Summary

- Decomposition of risk into estimation, approximation, generalization, optimization.
- Decoupling of concerns for some problems (SVM) versus tangled concerns for others (DL).
- General thought process: (a) we carefully identify the tightest class of predictors $\mathcal{F}$ considered by the algorithm on this particular data, (b) generalization seems to worsen with some intricate notion of size of $\mathcal{F}$.
- Next week: *statistical* learning theory, focusing on estimation and generalization.
\[ |\langle a, b \rangle| \leq \|a\|_{12} \cdot \|b\|_{12} \]

\[ \langle w, x \rangle \leq \|w\|_1 \cdot \|x\|_1 \]

Part 2...

* Hwk3 (midterm prep).

* Hwk2 scores tonight.
6. Approximation and generalization
Approximation and generalization

Last time: main errors are approximation error and generalization error.

Approximation error drops with model size; generalization error grows.

Today and next time we'll focus on generalization error.
Approximation and generalization

Last time: main errors are approximation error and generalization error.

Approximation error drops with model size; generalization error grows.

Today and next time we'll focus on generalization error.

Side issue: we don’t know how to measure “deep network model size”.
Example: RBF SVM

Recall: \( \|w\|_2^2 \leq \frac{2}{\lambda} \), or \( \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \leq \frac{2}{\lambda} \); thus \( C \) gives model size (where \( C' = \frac{1}{\lambda n} \)).
SVM primal (nonseparable):

\[
\begin{align*}
\text{min} & \quad \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^{n} \left[1 - y_i (x_i \cdot w)\right]^+ \\
\text{subject to} & \quad \sum_{i=1}^{n} y_i \alpha_i = 0 \\
& \quad \alpha_i \geq 0, i = 1, \ldots, n
\end{align*}
\]

SVM dual (with kernels):

\[
\begin{align*}
\text{max} & \quad \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j) - \frac{1}{2} \sum_{i} \alpha_i \\
\text{subject to} & \quad \sum_{i} y_i \alpha_i = 0, \\
& \quad 0 \leq \alpha_i \leq C, i = 1, \ldots, n
\end{align*}
\]
(binary case)

\[
R \left( h \right) = \frac{1}{n} \sum_{i=1}^{n} l \left( h(x_i), y_i \right)
\]

loss margin

**hinge loss:**

\[
l_h(z) = \max \left\{ 1 - z \right\}_+
\]

\[= \text{ReLU}(1 - z)\]

**zero-one loss:**

\[
l_{01}(\hat{y}, y) = \mathbb{I}[\hat{y} \neq y]
\]
Example: RBF SVM

Small $C \Rightarrow$ Small model class

$C = 0.040000$, $\lambda = 0.250000$; train 0.050000, test 0.076000

RBF kernel: $k(a, b) = \exp\left(-\frac{||a-b||^2}{2\sigma^2}\right)$
Example: RBF SVM

$C = 0.055546, \lambda = 0.180031$ ; train 0.050000, test 0.076000
Example: RBF SVM

\[ C = 0.077134, \lambda = 0.129644; \text{ train 0.050000, test 0.076000} \]
Example: RBF SVM

\[ C = 0.107112, \lambda = 0.093360; \text{ train } 0.050000, \text{ test } 0.076000 \]
Example: RBF SVM

\[ C = 0.148742, \lambda = 0.067231; \text{ train 0.050000, test 0.076000} \]
Example: RBF SVM

\[ C = 0.206550, \lambda = 0.048414 \; ; \; \text{train} \; 0.050000, \; \text{test} \; 0.078000 \]
Example: RBF SVM

\[ C = 0.286826, \lambda = 0.034864; \text{ train } 0.030000, \text{ test } 0.074000 \]
Example: RBF SVM

\[ C = 0.398302, \lambda = 0.025107; \text{ train } 0.020000, \text{ test } 0.068000 \]
Example: RBF SVM

