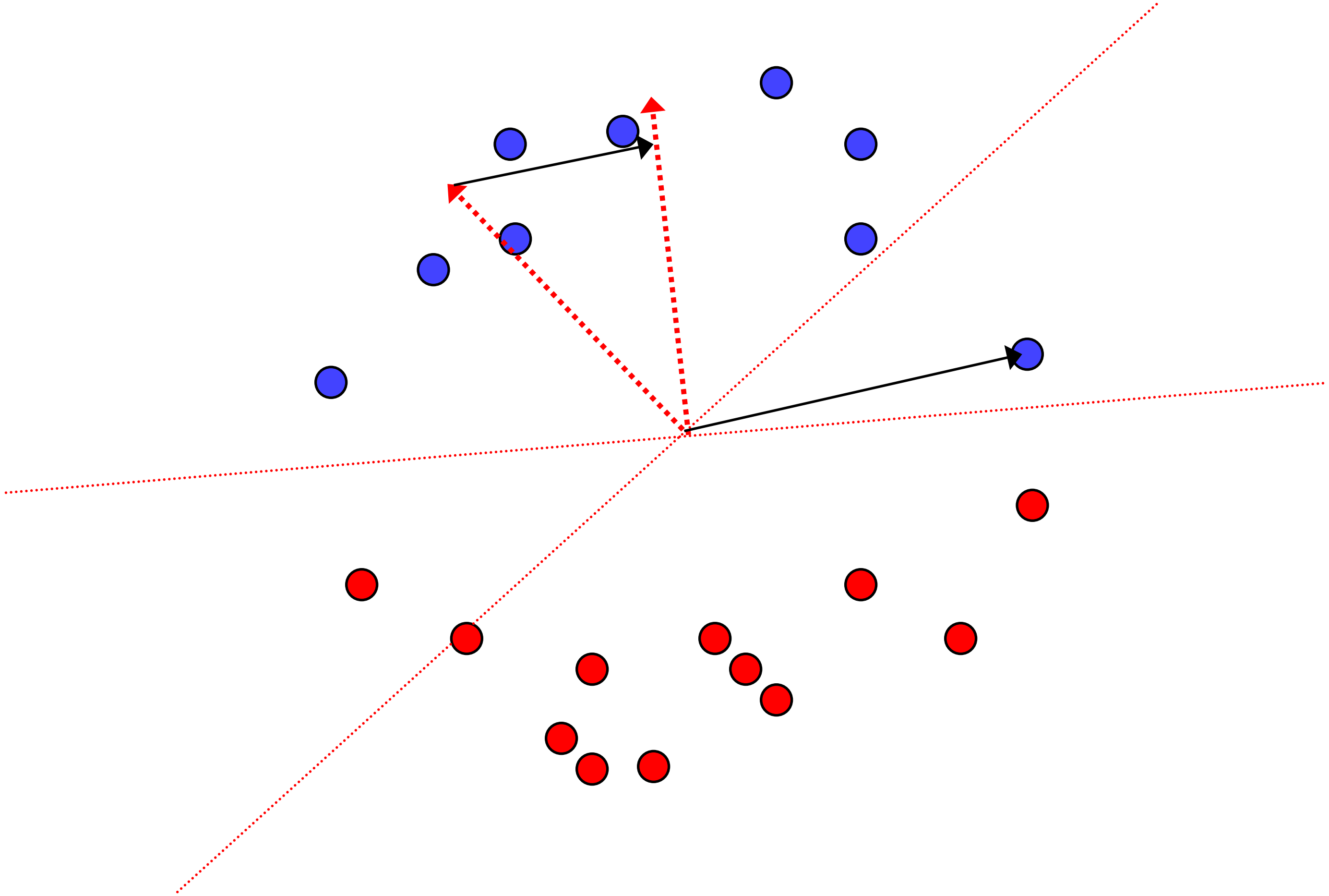
The margin

Margin $\epsilon = \max_{|w|\leq 1} \min_i \{w^T x_i \cdot y_i\}$



The Perceptron Algorithm

[Rosenblatt 1957, Novikoff 1962, Minsky&Papert 1969]



The Perceptron Algorithm

Iteratively:

1. Find vector x_i for which sign $(w^T x_i) \neq y_i$
2. Add x_i to w:

$$w_{t+1} \leftarrow w_t + y_i x_i$$

The Perceptron Algorithm

Thm [Novikoff 1962]: for data with margin ϵ, perceptron returns separating hyperplane in $\frac{1}{\epsilon^2}$ iterations

Perceptron:

$w_{t+1} \leftarrow w_t + y_i x_i$

Thm [Novikoff 1962]: converges in $1/\epsilon^2$ iterations

Proof:

Let w^* be the optimal hyperplane, s.t. $\forall i, y_i x_i^T w^* \geq \epsilon$

1. Observation 1: $w_{t+1}^T w^* = (w_t + y_t x_t) w^* \geq w_t^T w^* + \epsilon$
2. Observation 2:

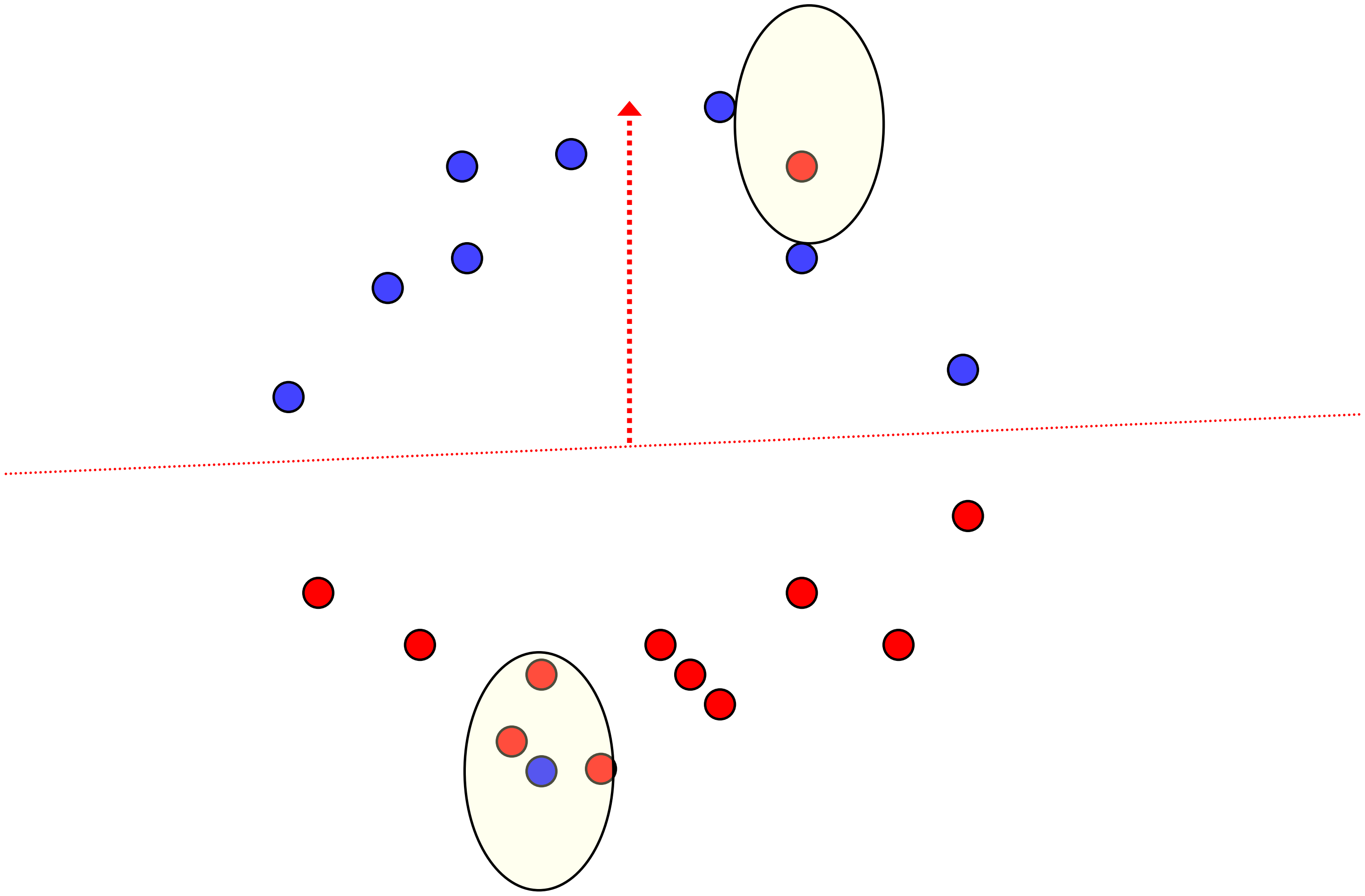
$$|w_{t+1}^T|^2 = |w_t^T|^2 + y_t x_t^T w_t + |y_t x_t|^2 \leq |w_t^T|^2 + 1$$

Thus:

$$1 \geq \frac{w_t}{|w_t|} \cdot w^* \geq \frac{t\epsilon}{\sqrt{t}} = \sqrt{t}\epsilon$$

And hence: $t \leq \frac{1}{\epsilon^2}$

Noise?



