Introduction to Machine Learning (67577)
Lecture 2

Shai Shalev-Shwartz

School of CS and Engineering,
The Hebrew University of Jerusalem

PAC learning

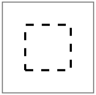
Outline

Recall: The Game Board

- Domain set, $\mathcal{X}$: This is the set of objects that we may wish to label.
- Label set, $\mathcal{Y}$: The set of possible labels.
- A prediction rule, $h : \mathcal{X} \to \mathcal{Y}$: used to label future examples. This function is called a *predictor*, a *hypothesis*, or a *classifier*.

Example

- $\mathcal{X} = \mathbb{R}^2$ representing color and shape of papayas.
- $\mathcal{Y} = \{\pm 1\}$ representing "tasty" or "non-tasty".
- $h(x) = 1$ if x is within the inner rectangle



Batch Learning

- The learner's input:
  - Training data, $S = ((x_1, y_1) \dots (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m$
- The learner's output:
  - A prediction rule, $h : \mathcal{X} \to \mathcal{Y}$

Batch Learning

- The learner's input:
  - Training data, $S = ((x_1, y_1) \ldots (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m$
- The learner's output:
  - A prediction rule, $h : \mathcal{X} \to \mathcal{Y}$

- What should be the goal of the learner?
- Intuitively, h should be correct on future examples

"Correct on future examples"

- Let f be the correct classifier, then we should find h s.t. $h \approx f$

"Correct on future examples"

- Let f be the correct classifier, then we should find h s.t. $h \approx f$
- One way: define the error of h w.r.t. f to be

$$L_{\mathcal{D},f}(h) = \underset{x \sim \mathcal{D}}{\mathbb{P}}[h(x) \neq f(x)]$$

where $\mathcal{D}$ is some (unknown) probability measure over $\mathcal{X}$

"Correct on future examples"

- Let f be the correct classifier, then we should find h s.t. $h \approx f$
- One way: define the error of h w.r.t. f to be

$$L_{\mathcal{D},f}(h) = \mathop{\mathbb{P}}_{x \sim \mathcal{D}}[h(x) \neq f(x)]$$

where $\mathcal{D}$ is some (unknown) probability measure over $\mathcal{X}$

- More formally, $\mathcal{D}$ is a distribution over $\mathcal{X}$, that is, for a given $A \subset \mathcal{X}$, the value of $\mathcal{D}(A)$ is the probability to see some $x \in A$. Then,

$$L_{\mathcal{D},f}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{P}}_{x \sim \mathcal{D}}[h(x) \neq f(x)] \stackrel{\text{def}}{=} \mathcal{D}\left(\{x \in \mathcal{X} : h(x) \neq f(x)\}\right) .$$

"Correct on future examples"

- Let f be the correct classifier, then we should find h s.t. $h \approx f$
- One way: define the error of h w.r.t. f to be

$$L_{\mathcal{D},f}(h) = \mathop{\mathbb{P}}_{x \sim \mathcal{D}}[h(x) \neq f(x)]$$

  where $\mathcal{D}$ is some (unknown) probability measure over $\mathcal{X}$
- More formally, $\mathcal{D}$ is a distribution over $\mathcal{X}$, that is, for a given $A \subset \mathcal{X}$, the value of $\mathcal{D}(A)$ is the probability to see some $x \in A$. Then,

$$L_{\mathcal{D},f}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{P}}_{x \sim \mathcal{D}}[h(x) \neq f(x)] \stackrel{\text{def}}{=} \mathcal{D}\left(\{x \in \mathcal{X} : h(x) \neq f(x)\}\right) .$$

- Can we find h s.t. $L_{\mathcal{D},f}(h)$ is small ?

Data-generation Model

- We must assume some relation between the training data and $\mathcal{D}, f$
- Simple data generation model:
  - Independently Identically Distributed (i.i.d.): Each x_i is sampled independently according to $\mathcal{D}$
  - Realizability: For every $i \in [m]$, $y_i = f(x_i)$

Can only be **Approximately** correct

- Claim: We can't hope to find h s.t. $L_{(\mathcal{D},f)}(h) = 0$

Can only be **Approximately** correct

- Claim: We can't hope to find h s.t. $L_{(\mathcal{D},f)}(h) = 0$
- Proof: for every $\epsilon \in (0,1)$ take $\mathcal{X} = \{x_1, x_2\}$ and $\mathcal{D}(\{x_1\}) = 1 - \epsilon$, $\mathcal{D}(\{x_2\}) = \epsilon$

Can only be **Approximately** correct

- Claim: We can't hope to find h s.t. $L_{(\mathcal{D},f)}(h) = 0$
- Proof: for every $\epsilon \in (0,1)$ take $\mathcal{X} = \{x_1, x_2\}$ and $\mathcal{D}(\{x_1\}) = 1 - \epsilon$, $\mathcal{D}(\{x_2\}) = \epsilon$
- The probability not to see x_2 at all among m i.i.d. examples is $(1-\epsilon)^m \approx e^{-\epsilon m}$

Can only be **Approximately** correct

- Claim: We can't hope to find h s.t. $L_{(\mathcal{D},f)}(h) = 0$
- Proof: for every $\epsilon \in (0,1)$ take $\mathcal{X} = \{x_1, x_2\}$ and $\mathcal{D}(\{x_1\}) = 1 - \epsilon$, $\mathcal{D}(\{x_2\}) = \epsilon$
- The probability not to see x_2 at all among m i.i.d. examples is $(1-\epsilon)^m \approx e^{-\epsilon m}$
- So, if $\epsilon \ll 1/m$ we're likely not to see x_2 at all, but then we can't know its label

Can only be **Approximately** correct

- Claim: We can't hope to find h s.t. $L_{(\mathcal{D},f)}(h) = 0$
- Proof: for every $\epsilon \in (0,1)$ take $\mathcal{X} = \{x_1, x_2\}$ and $\mathcal{D}(\{x_1\}) = 1 - \epsilon$, $\mathcal{D}(\{x_2\}) = \epsilon$
- The probability not to see x_2 at all among m i.i.d. examples is $(1-\epsilon)^m \approx e^{-\epsilon m}$
- So, if $\epsilon \ll 1/m$ we're likely not to see x_2 at all, but then we can't know its label
- Relaxation: We'd be happy with $L_{(\mathcal{D},f)}(h) \leq \epsilon$, where ϵ is user-specified

Can only be **Probably** correct

- Recall that the input to the learner is randomly generated

Can only be **Probably** correct

- Recall that the input to the learner is randomly generated
- There's always a (very small) chance to see the same example again and again

Can only be **Probably** correct

- Recall that the input to the learner is randomly generated
- There's always a (very small) chance to see the same example again and again
- Claim: No algorithm can guarantee $L_{(\mathcal{D},f)}(h) \le \epsilon$ for sure

Can only be **Probably** correct

- Recall that the input to the learner is randomly generated
- There's always a (very small) chance to see the same example again and again
- Claim: No algorithm can guarantee $L_{(\mathcal{D},f)}(h) \leq \epsilon$ for sure
- Relaxation: We'd allow the algorithm to fail with probability δ, where $\delta \in (0,1)$ is user-specified
  Here, the probability is over the random choice of examples

Probably Approximately Correct (PAC) learning

- The learner doesn't know $\mathcal{D}$ and f.