$C = 0.553102, \lambda = 0.018080; \text{ train } 0.010000, \text{ test } 0.064000$
Example: RBF SVM

\[ C = 0.768066, \; \lambda = 0.013020 ; \; \text{train} \; 0.010000, \; \text{test} \; 0.064000 \]
Example: RBF SVM

\[ C = 1.066575, \lambda = 0.009376; \text{ train 0.000000, test 0.070000} \]
Example: RBF SVM

\[ C = 1.481101 \text{, } \lambda = 0.006752 \text{ ; train } 0.010000 \text{, test } 0.060000 \]
Example: RBF SVM

\[ C = 2.056733 \text{ , } \lambda = 0.004862 \text{ ; train 0.010000 , test 0.060000} \]
Example: RBF SVM

\[ C = 2.856085 \, , \, \lambda = 0.003501 \, ; \, \text{train 0.010000, test 0.066000} \]
Example: RBF SVM

\[ C = 3.966103, \lambda = 0.002521; \text{ train } 0.000000, \text{ test } 0.066000 \]
Example: RBF SVM

\[ C = 5.507534, \lambda = 0.001816; \text{ train } 0.000000, \text{ test } 0.060000 \]
Example: RBF SVM

\[ C = 7.648044, \lambda = 0.001308; \text{ train } 0.000000, \text{ test } 0.078000 \]
Example: RBF SVM

\[ C = 10.620466, \lambda = 0.000942 \qquad \text{train 0.000000, test 0.076000} \]
Example: RBF SVM

\[ C = 14.748124, \lambda = 0.000678; \text{ train 0.000000, test 0.076000} \]
Example: RBF SVM

$C = 20.480000, \lambda = 0.000488$ ; train 0.000000 , test 0.076000
Example: RBF SVM

Decreasing regularization made the model more complicated:
Example: RBF SVM

The best model was still not that close.

\[ C = 1.481101, \lambda = 0.006752 ; \text{ train 0.010000, test 0.060000} \]
Example: RBF SVM

The best model was still not that close.

\[ C = 1.481101, \lambda = 0.006752; \text{ train } 0.010000, \text{ test } 0.060000 \]
Example: polynomial

**Question:** Polynomial least squares of which degree?

**Answer:** data \((x, y)\) has \(y \sim \mathcal{N}(0, 1)\) !!!

**Method:** ordinary least squares, monomial features.
Example: polynomial

**Question:** Polynomial least squares of which degree?

**Answer:** data \((x, y)\) has \(y \sim \mathcal{N}(0, 1)\) !!!

**Method:** ordinary least squares, monomial features.

\[
\hat{R}_1 = 0.00692304, R_1 = 0.00897457, \hat{R}_2 = 0.00690718, R_2 = 0.00870077
\]

(training data is red, testing data is green.)
Example: polynomial

**Question:** Polynomial least squares of which degree?

**Answer:** data \((x, y)\) has \(y \sim \mathcal{N}(0, 1)\) !!!

**Method:** ordinary least squares, monomial features.

\[
\hat{R}_1 = 0.00692304, \quad R_1 = 0.00897457, \quad \hat{R}_3 = 0.0064944, \quad R_3 = 0.00998528
\]

(training data is red, testing data is green.)
**Example: polynomial**

**Question:** Polynomial least squares of which degree?  
**Answer:** data \((x, y)\) has \(y \sim \mathcal{N}(0, 1)\) !!!  
**Method:** ordinary least squares, monomial features.

\[
\hat{R}_1 = 0.00692304, R_1 = 0.00897457, \hat{R}_4 = 0.0062397, R_4 = 0.013063
\]

(training data is red, testing data is green.)
**Example: polynomial**

**Question:** Polynomial least squares of which degree?
**Answer:** data \((x, y)\) has \(y \sim \mathcal{N}(0, 1)\) !!!
**Method:** ordinary least squares, monomial features.

\[
\hat{R}_1 = 0.00692304, R_1 = 0.00897457, \hat{R}_5 = 0.00582684, R_5 = 0.00975194
\]