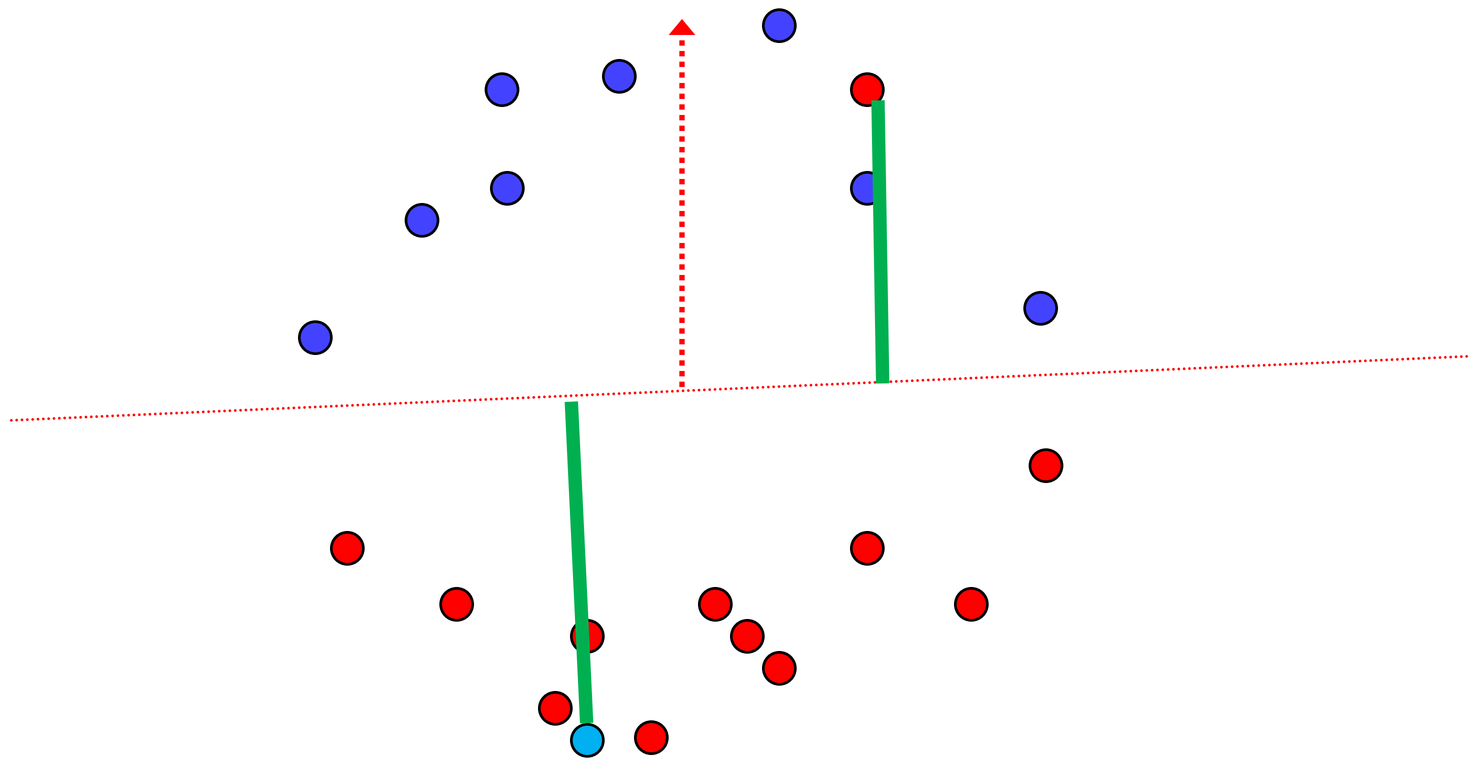
ERM for noisy linear separators?

Given a sample $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, find hyperplane (through the origin w.l.o.g) such that:

$$w = \arg\min_{|w| \leq 1} |\{i \quad \mathrm{s.}t.\ sign\ (w^T x_i) \neq y_i\}|$$

- NP-hard!
- → convex relaxation + optimization!

Noise – minimize sum of weighted violations



Soft-margin SVM (support vector machines)

Given a sample $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, find hyperplane (through the origin w.l.o.g) such that:

$$w = \arg\min_{|w| \leq 1} \{\frac{1}{m} \sum_i \max\{0, 1 - y_i w^\top x_i\}\}$$

- Efficiently solvable by greedy algorithm – gradient descent
- More general methodology: convex optimization

Summary

PAC / Statistical learning theory:

- Precise definition of learning from example
  - Powerful & very general model
- Exact characterization of # of examples to learn (sample complexity)
- Reduction from learning to optimization
- Argued finite hypothesis classes are wonderful (Python)
- Motivated efficient optimization
- Linear classification and the Perceptron + analysis
- SVM → convex optimization (next time!)