Probably Approximately Correct (PAC) learning

- The learner doesn't know $\mathcal{D}$ and f.
- The learner receives accuracy parameter ϵ and confidence parameter δ

Probably **A**pproximately **C**orrect (PAC) learning

- The learner doesn't know $\mathcal{D}$ and f.
- The learner receives accuracy parameter ϵ and confidence parameter δ
- The learner can ask for training data, S, containing $m(\epsilon, \delta)$ examples (that is, the number of examples can depend on the value of ϵ and δ, but it can't depend on $\mathcal{D}$ or f)

Probably Approximately Correct (PAC) learning

- The learner doesn't know $\mathcal{D}$ and f.
- The learner receives accuracy parameter ϵ and confidence parameter δ
- The learner can ask for training data, S, containing $m(\epsilon, \delta)$ examples (that is, the number of examples can depend on the value of ϵ and δ, but it can't depend on $\mathcal{D}$ or f)
- Learner should output a hypothesis h s.t. with probability of at least $1 - \delta$ it holds that $L_{\mathcal{D},f}(h) \leq \epsilon$

Probably Approximately Correct (PAC) learning

- The learner doesn't know $\mathcal{D}$ and f.
- The learner receives accuracy parameter ϵ and confidence parameter δ
- The learner can ask for training data, S, containing $m(\epsilon, \delta)$ examples (that is, the number of examples can depend on the value of ϵ and δ, but it can't depend on $\mathcal{D}$ or f)
- Learner should output a hypothesis h s.t. with probability of at least $1 - \delta$ it holds that $L_{\mathcal{D},f}(h) \leq \epsilon$
- That is, the learner should be **P**robably (with probability at least $1 - \delta$) **A**pproximately (up to accuracy ϵ) **C**orrect

No Free Lunch

- Suppose that $|\mathcal{X}| = \infty$
- For any finite $C \subset \mathcal{X}$ take $\mathcal{D}$ to be uniform distribution over C
- If number of training examples is $m \leq |C|/2$ the learner has no knowledge on at least half the elements in C
- Formalizing the above, it can be shown that:

No Free Lunch

- Suppose that $|\mathcal{X}| = \infty$
- For any finite $C \subset \mathcal{X}$ take $\mathcal{D}$ to be uniform distribution over C
- If number of training examples is $m \leq |C|/2$ the learner has no knowledge on at least half the elements in C
- Formalizing the above, it can be shown that:

Theorem (No Free Lunch)

Fix $\delta \in (0,1), \epsilon < 1/2$. For every learner A and training set size m, there exists $\mathcal{D}, f$ such that with probability of at least δ over the generation of a training data, S, of m examples, it holds that $L_{\mathcal{D},f}(A(S)) \geq \epsilon$.

No Free Lunch

- Suppose that $|\mathcal{X}| = \infty$
- For any finite $C \subset \mathcal{X}$ take $\mathcal{D}$ to be uniform distribution over C
- If number of training examples is $m \leq |C|/2$ the learner has no knowledge on at least half the elements in C
- Formalizing the above, it can be shown that:

Theorem (No Free Lunch)

Fix $\delta \in (0,1), \epsilon < 1/2$. For every learner A and training set size m, there exists $\mathcal{D}, f$ such that with probability of at least δ over the generation of a training data, S, of m examples, it holds that $L_{D,f}(A(S)) \geq \epsilon$.

Remark: $L_{D,f}(\text{random guess}) = 1/2$, so the theorem states that you can't be better than a random guess

Prior Knowledge

- Give more knowledge to the learner: the target f comes from some hypothesis class, $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$
- The learner knows $\mathcal{H}$
- Is it possible to PAC learn now ?
- Of course, the answer depends on $\mathcal{H}$ since the No Free Lunch theorem tells us that for $\mathcal{H} = \mathcal{Y}^{\mathcal{X}}$ the problem is not solvable ...

Outline

Learning Finite Classes

- Assume that $\mathcal{H}$ is a finite hypothesis class

Learning Finite Classes

- Assume that $\mathcal{H}$ is a finite hypothesis class
  - E.g.: $\mathcal{H}$ is all the functions from $\mathcal{X}$ to $\mathcal{Y}$ that can be implemented using a Python program of length at most b

Learning Finite Classes

- Assume that $\mathcal{H}$ is a finite hypothesis class
  - E.g.: $\mathcal{H}$ is all the functions from $\mathcal{X}$ to $\mathcal{Y}$ that can be implemented using a Python program of length at most b
- Use the Consistent learning rule:

Learning Finite Classes

- Assume that $\mathcal{H}$ is a finite hypothesis class
  - E.g.: $\mathcal{H}$ is all the functions from $\mathcal{X}$ to $\mathcal{Y}$ that can be implemented using a Python program of length at most b
- Use the Consistent learning rule:
  - Input: $\mathcal{H}$ and $S = (x_1, y_1), \ldots, (x_m, y_m)$

Learning Finite Classes

- Assume that $\mathcal{H}$ is a finite hypothesis class
  - E.g.: $\mathcal{H}$ is all the functions from $\mathcal{X}$ to $\mathcal{Y}$ that can be implemented using a Python program of length at most b
- Use the Consistent learning rule:
  - Input: $\mathcal{H}$ and $S = (x_1, y_1), \ldots, (x_m, y_m)$
  - Output: any $h \in \mathcal{H}$ s.t. $\forall i,\ y_i = h(x_i)$

Learning Finite Classes

- Assume that $\mathcal{H}$ is a finite hypothesis class
  - E.g.: $\mathcal{H}$ is all the functions from $\mathcal{X}$ to $\mathcal{Y}$ that can be implemented using a Python program of length at most b
- Use the Consistent learning rule:
  - Input: $\mathcal{H}$ and $S = (x_1, y_1), \ldots, (x_m, y_m)$
  - Output: any $h \in \mathcal{H}$ s.t. $\forall i,\ y_i = h(x_i)$
- This is also called Empirical Risk Minimization (ERM)

Learning Finite Classes

- Assume that $\mathcal{H}$ is a finite hypothesis class
  - E.g.: $\mathcal{H}$ is all the functions from $\mathcal{X}$ to $\mathcal{Y}$ that can be implemented using a Python program of length at most b
- Use the Consistent learning rule:
  - Input: $\mathcal{H}$ and $S = (x_1, y_1), \ldots, (x_m, y_m)$
  - Output: any $h \in \mathcal{H}$ s.t. $\forall i,\ y_i = h(x_i)$
- This is also called Empirical Risk Minimization (ERM)

Learning Finite Classes

- Assume that $\mathcal{H}$ is a finite hypothesis class
  - E.g.: $\mathcal{H}$ is all the functions from $\mathcal{X}$ to $\mathcal{Y}$ that can be implemented using a Python program of length at most b
- Use the Consistent learning rule:
  - Input: $\mathcal{H}$ and $S = (x_1, y_1), \ldots, (x_m, y_m)$
  - Output: any $h \in \mathcal{H}$ s.t. $\forall i,\ y_i = h(x_i)$
- This is also called Empirical Risk Minimization (ERM)

$\mathrm{ERM}_{\mathcal{H}}(S)$

- Input: training set $S = (x_1, y_1), \ldots, (x_m, y_m)$
- Define the empirical risk: $L_S(h) = \frac{1}{m}|\{i : h(x_i) \neq y_i\}|$
- Output: any $h \in \mathcal{H}$ that minimizes $L_S(h)$

Learning Finite Classes

Theorem

Fix ϵ, δ. *If* $m \geq \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}$ *then for every* $\mathcal{D}, f$, *with probability of at least* $1 - \delta$ *(over the choice of* S *of size* m*),* $L_{\mathcal{D},f}(\mathrm{ERM}_{\mathcal{H}}(S)) \leq \epsilon$.