(training data is red, testing data is green.)
Example: polynomial

**Question:** Polynomial least squares of which degree?

**Answer:** data \((x, y)\) has \(y \sim \mathcal{N}(0, 1)\) !!!

**Method:** ordinary least squares, monomial features.

\[
\hat{R}_1 = 0.00692304, \ R_1 = 0.00897457, \ \hat{R}_6 = 0.00571136, \ R_6 = 0.0142185
\]
Example: polynomial

**Question:** Polynomial least squares of which degree?
**Answer:** data \((x, y)\) has \(y \sim \mathcal{N}(0, 1)\) !!!

**Method:** ordinary least squares, monomial features.

\[
\hat{R}_1 = 0.00692304,\ R_1 = 0.00897457,\ \hat{R}_7 = 0.00565266,\ R_7 = 0.0282631
\]

(training data is red, testing data is green.)
Example: polynomial

**Question:** Polynomial least squares of which degree?

**Answer:** data \((x, y)\) has \(y \sim \mathcal{N}(0, 1)\) !!!

**Method:** ordinary least squares, monomial features.

\[
\hat{R}_1 = 0.00692304, \quad R_1 = 0.00897457, \quad \hat{R}_8 = 0.00564127, \quad R_8 = 0.0440347
\]

(training data is red, testing data is green.)
Example: polynomial

**Question:** Polynomial least squares of which degree?

**Answer:** data \((x, y)\) has \(y \sim \mathcal{N}(0, 1)\) !!!

**Method:** ordinary least squares, monomial features.

\[
\hat{R}_1 = 0.00692304, R_1 = 0.00897457, \hat{R}_9 = 0.00541878, R_9 = 0.42463
\]

(training data is red, testing data is green.)
**Example: polynomial**

**Question:** Polynomial least squares of which degree?

**Answer:** data \((x, y)\) has \(y \sim \mathcal{N}(0, 1)\) !!

**Method:** ordinary least squares, monomial features.

\[
\hat{R}_1 = 0.00692304, R_1 = 0.00897457, \hat{R}_{11} = 0.00513565, R_{11} = 9.33301
\]

(training data is red, testing data is green.)
Question: Polynomial least squares of which degree?
Answer: data \((x, y)\) has \(y \sim \mathcal{N}(0, 1)\) !!!
Method: ordinary least squares, monomial features.

\[
\hat{R}_1 = 0.00692304, R_1 = 0.00897457, \hat{R}_{12} = 0.00503499, R_{12} = 40.6796
\]
**Example: polynomial**

**Question:** Polynomial least squares of which degree?
**Answer:** data \((x, y)\) has \(y \sim \mathcal{N}(0, 1)\) !!!
**Method:** ordinary least squares, monomial features.

\[
\hat{R}_1 = 0.00692304, R_1 = 0.00897457, \hat{R}_{13} = 0.00503326, R_{13} = 30.7593
\]

(training data is red, testing data is green.)
Example: polynomial

**Question:** Polynomial least squares of which degree?

**Answer:** data \((x, y)\) has \(y \sim \mathcal{N}(0, 1)\) !!!

**Method:** ordinary least squares, monomial features.

\[
\hat{R}_1 = 0.00692304, R_1 = 0.00897457, \hat{R}_{14} = 0.00444016, R_{14} = 985.645
\]

(training data is red, testing data is green.)
Example: polynomial

**Question:** Polynomial least squares of which degree?

**Answer:** data \((x, y)\) has \(y \sim \mathcal{N}(0, 1)\) !!!

**Method:** ordinary least squares, monomial features.

\[
\hat{R}_1 = 0.00692304, R_1 = 0.00897457, \hat{R}_{15} = 0.00361723, R_{15} = 13158.5
\]

(training data is red, testing data is green.)
**Example: polynomial**

**Question:** Polynomial least squares of which degree?

**Answer:** data \((x, y)\) has \(y \sim \mathcal{N}(0, 1) \) !!!

**Method:** ordinary least squares, monomial features.

\[
\hat{R}_1 = 0.00692304, R_1 = 0.00897457, \hat{R}_{16} = 0.00312787, R_{16} = 30910.6
\]

*(training data is red, testing data is green.)*
Our goal: to understand the math behind generalization error.
7. Binomials, random walks, coin tosses, classification
Zero-one loss and Bernoulli random variables

Consider a fixed binary classifier \( h : \mathcal{X} \rightarrow \{-1, +1\} \).
(We’ll explain “fixed” in some slides.)
Consider a fixed binary classifier $h : \mathcal{X} \rightarrow \{-1, +1\}$.
(We’ll explain “fixed” in some slides.)

- Suppose $(X_i, Y_i)$ are drawn IID from some distribution on $(X, Y)$.
This is the setting of statistical learning theory;
by having a link between past and future data, we can learn.
Consider a fixed binary classifier $h : \mathcal{X} \rightarrow \{-1, +1\}$.
(We’ll explain “fixed” in some slides.)

- Suppose $(X_i, Y_i)$ are drawn IID from some distribution on $(X, Y)$.
  This is the setting of statistical learning theory;
  by having a link between past and future data, we can learn.
- Define $Z_i := \mathbb{1}[h(X_i) \neq Y_i]$. \

Then $(Z_i)_{i=1}^n$ are IID Bernoulli random variables!
Their common mean satisfies $\mathbb{E} Z_i = \mathbb{E} \mathbb{1}[h(X) \neq Y] = \Pr[h(X) \neq Y] = R_0 / 1(h)$, which is the true/population zero-one risk!
Consider a fixed binary classifier $h : \mathcal{X} \to \{-1, +1\}$. (We’ll explain “fixed” in some slides.)

▶ Suppose $(X_i, Y_i)$ are drawn IID from some distribution on $(X, Y)$. This is the setting of statistical learning theory; by having a link between past and future data, we can learn.

▶ Define $Z_i := 1[h(X_i) \neq Y_i]$.

▶ Then $(Z_i)_{i=1}^n$ are IID Bernoulli random variables!
Consider a **fixed binary classifier** $h : \mathcal{X} \to \{-1, +1\}$. (We’ll explain “fixed” in some slides.)

- Suppose $(X_i, Y_i)$ are drawn IID from some distribution on $(X, Y)$. This is the setting of **statistical learning theory**; by having a link between past and future data, we can learn.
- Define $Z_i := 1[h(X_i) \neq Y_i]$.
- Then $(Z_i)_{i=1}^n$ are IID Bernoulli random variables!
- Their common mean satisfies
  \[ \mathbb{E}Z_i = \mathbb{E}1[h(X_i) \neq Y_i] = \text{Pr}[h(X) \neq Y] = \mathcal{R}_{0/1}(h), \]
  which is the **true/population zero-one risk**!
Law of Large Numbers (LLN)

**Fact (super informal).** Let $Z_i$ record the $i$th flip of a coin with bias $p$. Then

$$\frac{1}{n} \sum_{i=1}^{n} Z_i \xrightarrow{n \to \infty} p.$$
Law of Large Numbers (LLN)

**Fact (super informal).** Let $Z_i$ record the $i$th flip of a coin with bias $p$. Then

$$\frac{1}{n} \sum_{i=1}^{n} Z_i \xrightarrow{n \to \infty} p.$$ 

**Fact (informal)** Under some mild conditions (e.g., $Z_i$ IID with finite variance), the limit

$$\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} Z_i$$

defines a random variable which exists and is (almost) always equal to $\mathbb{E}Z_1$. 

"a.s." means "almost surely"; can infinitely many coin flips all be heads?)
Law of Large Numbers (LLN)

**Fact** (super informal). Let $Z_i$ record the $i$th flip of a coin with bias $p$. Then

$$\frac{1}{n} \sum_{i=1}^{n} Z_i \xrightarrow{n \to \infty} p.$$ 

**Fact** (informal) Under some mild conditions (e.g., $Z_i$ IID with finite variance), the limit

$$\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} Z_i$$

defines a random variable which exists and is (almost) always equal to $\mathbb{E}Z_1$.