Proof

- Let $S|_x = (x_1, \ldots, x_m)$ be the instances of the training set

Proof

- Let $S|_x = (x_1, \ldots, x_m)$ be the instances of the training set
- We would like to prove:

$$\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(\mathrm{ERM}_{\mathcal{H}}(S)) > \epsilon\}) \leq \delta$$

Proof

- Let $S|_x = (x_1, \ldots, x_m)$ be the instances of the training set
- We would like to prove:

$$\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(\mathrm{ERM}_{\mathcal{H}}(S)) > \epsilon\}) \leq \delta$$

- Let $\mathcal{H}_B$ be the set of "bad" hypotheses,

$$\mathcal{H}_B = \{h \in \mathcal{H} : L_{(\mathcal{D},f)}(h) > \epsilon\}$$

Proof

- Let $S|_x = (x_1, \ldots, x_m)$ be the instances of the training set
- We would like to prove:

$$\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(\mathrm{ERM}_{\mathcal{H}}(S)) > \epsilon\}) \leq \delta$$

- Let $\mathcal{H}_B$ be the set of "bad" hypotheses,

$$\mathcal{H}_B = \{h \in \mathcal{H} : L_{(\mathcal{D},f)}(h) > \epsilon\}$$

- Let M be the set of "misleading" samples,

$$M = \{S|_x : \exists h \in \mathcal{H}_B, L_S(h) = 0\}$$

Proof

- Let $S|_x = (x_1, \ldots, x_m)$ be the instances of the training set
- We would like to prove:

$$\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(\mathrm{ERM}_{\mathcal{H}}(S)) > \epsilon\}) \le \delta$$

- Let $\mathcal{H}_B$ be the set of "bad" hypotheses,

$$\mathcal{H}_B = \{h \in \mathcal{H} : L_{(\mathcal{D},f)}(h) > \epsilon\}$$

- Let M be the set of "misleading" samples,

$$M = \{S|_x : \exists h \in \mathcal{H}_B, L_S(h) = 0\}$$

- Observe:

$$\{S|_x : L_{(\mathcal{D},f)}(\mathrm{ERM}_{\mathcal{H}}(S)) > \epsilon\} \subseteq M = \bigcup_{h \in \mathcal{H}_B} \{S|_x : L_S(h) = 0\}$$

Proof (Cont.)

Lemma (Union bound)

For any two sets A, B *and a distribution* $\mathcal{D}$ *we have*

$$\mathcal{D}(A \cup B) \leq \mathcal{D}(A) + \mathcal{D}(B) \ .$$

Proof (Cont.)

Lemma (Union bound)

For any two sets A, B and a distribution $\mathcal{D}$ we have

$$\mathcal{D}(A \cup B) \leq \mathcal{D}(A) + \mathcal{D}(B) \ .$$

- We have shown:
  $\{S|_x : L_{(\mathcal{D},f)}(\mathrm{ERM}_{\mathcal{H}}(S)) > \epsilon\} \subseteq \bigcup_{h \in \mathcal{H}_B} \{S|_x : L_S(h) = 0\}$

Proof (Cont.)

Lemma (Union bound)

For any two sets A, B and a distribution $\mathcal{D}$ we have

$$\mathcal{D}(A \cup B) \leq \mathcal{D}(A) + \mathcal{D}(B) \ .$$

- We have shown: $\{S|_x : L_{(\mathcal{D},f)}(\mathrm{ERM}_{\mathcal{H}}(S)) > \epsilon\} \subseteq \bigcup_{h \in \mathcal{H}_B} \{S|_x : L_S(h) = 0\}$
- Therefore, using the union bound

$$\begin{aligned} \mathcal{D}^m(\{S|_x : & L_{(\mathcal{D},f)}(\mathrm{ERM}_{\mathcal{H}}(S)) > \epsilon\}) \\ & \leq \sum_{h \in \mathcal{H}_B} \mathcal{D}^m(\{S|_x : L_S(h) = 0\}) \\ & \leq |\mathcal{H}_B| \max_{h \in \mathcal{H}_B} \mathcal{D}^m(\{S|_x : L_S(h) = 0\}) \end{aligned}$$

Proof (Cont.)

- Observe:

$$\mathcal{D}^m(\{S|_x : L_S(h) = 0\}) \;=\; (1 - L_{\mathcal{D},f}(h))^m$$

Proof (Cont.)

- Observe:

$$\mathcal{D}^m(\{S|_x : L_S(h) = 0\}) \;=\; (1 - L_{\mathcal{D},f}(h))^m$$

- If $h \in \mathcal{H}_B$ then $L_{\mathcal{D},f}(h) > \epsilon$ and therefore

$$\mathcal{D}^m(\{S|_x : L_S(h) = 0\}) \;<\; (1 - \epsilon)^m$$

Proof (Cont.)

- Observe:

$$\mathcal{D}^m(\{S|_x : L_S(h) = 0\}) \;=\; (1 - L_{\mathcal{D},f}(h))^m$$

- If $h \in \mathcal{H}_B$ then $L_{\mathcal{D},f}(h) > \epsilon$ and therefore

$$\mathcal{D}^m(\{S|_x : L_S(h) = 0\}) \;<\; (1 - \epsilon)^m$$

- We have shown:

$$\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(\mathrm{ERM}_{\mathcal{H}}(S)) > \epsilon\}) \;<\; |\mathcal{H}_B|\,(1 - \epsilon)^m$$

Proof (Cont.)

- Observe:

$$\mathcal{D}^m(\{S|_x : L_S(h) = 0\}) \;=\; (1 - L_{\mathcal{D},f}(h))^m$$

- If $h \in \mathcal{H}_B$ then $L_{\mathcal{D},f}(h) > \epsilon$ and therefore

$$\mathcal{D}^m(\{S|_x : L_S(h) = 0\}) \;<\; (1 - \epsilon)^m$$

- We have shown:

$$\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(\mathrm{ERM}_{\mathcal{H}}(S)) > \epsilon\}) \;<\; |\mathcal{H}_B|\,(1 - \epsilon)^m$$

- Finally, using $1 - \epsilon \le e^{-\epsilon}$ and $|\mathcal{H}_B| \le |\mathcal{H}|$ we conclude:

$$\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(\mathrm{ERM}_{\mathcal{H}}(S)) > \epsilon\}) \;<\; |\mathcal{H}|\,e^{-\epsilon\,m}$$

Proof (Cont.)

- Observe:

$$\mathcal{D}^m(\{S|_x : L_S(h) = 0\}) \;=\; (1 - L_{\mathcal{D},f}(h))^m$$

- If $h \in \mathcal{H}_B$ then $L_{\mathcal{D},f}(h) > \epsilon$ and therefore

$$\mathcal{D}^m(\{S|_x : L_S(h) = 0\}) \;<\; (1 - \epsilon)^m$$

- We have shown:

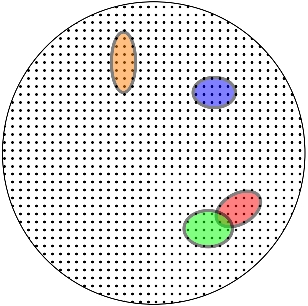
$$\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(\mathrm{ERM}_{\mathcal{H}}(S)) > \epsilon\}) \;<\; |\mathcal{H}_B|\,(1 - \epsilon)^m$$

- Finally, using $1 - \epsilon \leq e^{-\epsilon}$ and $|\mathcal{H}_B| \leq |\mathcal{H}|$ we conclude:

$$\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(\mathrm{ERM}_{\mathcal{H}}(S)) > \epsilon\}) \;<\; |\mathcal{H}|\,e^{-\epsilon\,m}$$

- The right-hand side would be $\leq \delta$ if $m \geq \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}$. $\square$

Illustrating the use of the union bound



- Each point is a possible sample $S|_x$. Each colored oval represents misleading samples for some $h \in \mathcal{H}_B$. The probability mass of each such oval is at most $(1-\epsilon)^m$. But, the algorithm might err if it samples $S|_x$ from any of these ovals.