For rigorous versions: look up “strong” and “weak” laws of large numbers.
Law of Large Numbers (LLN)

**Fact** (super informal). Let $Z_i$ record the $i$th flip of a coin with bias $p$. Then

$$\frac{1}{n} \sum_{i=1}^{n} Z_i \xrightarrow{n \to \infty} p.$$

**Fact** (informal) Under some mild conditions (e.g., $Z_i$ IID with finite variance), the limit

$$\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} Z_i$$

defines a random variable which exists and is (almost) always equal to $\mathbb{E}Z_1$.

For rigorous versions: look up “strong” and “weak” laws of large numbers.

**Relevance to us:** The random variables $(Z_i)_{i=1}^{n}$, defined $Z_i := 1[h(X_i) \neq Y_i]$, satisfy

$$\hat{R}_{0/1}(h) = \frac{1}{n} \sum_{i=1}^{n} 1[h(X_i) \neq Y_i] \xrightarrow{\text{a.s.}} \mathbb{E}1[h(X_1) \neq Y_1] = R_{0,1}(h).$$

(“a.s.” means “almost surely”; can infinitely many coin flips all be heads?)
Finite sample bounds

**Relevance to us** The random variables \((Z_i)_{i=1}^n\), defined \(Z_i := 1[h(X_i) \neq Y_i]\), satisfy

\[
\hat{R}_{0/1}(h) = \frac{1}{n} \sum_{i=1}^{n} 1[h(X_i) \neq Y_i] \xrightarrow{\text{a.s.}} \mathbb{E}1[h(X_1) \neq Y_1] = R_{0,1}(h).
\]

- In machine learning, we want to reason about which algorithms do better for finite \(n\).
- Going back to coin flips, how does \(\frac{1}{n} \sum_{i=1}^{n} Z_i - \mathbb{E}Z_1\) look for finite \(n\)?
Bernoulli walks

Let $Z_i$ be Bernoulli with $EZ_i = 1/2$; consider $\sum_{i=1}^t (2Z_i - 1)$. 
Bernoulli walks

Let $Z_i$ be Bernoulli with $\mathbb{E}Z_i = 1/2$; consider $\sum_{i=1}^t (2Z_i - 1)$.

Fact: with probability $1 - \frac{1}{e}$, position $\leq \sqrt{2t}$.

Thus, with high probability, $R(h) - \hat{R}(h)$ should be $\leq \sqrt{\frac{1}{2n}}$. 
Markov’s inequality

Markov’s inequality. If $S$ is a nonnegative r.v. and $a > 0$, then

$$\Pr[S \geq a] \leq \frac{\mathbb{E}S}{a}.$$  

Proof. Note $a\mathbb{1}[S \geq a] \leq S$ and apply $\mathbb{E}$ to both sides. □
Markov’s inequality

**Markov's inequality.** If $S$ is a nonnegative r.v. and $a > 0$, then

$$\Pr[S \geq a] \leq \frac{\mathbb{E}[S]}{a}.$$  

**Proof.** Note $a \mathbb{1}[S \geq a] \leq S$ and apply $\mathbb{E}$ to both sides. □

Can we use this on $\frac{1}{n} \sum_{i=1}^{n} Z_i = \hat{R}(h)$?

- **Directly**

  $$\Pr \left[ \hat{R}(h) \geq R(h) + \epsilon \right] \leq \frac{\mathbb{E}[\hat{R}(h)]}{\mathbb{E}[R(h)] + \epsilon} = \frac{\mathbb{E}[\hat{R}(h)]}{\mathbb{E}[R(h)] + \epsilon}.$$  

  Not super useful, and no benefit to large $n$!

- **Alternatively,**

  $$\Pr \left[ |\hat{R}(h) - R(h)| \geq \epsilon \right] \leq \frac{\mathbb{E}[|\hat{R}(h) - R(h)|]}{\epsilon},$$

  where it's again not clear what to do, and large $n$ doesn't seem to help.
**Chebyshev’s inequality.** If \( f : \mathbb{R} \to \mathbb{R}_{\geq 0} \) nondecreasing and \( f(a) > 0 \),

\[
\Pr[S \geq a] \leq \frac{\mathbb{E}f(S)}{f(a)}.
\]

For example, \( \Pr[|S - \mathbb{E}S| \geq a] \leq \text{var}(S)/a^2 \).

**Proof.** Note \( \Pr[S \geq a] \leq \Pr[f(S) \geq f(a)] \) and apply Markov. \( \square \)
**Chebyshev’s inequality.** If \( f : \mathbb{R} \to \mathbb{R}_{\geq 0} \) nondecreasing and \( f(a) > 0 \),

\[
\Pr[S \geq a] \leq \frac{\mathbb{E} f(S)}{f(a)}.
\]

For example, \( \Pr[|S - \mathbb{E}S| \geq a] \leq \text{var}(S)/a^2 \).

**Proof.** Note \( \Pr[S \geq a] \leq \Pr[f(S) \geq f(a)] \) and apply Markov. □

- Sometimes the variance form is called “Chebyshev’s inequality”.
- Applying this as before,

\[
\Pr \left[ \left| \hat{\mathcal{R}}(h) - \mathcal{R}(h) \right| \geq \epsilon \right] \leq \frac{\text{var}(\hat{\mathcal{R}}(h))}{\epsilon^2} = \frac{\text{var}(1[h(X) \neq Y])}{n\epsilon^2} \leq \frac{1}{4n\epsilon^2}.
\]

Now we get benefit of \( n! \).
$V_i \text{ iid}$

$\text{Var} \left( \frac{1}{n} \sum_{i=1}^{n} V_i \right) = \frac{1}{n} \text{Var}(V_1)$

Averages concentrate probability mass
**Chebyshev’s inequality.** If \( f : \mathbb{R} \to \mathbb{R}_{\geq 0} \) nondecreasing and \( f(a) > 0 \),

\[
\Pr[S \geq a] \leq \frac{\mathbb{E}f(S)}{f(a)}.
\]

For example, \( \Pr[|S - \mathbb{E}S| \geq a] \leq \text{var}(S)/a^2 \).

**Proof.** Note \( \Pr[S \geq a] \leq \Pr[f(S) \geq f(a)] \) and apply Markov. \( \square \)

- Sometimes the variance form is called “Chebyshev’s inequality”.
- Applying this as before,

\[
\Pr \left[ |\hat{R}(h) - R(h)| \geq \epsilon \right] \leq \frac{\text{var}(\hat{R}(h))}{\epsilon^2} = \frac{\text{var}(\mathbb{1}[h(X) \neq Y])}{ne^2} \leq \frac{1}{4ne^2}.
\]

Now we get benefit of \( n! \)

- Set \( \frac{1}{4ne^2} = \frac{1}{e} \) and solve for \( \epsilon \): with probability \( \geq 1 - \frac{1}{e} \),

\[
R(h) \leq \hat{R}(h) + \sqrt{\frac{e}{4n}}.
\]

This is called “inverting the probability bound”, is the usual form in machine learning.
Let $Z_i$ be bernoulli with $\mathbb{E}Z_i = 1/2$; consider $\sum_{i=1}^{t}(2Z_i - 1)$. 
Bernoulli walks

Let $Z_i$ be bernoulli with $\mathbb{E}Z_i = 1/2$; consider $\sum_{i=1}^{t} (2Z_i - 1)$.

With probability $1 - \frac{1}{e}$, position $\leq \sqrt{2n}$.

A refinement: we want $1 - \delta$, for any confidence parameter $\delta > 0$.