Outline

Definition (PAC learnability)

A hypothesis class $\mathcal{H}$ is PAC learnable if there exists a function $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ and a learning algorithm with the following property:

- for every $\epsilon, \delta \in (0,1)$
- for every distribution $\mathcal{D}$ over $\mathcal{X}$, and for every labeling function $f : \mathcal{X} \to \{0,1\}$

when running the learning algorithm on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathcal{D}$ and labeled by f, the algorithm returns a hypothesis h such that, with probability of at least $1 - \delta$ (over the choice of the examples), $L_{(\mathcal{D},f)}(h) \leq \epsilon$.

PAC learning

Definition (PAC learnability)

A hypothesis class $\mathcal{H}$ is PAC learnable if there exists a function $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ and a learning algorithm with the following property:

- for every $\epsilon, \delta \in (0,1)$
- for every distribution $\mathcal{D}$ over $\mathcal{X}$, and for every labeling function $f : \mathcal{X} \to \{0,1\}$

when running the learning algorithm on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathcal{D}$ and labeled by f, the algorithm returns a hypothesis h such that, with probability of at least $1 - \delta$ (over the choice of the examples), $L_{(\mathcal{D},f)}(h) \leq \epsilon$.

$m_{\mathcal{H}}$ is called the sample complexity of learning $\mathcal{H}$

PAC learning

Leslie Valiant, Turing award 2010

> *For transformative contributions to the theory of computation, including the theory of probably approximately correct (PAC) learning, the complexity of enumeration and of algebraic computation, and the theory of parallel and distributed computing.*

What is learnable and how to learn?

- We have shown:

Corollary

Let $\mathcal{H}$ be a finite hypothesis class.

- *$\mathcal{H}$ is PAC learnable with sample complexity* $m_{\mathcal{H}}(\epsilon, \delta) \leq \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}$
- *This sample complexity is obtained by using the* $\mathrm{ERM}_{\mathcal{H}}$ *learning rule*

What is learnable and how to learn?

- We have shown:

Corollary

Let $\mathcal{H}$ be a finite hypothesis class.

- *$\mathcal{H}$ is PAC learnable with sample complexity* $m_{\mathcal{H}}(\epsilon, \delta) \le \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}$
- *This sample complexity is obtained by using the* $\mathrm{ERM}_{\mathcal{H}}$ *learning rule*

- What about infinite hypothesis classes?
- What is the sample complexity of a given class?
- Is there a generic learning algorithm that achieves the optimal sample complexity ?

What is learnable and how to learn?

The fundamental theorem of statistical learning:

- The sample complexity is characterized by the VC dimension
- The ERM learning rule is a generic (near) optimal learner

Chervonenkis

Vapnik

Outline

The VC dimension — Motivation

if someone can explain every phenomena, her explanations are worthless.

Example: http://www.youtube.com/watch?v=p_MzP2MZaOo

Pay attention to the retrospect explanations at 5:00

The VC dimension — Motivation

- Suppose we got a training set $S = (x_1, y_1), \ldots, (x_m, y_m)$

The VC dimension — Motivation

- Suppose we got a training set $S = (x_1, y_1), \ldots, (x_m, y_m)$
- We try to explain the labels using a hypothesis from $\mathcal{H}$

The VC dimension — Motivation

- Suppose we got a training set $S = (x_1, y_1), \ldots, (x_m, y_m)$
- We try to explain the labels using a hypothesis from $\mathcal{H}$
- Then, ooops, the labels we received are incorrect and we get the same instances with different labels, $S' = (x_1, y'_1), \ldots, (x_m, y'_m)$

The VC dimension — Motivation

- Suppose we got a training set $S = (x_1, y_1), \ldots, (x_m, y_m)$
- We try to explain the labels using a hypothesis from $\mathcal{H}$
- Then, ooops, the labels we received are incorrect and we get the same instances with different labels, $S' = (x_1, y'_1), \ldots, (x_m, y'_m)$
- We again try to explain the labels using a hypothesis from $\mathcal{H}$

The VC dimension — Motivation

- Suppose we got a training set $S = (x_1, y_1), \ldots, (x_m, y_m)$
- We try to explain the labels using a hypothesis from $\mathcal{H}$
- Then, ooops, the labels we received are incorrect and we get the same instances with different labels, $S' = (x_1, y'_1), \ldots, (x_m, y'_m)$
- We again try to explain the labels using a hypothesis from $\mathcal{H}$
- If this works for us, no matter what the labels are, then something is fishy ...

The VC dimension — Motivation

- Suppose we got a training set $S = (x_1, y_1), \ldots, (x_m, y_m)$
- We try to explain the labels using a hypothesis from $\mathcal{H}$
- Then, ooops, the labels we received are incorrect and we get the same instances with different labels, $S' = (x_1, y_1'), \ldots, (x_m, y_m')$
- We again try to explain the labels using a hypothesis from $\mathcal{H}$
- If this works for us, no matter what the labels are, then something is fishy ...
- Formally, if $\mathcal{H}$ allows all functions over some set C of size m, then based on the No Free Lunch, we can't learn from, say, $m/2$ examples

The VC dimension — Formal Definition

- Let $C = \{x_1, \ldots, x_{|C|}\} \subset \mathcal{X}$

The VC dimension — Formal Definition

- Let $C = \{x_1, \ldots, x_{|C|}\} \subset \mathcal{X}$
- Let $\mathcal{H}_C$ be the restriction of $\mathcal{H}$ to C, namely, $\mathcal{H}_C = \{h_C : h \in \mathcal{H}\}$ where $h_C : C \to \{0, 1\}$ is s.t. $h_C(x_i) = h(x_i)$ for every $x_i \in C$

The VC dimension — Formal Definition

- Let $C = \{x_1, \ldots, x_{|C|}\} \subset \mathcal{X}$
- Let $\mathcal{H}_C$ be the restriction of $\mathcal{H}$ to C, namely, $\mathcal{H}_C = \{h_C : h \in \mathcal{H}\}$ where $h_C : C \to \{0, 1\}$ is s.t. $h_C(x_i) = h(x_i)$ for every $x_i \in C$
- Observe: we can represent each h_C as the vector $(h(x_1), \ldots, h(x_{|C|})) \in \{\pm 1\}^{|C|}$

The VC dimension — Formal Definition

- Let $C = \{x_1, \ldots, x_{|C|}\} \subset \mathcal{X}$
- Let $\mathcal{H}_C$ be the restriction of $\mathcal{H}$ to C, namely, $\mathcal{H}_C = \{h_C : h \in \mathcal{H}\}$ where $h_C : C \to \{0, 1\}$ is s.t. $h_C(x_i) = h(x_i)$ for every $x_i \in C$
- Observe: we can represent each h_C as the vector $(h(x_1), \ldots, h(x_{|C|})) \in \{\pm 1\}^{|C|}$
- Therefore: $|\mathcal{H}_C| \leq 2^{|C|}$

The VC dimension — Formal Definition

- Let $C = \{x_1, \ldots, x_{|C|}\} \subset \mathcal{X}$
- Let $\mathcal{H}_C$ be the restriction of $\mathcal{H}$ to C, namely, $\mathcal{H}_C = \{h_C : h \in \mathcal{H}\}$ where $h_C : C \to \{0, 1\}$ is s.t. $h_C(x_i) = h(x_i)$ for every $x_i \in C$
- Observe: we can represent each h_C as the vector $(h(x_1), \ldots, h(x_{|C|})) \in \{\pm 1\}^{|C|}$
- Therefore: $|\mathcal{H}_C| \le 2^{|C|}$
- We say that $\mathcal{H}$ shatters C if $|\mathcal{H}_C| = 2^{|C|}$

The VC dimension — Formal Definition

- Let $C = \{x_1, \ldots, x_{|C|}\} \subset \mathcal{X}$
- Let $\mathcal{H}_C$ be the restriction of $\mathcal{H}$ to C, namely, $\mathcal{H}_C = \{h_C : h \in \mathcal{H}\}$ where $h_C : C \to \{0, 1\}$ is s.t. $h_C(x_i) = h(x_i)$ for every $x_i \in C$
- Observe: we can represent each h_C as the vector $(h(x_1), \ldots, h(x_{|C|})) \in \{\pm 1\}^{|C|}$
- Therefore: $|\mathcal{H}_C| \leq 2^{|C|}$
- We say that $\mathcal{H}$ shatters C if $|\mathcal{H}_C| = 2^{|C|}$
- $\mathrm{VCdim}(\mathcal{H}) = \sup\{|C| \ : \ \mathcal{H} \text{ shatters } C\}$

The VC dimension — Formal Definition

- Let $C = \{x_1, \ldots, x_{|C|}\} \subset \mathcal{X}$
- Let $\mathcal{H}_C$ be the restriction of $\mathcal{H}$ to C, namely, $\mathcal{H}_C = \{h_C : h \in \mathcal{H}\}$ where $h_C : C \to \{0, 1\}$ is s.t. $h_C(x_i) = h(x_i)$ for every $x_i \in C$
- Observe: we can represent each h_C as the vector $(h(x_1), \ldots, h(x_{|C|})) \in \{\pm 1\}^{|C|}$
- Therefore: $|\mathcal{H}_C| \le 2^{|C|}$
- We say that $\mathcal{H}$ shatters C if $|\mathcal{H}_C| = 2^{|C|}$
- $\mathrm{VCdim}(\mathcal{H}) = \sup\{|C| \ : \ \mathcal{H} \text{ shatters } C\}$
- That is, the VC dimension is the maximal size of a set C such that $\mathcal{H}$ gives no prior knowledge w.r.t. C

VC dimension — Examples

To show that $\mathrm{VCdim}(\mathcal{H}) = d$ we need to show that:

1. There exists a set C of size d which is shattered by $\mathcal{H}$.

VC dimension — Examples

To show that $\mathrm{VCdim}(\mathcal{H}) = d$ we need to show that:

1. There exists a set C of size d which is shattered by $\mathcal{H}$.
2. Every set C of size $d+1$ is not shattered by $\mathcal{H}$.

VC dimension — Examples

Threshold functions: $\mathcal{X} = \mathbb{R}$, $\mathcal{H} = \{x \mapsto \text{sign}(x - \theta) : \theta \in \mathbb{R}\}$

- Show that $\{0\}$ is shattered

VC dimension — Examples

Threshold functions: $\mathcal{X} = \mathbb{R}$, $\mathcal{H} = \{x \mapsto \text{sign}(x - \theta) : \theta \in \mathbb{R}\}$

- Show that $\{0\}$ is shattered
- Show that any two points cannot be shattered

VC dimension — Examples

Intervals: $\mathcal{X} = \mathbb{R}$, $\mathcal{H} = \{h_{a,b} : a < b \in \mathbb{R}\}$, where $h_{a,b}(x) = 1$ iff $x \in [a, b]$

- Show that $\{0, 1\}$ is shattered

VC dimension — Examples

Intervals: $\mathcal{X} = \mathbb{R}$, $\mathcal{H} = \{h_{a,b} : a < b \in \mathbb{R}\}$, where $h_{a,b}(x) = 1$ iff $x \in [a, b]$

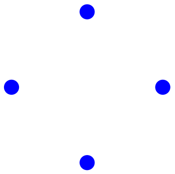
- Show that $\{0, 1\}$ is shattered
- Show that any three points cannot be shattered

VC dimension — Examples

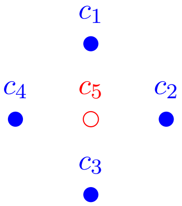
Axis aligned rectangles: $\mathcal{X} = \mathbb{R}^2$,
$\mathcal{H} = \{h_{(a_1,a_2,b_1,b_2)} : a_1 < a_2 \text{ and } b_1 < b_2\}$, where $h_{(a_1,a_2,b_1,b_2)}(x_1, x_2) = 1$ iff $x_1 \in [a_1, a_2]$ and $x_2 \in [b_1, b_2]$

Show:

Shattered



Not Shattered



VC dimension — Examples

Finite classes:

- Show that the VC dimension of a finite $\mathcal{H}$ is at most $\log_2(|\mathcal{H}|)$.

VC dimension — Examples

Finite classes:

- Show that the VC dimension of a finite $\mathcal{H}$ is at most $\log_2(|\mathcal{H}|)$.
- Show that there can be arbitrary gap between $\mathrm{VCdim}(\mathcal{H})$ and $\log_2(|\mathcal{H}|)$

VC dimension — Examples

Halfspaces: $\mathcal{X} = \mathbb{R}^d$, $\mathcal{H} = \{\mathbf{x} \mapsto \mathrm{sign}(\langle \mathbf{w}, \mathbf{x} \rangle) : \mathbf{w} \in \mathbb{R}^d\}$

- Show that $\{\mathbf{e}_1, \ldots, \mathbf{e}_d\}$ is shattered

VC dimension — Examples

Halfspaces: $\mathcal{X} = \mathbb{R}^d$, $\mathcal{H} = \{\mathbf{x} \mapsto \mathrm{sign}(\langle \mathbf{w}, \mathbf{x} \rangle) : \mathbf{w} \in \mathbb{R}^d\}$

- Show that $\{\mathbf{e}_1, \ldots, \mathbf{e}_d\}$ is shattered
- Show that any $d+1$ points cannot be shattered

The Fundamental Theorem of Statistical Learning

Theorem (The Fundamental Theorem of Statistical Learning)

Let $\mathcal{H}$ be a hypothesis class of binary classifiers. Then, there are absolute constants C_1, C_2 such that the sample complexity of PAC learning $\mathcal{H}$ is

$$C_1 \frac{d + \log(1/\delta)}{\epsilon} \leq m_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d \log(1/\epsilon) + \log(1/\delta)}{\epsilon}$$

Furthermore, this sample complexity is achieved by the ERM learning rule.