$\delta = \frac{1}{100} \Rightarrow n \text{ with probability } \geq 1 - \frac{1}{100} = 99\%$.
Theorem (Hoeffding’s inequality). Given IID $Z_i \in [a, b]$,

$$
\Pr \left[ \frac{1}{n} \sum_i Z_i - \mathbb{E}Z_1 \geq \epsilon \right] \leq \exp \left( \frac{-2n\epsilon^2}{(b-a)^2} \right).
$$

Alternatively, with probability at least $1 - \delta$,

$$
\frac{1}{n} \sum_{i=1}^n Z_i \leq \mathbb{E}Z_1 + (b-a)\sqrt{\frac{\ln(1/\delta)}{2n}}.
$$

Interpretation: confidence interval
Hoeffding’s inequality

**Theorem (Hoeffding’s inequality).** Given IID $Z_i \in [a, b]$, 

$$\Pr \left[ \frac{1}{n} \sum_{i} Z_i - \mathbb{E}Z_1 \geq \epsilon \right] \leq \exp \left( \frac{-2nc^2}{(b - a)^2} \right).$$

Alternatively, with probability at least $1 - \delta$,

$$\frac{1}{n} \sum_{i=1}^{n} Z_i \leq \mathbb{E}Z_1 + (b - a) \sqrt{\frac{\ln(1/\delta)}{2n}}.$$

**Remarks.**

- Can flip inequality by replacing $Z_i$ with $-Z_i$.
- Using the second (“inverted”) form: with probability at least $1 - \delta$,

  $$\mathcal{R}(h) \leq \hat{\mathcal{R}}(h) + \sqrt{\frac{\ln(1/\delta)}{2n}}.$$

- Alternatively: setting $\delta = 10^{-k}$, with probability at least $99.99 \cdots 9\%$ ($k$ 9s), we have $\mathcal{R}(h) \leq \hat{\mathcal{R}}(h) + \sqrt{k \ln(10)/(2n)}$: to add more bits of confidence, we must increase sample size $n$ linearly.
- Hoeffding captures a “concentration of measure” phenomenon: probability mass concentrates within $[-1/\sqrt{n}, 1/\sqrt{n}]$. 
8. Overfitting
Hoeffding and algorithm output

For fixed \( \hat{f} \), with probability \( \geq 1 - \delta \),
\[
\mathcal{R}(h) \leq \hat{\mathcal{R}}(h) + \sqrt{\frac{\ln(1/\delta)}{2n}}.
\]

Something must break if we apply this to the output of an algorithm running on a larger or smaller collections of predictors. (E.g., SVM with varying \( \lambda \).)
Suppose we observe \(((x_i, y_i))_{i=1}^{n}\) IID, and output

\[
\hat{f}(x) := \begin{cases} 
  y_i & \text{when } x = x_i \\
  \text{“bear”} & \text{otherwise.}
\end{cases}
\]

Can easily have \(\hat{R}_{0/1}(\hat{f}) = 0\) but \(R_{0/1}(\hat{f}) = 1\)!
Suppose we observe \(((x_i, y_i))_{i=1}^n\) IID, and output

\[
\hat{f}(x) := \begin{cases} 
  y_i & \text{when } x = x_i \\
  "\text{bear}" & \text{otherwise.}
\end{cases}
\]

Can easily have \(\hat{R}_{0/1}(\hat{f}) = 0\) but \(R_{0/1}(\hat{f}) = 1!\)

Remarks.
Suppose we observe \( ((x_i, y_i))_{i=1}^n \) IID, and output

\[ \hat{f}(x) : = \begin{cases} y_i & \text{when } x = x_i \\ "bear" & \text{otherwise.} \end{cases} \]

Can easily have \( \hat{R}_{0/1}(\hat{f}) = 0 \) but \( R_{0/1}(\hat{f}) = 1 \)!

**Remarks.**

- \( \hat{f} \) depends on \( ((x_i, y_i))_{i=1}^n \), therefore \( (Z_i)_{i=1}^n \) with \( Z_i = 1[h(x_i) \neq y_i] \) are not all independent!
Suppose we observe \(((x_i, y_i))_{i=1}^n\) IID, and output

\[
\hat{f}(x) := \begin{cases} 
  y_i & \text{when } x = x_i \\
  "\text{bear}" & \text{otherwise.}
\end{cases}
\]

Can easily have \(\hat{R}_{0/1}(\hat{f}) = 0\) but \(R_{0/1}(\hat{f}) = 1\)!

**Remarks.**

▶ \(\hat{f}\) depends on \(((x_i, y_i))_{i=1}^n\), therefore \((Z_i)_{i=1}^n\) with \(Z_i = 1[h(x_i) \neq y_i]\) are not all independent!

▶ Situation less severe but similar with ERM!
Suppose we observe \((x_i, y_i)_{i=1}^n\) IID, and output

\[
\hat{f}(x) := \begin{cases} 
  y_i & \text{when } x = x_i \\
  \text{"bear"} & \text{otherwise.}
\end{cases}
\]

Can easily have \(\hat{R}_{0/1}(\hat{f}) = 0\) but \(R_{0/1}(\hat{f}) = 1\)!

Remarks.

- \(\hat{f}\) depends on \(((x_i, y_i))_{i=1}^n\), therefore \((Z_i)_{i=1}^n\) with \(Z_i = 1[h(x_i) \neq y_i]\) are not all independent!

- Situation less severe but similar with ERM!

- Hence our requirement of "fixed \(h\)"; it must be seen before data!
Suppose we observe \( ((x_i, y_i))_{i=1}^n \) IID, and output

\[
\hat{f}(x) := \begin{cases} 
  y_i & \text{when } x = x_i \\
  \text{“bear”} & \text{otherwise.}
\end{cases}
\]

Can easily have \( \hat{\mathcal{R}}_{0/1}(\hat{f}) = 0 \) but \( \mathcal{R}_{0/1}(\hat{f}) = 1 \)!

**Remarks.**

- \( \hat{f} \) depends on \( ((x_i, y_i))_{i=1}^n \), therefore \( (Z_i)_{i=1}^n \) with \( Z_i = 1[h(x_i) \neq y_i] \) are not all independent!

- Situation less severe but similar with ERM!

- Hence our requirement of “fixed \( h \)”: it must be seen before data!

- One fix is to train and evaluate with different data; we do this with validation sets.
Overfitting example

Suppose we observe $((x_i, y_i))_{i=1}^{n}$ IID, and output

$$\hat{f}(x) := \begin{cases} y_i & \text{when } x = x_i \\ \text{“bear”} & \text{otherwise.} \end{cases}$$

Can easily have $\hat{R}_{0/1}(\hat{f}) = 0$ but $R_{0/1}(\hat{f}) = 1$!

Remarks.

- $\hat{f}$ depends on $((x_i, y_i))_{i=1}^{n}$, therefore $(Z_i)_{i=1}^{n}$ with $Z_i = 1[h(x_i) \neq y_i]$ are not all independent!

- Situation less severe but similar with ERM!

- Hence our requirement of “fixed $h$”: it must be seen before data!

- One fix is to train and evaluate with different data; we do this with validation sets.

- Another fix is a modified bound sensitive to the algorithm!
We will extend our tools to give: with probability \( \geq 1 - \delta \),

\[
\mathcal{R}(h) \leq \widehat{\mathcal{R}}(h) + O \left( \sqrt{\frac{\text{complexity}(\mathcal{F}) + \ln(1/\delta)}{n}} \right),
\]

where "\text{complexity}(\mathcal{F})" is some measure of the complexity of the tightest class of functions \( \mathcal{F} \) considered by the algorithm on this data.
9. Summary of part 2
Summary

- Overfitting arises when we evaluate and train on the same data.
- We can bound error of a fixed function with Hoeffding’s inequality.
- Next lecture we’ll get a version sensitive to function class size.
Part 3...