Proof of the lower bound – main ideas

- Suppose $\mathrm{VCdim}(\mathcal{H}) = d$ and let $C = \{x_1, \ldots, x_d\}$ be a shattered set

Proof of the lower bound – main ideas

- Suppose $\mathrm{VCdim}(\mathcal{H}) = d$ and let $C = \{x_1, \ldots, x_d\}$ be a shattered set
- Consider the distribution $\mathcal{D}$ supported on C s.t.

$$\mathcal{D}(\{x_i\}) = \begin{cases} 1 - 4\epsilon & \text{if } i = 1 \\ 4\epsilon/(d-1) & \text{if } i > 1 \end{cases}$$

Proof of the lower bound – main ideas

- Suppose $\mathrm{VCdim}(\mathcal{H}) = d$ and let $C = \{x_1, \ldots, x_d\}$ be a shattered set
- Consider the distribution $\mathcal{D}$ supported on C s.t.

$$\mathcal{D}(\{x_i\}) = \begin{cases} 1 - 4\epsilon & \text{if } i = 1 \\ 4\epsilon/(d-1) & \text{if } i > 1 \end{cases}$$

- If we see m i.i.d. examples then the expected number of examples from $C \setminus \{x_1\}$ is $4\epsilon m$

Proof of the lower bound – main ideas

- Suppose $\mathrm{VCdim}(\mathcal{H}) = d$ and let $C = \{x_1, \ldots, x_d\}$ be a shattered set
- Consider the distribution $\mathcal{D}$ supported on C s.t.

$$\mathcal{D}(\{x_i\}) = \begin{cases} 1 - 4\epsilon & \text{if } i = 1 \\ 4\epsilon/(d-1) & \text{if } i > 1 \end{cases}$$

- If we see m i.i.d. examples then the expected number of examples from $C \setminus \{x_1\}$ is $4\epsilon m$
- If $m < \frac{d-1}{8\epsilon}$ then $4\epsilon m < \frac{d-1}{2}$ and therefore, we have no information on the labels of at least half the examples in $C \setminus \{x_1\}$

Proof of the lower bound – main ideas

- Suppose $\mathrm{VCdim}(\mathcal{H}) = d$ and let $C = \{x_1, \ldots, x_d\}$ be a shattered set
- Consider the distribution $\mathcal{D}$ supported on C s.t.

$$\mathcal{D}(\{x_i\}) = \begin{cases} 1 - 4\epsilon & \text{if } i = 1 \\ 4\epsilon/(d-1) & \text{if } i > 1 \end{cases}$$

- If we see m i.i.d. examples then the expected number of examples from $C \setminus \{x_1\}$ is $4\epsilon m$
- If $m < \frac{d-1}{8\epsilon}$ then $4\epsilon m < \frac{d-1}{2}$ and therefore, we have no information on the labels of at least half the examples in $C \setminus \{x_1\}$
- Best we can do is to guess, but then our error is $\geq \frac{1}{2} \cdot 2\epsilon = \epsilon$

Proof of the upper bound – main ideas

- Recall our proof for finite class:

Proof of the upper bound – main ideas

- Recall our proof for finite class:
  - For a single hypothesis, we've shown that the probability of the event: $L_S(h) = 0$ given that $L_{(\mathcal{D},f)} > \epsilon$ is at most $e^{-\epsilon m}$

Proof of the upper bound – main ideas

- Recall our proof for finite class:
  - For a single hypothesis, we've shown that the probability of the event: $L_S(h) = 0$ given that $L_{(\mathcal{D},f)} > \epsilon$ is at most $e^{-\epsilon m}$
  - Then we applied the union bound over all "bad" hypotheses, to obtain the bound on ERM failure: $|\mathcal{H}|\, e^{-\epsilon m}$

Proof of the upper bound – main ideas

- Recall our proof for finite class:
  - For a single hypothesis, we've shown that the probability of the event: $L_S(h) = 0$ given that $L_{(\mathcal{D},f)} > \epsilon$ is at most $e^{-\epsilon m}$
  - Then we applied the union bound over all "bad" hypotheses, to obtain the bound on ERM failure: $|\mathcal{H}| \, e^{-\epsilon m}$
- If $\mathcal{H}$ is infinite, or very large, the union bound yields a meaningless bound

Proof of the upper bound – main ideas

- The two samples trick: show that

$$\begin{aligned}&\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[\exists h\in\mathcal{H}_B : L_S(h)=0]\\ &\quad\le 2\mathop{\mathbb{P}}_{S,T\sim\mathcal{D}^m}[\exists h\in\mathcal{H}_B : L_S(h)=0 \text{ and } L_T(h)\ge\epsilon/2]\end{aligned}$$

Proof of the upper bound – main ideas

- The two samples trick: show that

$$\underset{S\sim\mathcal{D}^m}{\mathbb{P}}[\exists h \in \mathcal{H}_B : L_S(h) = 0]$$
$$\leq 2 \underset{S,T\sim\mathcal{D}^m}{\mathbb{P}}[\exists h \in \mathcal{H}_B : L_S(h) = 0 \text{ and } L_T(h) \geq \epsilon/2]$$

- Symmetrization: Since S, T are i.i.d., we can think on first sampling $2m$ examples and then splitting them to S, T at random

Proof of the upper bound – main ideas

- The two samples trick: show that

$$\begin{aligned}&\mathbb{P}_{S\sim\mathcal{D}^m}[\exists h \in \mathcal{H}_B : L_S(h) = 0] \\ &\quad \le 2 \mathbb{P}_{S,T\sim\mathcal{D}^m}[\exists h \in \mathcal{H}_B : L_S(h) = 0 \text{ and } L_T(h) \ge \epsilon/2]\end{aligned}$$

- Symmetrization: Since S, T are i.i.d., we can think on first sampling $2m$ examples and then splitting them to S, T at random
- If we fix h, and $S \cup T$, the probability to have $L_S(h) = 0$ while $L_T(h) \ge \epsilon/2$ is $\le e^{-\epsilon m/4}$

Proof of the upper bound – main ideas

- The two samples trick: show that

$$\begin{aligned}&\mathbb{P}_{S\sim\mathcal{D}^m}[\exists h \in \mathcal{H}_B : L_S(h) = 0] \\ &\leq 2 \mathbb{P}_{S,T\sim\mathcal{D}^m}[\exists h \in \mathcal{H}_B : L_S(h) = 0 \text{ and } L_T(h) \geq \epsilon/2]\end{aligned}$$

- Symmetrization: Since S, T are i.i.d., we can think on first sampling $2m$ examples and then splitting them to S, T at random
- If we fix h, and $S \cup T$, the probability to have $L_S(h) = 0$ while $L_T(h) \geq \epsilon/2$ is $\leq e^{-\epsilon m/4}$
- Once we fixed $S \cup T$, we can take a union bound only over $\mathcal{H}_{S\cup T}$!

Proof of the upper bound – main ideas

Lemma (Sauer-Shelah-Perles^2-Vapnik-Chervonenkis)

Let $\mathcal{H}$ *be a hypothesis class with* $\mathrm{VCdim}(\mathcal{H}) \leq d < \infty$. *Then, for all* $C \subset \mathcal{X}$ *s.t.* $|C| = m > d + 1$ *we have*

$$|\mathcal{H}_C| \leq \left(\frac{em}{d}\right)^d$$

Outline

ERM for halfspaces

- Recall:

$$\mathcal{H} = \{\mathbf{x} \mapsto \mathrm{sign}(\langle \mathbf{w}, \mathbf{x} \rangle) : \mathbf{w} \in \mathbb{R}^d\}$$

ERM for halfspaces

- Recall:
$$\mathcal{H} = \{\mathbf{x} \mapsto \mathrm{sign}(\langle \mathbf{w}, \mathbf{x} \rangle) : \mathbf{w} \in \mathbb{R}^d\}$$
- ERM for Halfspaces:
given $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ find $\mathbf{w}$ s.t. for all i, $\mathrm{sign}(\langle \mathbf{w}, \mathbf{x}_i \rangle) = y_i$.

ERM for halfspaces

- Recall:
$$\mathcal{H} = \{\mathbf{x} \mapsto \mathrm{sign}(\langle \mathbf{w}, \mathbf{x} \rangle) : \mathbf{w} \in \mathbb{R}^d\}$$
- ERM for Halfspaces:
given $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ find $\mathbf{w}$ s.t. for all i, $\mathrm{sign}(\langle \mathbf{w}, \mathbf{x}_i \rangle) = y_i$.
- Cast as a Linear Program:
Find $\mathbf{w}$ s.t.
$$\forall i, \quad y_i \langle \mathbf{w}, \mathbf{x}_i \rangle > 0 \ .$$

ERM for halfspaces

- Recall:
$$\mathcal{H} = \{\mathbf{x} \mapsto \mathrm{sign}(\langle \mathbf{w}, \mathbf{x} \rangle) : \mathbf{w} \in \mathbb{R}^d\}$$
- ERM for Halfspaces:
given $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ find $\mathbf{w}$ s.t. for all i, $\mathrm{sign}(\langle \mathbf{w}, \mathbf{x}_i \rangle) = y_i$.
- Cast as a Linear Program:
Find $\mathbf{w}$ s.t.
$$\forall i, \quad y_i \langle \mathbf{w}, \mathbf{x}_i \rangle > 0 \ .$$
- Can solve efficiently using standard methods

ERM for halfspaces

- Recall:
$$\mathcal{H} = \{\mathbf{x} \mapsto \mathrm{sign}(\langle \mathbf{w}, \mathbf{x} \rangle) : \mathbf{w} \in \mathbb{R}^d\}$$
- ERM for Halfspaces:
given $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ find $\mathbf{w}$ s.t. for all i, $\mathrm{sign}(\langle \mathbf{w}, \mathbf{x}_i \rangle) = y_i$.
- Cast as a Linear Program:
Find $\mathbf{w}$ s.t.
$$\forall i, \quad y_i \langle \mathbf{w}, \mathbf{x}_i \rangle > 0 \ .$$
- Can solve efficiently using standard methods
- Exercise: show how to solve the above Linear Program using the Ellipsoid learner from the previous lecture

ERM for halfspaces using the Perceptron Algorithm

Perceptron

initialize: $\mathbf{w} = (0, \ldots, 0) \in \mathbb{R}^d$
while $\exists i$ s.t. $y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \leq 0$
$\mathbf{w} \leftarrow \mathbf{w} + y_i \mathbf{x}_i$

- Dates back at least to Rosenblatt 1958.

Analysis

Theorem (Agmon'54, Novikoff'62)

Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ *be a sequence of examples such that there exists* $\mathbf{w}^* \in \mathbb{R}^d$ *such that for every* i, $y_i \langle \mathbf{w}^*, \mathbf{x}_i \rangle \geq 1$. *Then, the Perceptron will make at most*

$$\|\mathbf{w}^*\|^2 \max_i \|\mathbf{x}_i\|^2$$

updates before breaking with an ERM halfspace.

Analysis

Theorem (Agmon'54, Novikoff'62)

Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ *be a sequence of examples such that there exists* $\mathbf{w}^* \in \mathbb{R}^d$ *such that for every* i, $y_i \langle \mathbf{w}^*, \mathbf{x}_i \rangle \geq 1$. *Then, the Perceptron will make at most*

$$\|\mathbf{w}^*\|^2 \max_i \|\mathbf{x}_i\|^2$$

updates before breaking with an ERM halfspace.

- The condition would always hold if the data is realizable by some halfspace
- However, $\|\mathbf{w}^*\|$ might be very large
- In many practical cases, $\|\mathbf{w}^*\|$ would not be too large

Proof

- Let $\mathbf{w}^{(t)}$ be the value of $\mathbf{w}$ at iteration t

Proof

- Let $\mathbf{w}^{(t)}$ be the value of $\mathbf{w}$ at iteration t
- Let $(\mathbf{x}_t, y_t)$ be the example used to update $\mathbf{w}$ at iteration t

Proof

- Let $\mathbf{w}^{(t)}$ be the value of $\mathbf{w}$ at iteration t
- Let $(\mathbf{x}_t, y_t)$ be the example used to update $\mathbf{w}$ at iteration t
- Denote $R = \max_i \|\mathbf{x}_i\|$

Proof

- Let $\mathbf{w}^{(t)}$ be the value of $\mathbf{w}$ at iteration t
- Let $(\mathbf{x}_t, y_t)$ be the example used to update $\mathbf{w}$ at iteration t
- Denote $R = \max_i \|\mathbf{x}_i\|$
- The cosine of the angle between $\mathbf{w}^*$ and $\mathbf{w}^{(t)}$ is $\frac{\langle \mathbf{w}^{(t)}, \mathbf{w}^* \rangle}{\|\mathbf{w}^{(t)}\| \, \|\mathbf{w}^*\|}$

Proof

- Let $\mathbf{w}^{(t)}$ be the value of $\mathbf{w}$ at iteration t
- Let $(\mathbf{x}_t, y_t)$ be the example used to update $\mathbf{w}$ at iteration t
- Denote $R = \max_i \|\mathbf{x}_i\|$
- The cosine of the angle between $\mathbf{w}^*$ and $\mathbf{w}^{(t)}$ is $\frac{\langle \mathbf{w}^{(t)}, \mathbf{w}^* \rangle}{\|\mathbf{w}^{(t)}\| \, \|\mathbf{w}^*\|}$
- By the Cauchy-Schwartz inequality, this is always ≤ 1

Proof

- Let $\mathbf{w}^{(t)}$ be the value of $\mathbf{w}$ at iteration t
- Let $(\mathbf{x}_t, y_t)$ be the example used to update $\mathbf{w}$ at iteration t
- Denote $R = \max_i \|\mathbf{x}_i\|$
- The cosine of the angle between $\mathbf{w}^*$ and $\mathbf{w}^{(t)}$ is $\frac{\langle \mathbf{w}^{(t)}, \mathbf{w}^* \rangle}{\|\mathbf{w}^{(t)}\| \, \|\mathbf{w}^*\|}$
- By the Cauchy-Schwartz inequality, this is always ≤ 1
- We will show:

Proof

- Let $\mathbf{w}^{(t)}$ be the value of $\mathbf{w}$ at iteration t
- Let $(\mathbf{x}_t, y_t)$ be the example used to update $\mathbf{w}$ at iteration t
- Denote $R = \max_i \|\mathbf{x}_i\|$
- The cosine of the angle between $\mathbf{w}^*$ and $\mathbf{w}^{(t)}$ is $\frac{\langle \mathbf{w}^{(t)}, \mathbf{w}^* \rangle}{\|\mathbf{w}^{(t)}\| \, \|\mathbf{w}^*\|}$
- By the Cauchy-Schwartz inequality, this is always ≤ 1
- We will show:
  1. $\langle \mathbf{w}^{(t+1)}, \mathbf{w}^* \rangle \geq t$

Proof

- Let $\mathbf{w}^{(t)}$ be the value of $\mathbf{w}$ at iteration t
- Let $(\mathbf{x}_t, y_t)$ be the example used to update $\mathbf{w}$ at iteration t
- Denote $R = \max_i \|\mathbf{x}_i\|$
- The cosine of the angle between $\mathbf{w}^*$ and $\mathbf{w}^{(t)}$ is $\frac{\langle \mathbf{w}^{(t)}, \mathbf{w}^* \rangle}{\|\mathbf{w}^{(t)}\| \, \|\mathbf{w}^*\|}$
- By the Cauchy-Schwartz inequality, this is always ≤ 1
- We will show:
  1. $\langle \mathbf{w}^{(t+1)}, \mathbf{w}^* \rangle \geq t$
  2. $\|\mathbf{w}^{(t+1)}\| \leq R\sqrt{t}$

Proof

- Let $\mathbf{w}^{(t)}$ be the value of $\mathbf{w}$ at iteration t
- Let $(\mathbf{x}_t, y_t)$ be the example used to update $\mathbf{w}$ at iteration t
- Denote $R = \max_i \|\mathbf{x}_i\|$
- The cosine of the angle between $\mathbf{w}^*$ and $\mathbf{w}^{(t)}$ is $\frac{\langle \mathbf{w}^{(t)}, \mathbf{w}^* \rangle}{\|\mathbf{w}^{(t)}\| \, \|\mathbf{w}^*\|}$
- By the Cauchy-Schwartz inequality, this is always ≤ 1
- We will show:
  1. $\langle \mathbf{w}^{(t+1)}, \mathbf{w}^* \rangle \geq t$
  2. $\|\mathbf{w}^{(t+1)}\| \leq R\sqrt{t}$
- This would yield

$$\frac{t}{R\sqrt{t}\,\|\mathbf{w}^*\|} \leq \frac{\langle \mathbf{w}^{(t)}, \mathbf{w}^* \rangle}{\|\mathbf{w}^{(t)}\| \, \|\mathbf{w}^*\|} \leq 1$$

Proof

- Let $\mathbf{w}^{(t)}$ be the value of $\mathbf{w}$ at iteration t
- Let $(\mathbf{x}_t, y_t)$ be the example used to update $\mathbf{w}$ at iteration t
- Denote $R = \max_i \|\mathbf{x}_i\|$
- The cosine of the angle between $\mathbf{w}^*$ and $\mathbf{w}^{(t)}$ is $\frac{\langle \mathbf{w}^{(t)}, \mathbf{w}^* \rangle}{\|\mathbf{w}^{(t)}\| \, \|\mathbf{w}^*\|}$
- By the Cauchy-Schwartz inequality, this is always ≤ 1
- We will show:
  1. $\langle \mathbf{w}^{(t+1)}, \mathbf{w}^* \rangle \geq t$
  2. $\|\mathbf{w}^{(t+1)}\| \leq R\sqrt{t}$
- This would yield

$$\frac{t}{R\sqrt{t}\,\|\mathbf{w}^*\|} \leq \frac{\langle \mathbf{w}^{(t)}, \mathbf{w}^* \rangle}{\|\mathbf{w}^{(t)}\| \, \|\mathbf{w}^*\|} \leq 1$$

- Rearranging the above would yield $t \leq \|\mathbf{w}^*\|^2 R^2$ as required.

Proof (Cont.)

Showing $\langle \mathbf{w}^{(t+1)}, \mathbf{w}^* \rangle \geq t$

- Initially, $\langle \mathbf{w}^{(1)}, \mathbf{w}^* \rangle = 0$

Showing $\|\mathbf{w}^{(t+1)}\|^2 \leq R^2\, t$

Proof (Cont.)

Showing $\langle \mathbf{w}^{(t+1)}, \mathbf{w}^* \rangle \geq t$

- Initially, $\langle \mathbf{w}^{(1)}, \mathbf{w}^* \rangle = 0$
- Whenever we update, $\langle \mathbf{w}^{(t)}, \mathbf{w}^* \rangle$ increases by at least 1:

$$\langle \mathbf{w}^{(t+1)}, \mathbf{w}^* \rangle = \langle \mathbf{w}^{(t)} + y_t \mathbf{x}_t, \mathbf{w}^* \rangle = \langle \mathbf{w}^{(t)}, \mathbf{w}^* \rangle + \underbrace{y_t \langle \mathbf{x}_t, \mathbf{w}^* \rangle}_{\geq 1}$$

Showing $\|\mathbf{w}^{(t+1)}\|^2 \leq R^2\, t$

Proof (Cont.)

Showing $\langle \mathbf{w}^{(t+1)}, \mathbf{w}^* \rangle \geq t$

- Initially, $\langle \mathbf{w}^{(1)}, \mathbf{w}^* \rangle = 0$
- Whenever we update, $\langle \mathbf{w}^{(t)}, \mathbf{w}^* \rangle$ increases by at least 1:

$$\langle \mathbf{w}^{(t+1)}, \mathbf{w}^* \rangle = \langle \mathbf{w}^{(t)} + y_t \mathbf{x}_t, \mathbf{w}^* \rangle = \langle \mathbf{w}^{(t)}, \mathbf{w}^* \rangle + \underbrace{y_t \langle \mathbf{x}_t, \mathbf{w}^* \rangle}_{\geq 1}$$

Showing $\|\mathbf{w}^{(t+1)}\|^2 \leq R^2\, t$

- Initially, $\|\mathbf{w}^{(1)}\|^2 = 0$

Proof (Cont.)

Showing $\langle \mathbf{w}^{(t+1)}, \mathbf{w}^* \rangle \geq t$

- Initially, $\langle \mathbf{w}^{(1)}, \mathbf{w}^* \rangle = 0$
- Whenever we update, $\langle \mathbf{w}^{(t)}, \mathbf{w}^* \rangle$ increases by at least 1:

$$\langle \mathbf{w}^{(t+1)}, \mathbf{w}^* \rangle = \langle \mathbf{w}^{(t)} + y_t \mathbf{x}_t, \mathbf{w}^* \rangle = \langle \mathbf{w}^{(t)}, \mathbf{w}^* \rangle + \underbrace{y_t \langle \mathbf{x}_t, \mathbf{w}^* \rangle}_{\geq 1}$$

Showing $\|\mathbf{w}^{(t+1)}\|^2 \leq R^2\, t$

- Initially, $\|\mathbf{w}^{(1)}\|^2 = 0$
- Whenever we update, $\|\mathbf{w}^{(t)}\|^2$ increases by at most 1:

$$\begin{aligned} \|\mathbf{w}^{(t+1)}\|^2 = \|\mathbf{w}^{(t)} + y_t \mathbf{x}_t\|^2 &= \|\mathbf{w}^{(t)}\|^2 + \underbrace{2 y_t \langle \mathbf{w}^{(t)}, \mathbf{x}_t \rangle}_{\leq 0} + y_t^2 \|\mathbf{x}_t\|^2 \\ &\leq \|\mathbf{w}^{(t)}\|^2 + R^2 \ . \end{aligned}$$

Summary

- The PAC Learning model
- What is PAC learnable?
- PAC learning of finite classes using ERM
- The VC dimension and the fundmental theorem of learning
  - Classes of finite VC dimension
- How to PAC learn?
  - Using ERM
- Learning halfspaces using: Linear programming, Ellipsoid, Perceptron