DDA5001 Machine Learning

Training versus Testing (Part I)

Xiao Li

School of Data Science
The Chinese University of Hong Kong, Shenzhen

Recap: Solution of Least Squares

LS formulation:

$$\min_{\boldsymbol{\theta}} \ \frac{1}{n}\|\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}\|_2^2.$$

- The optimal solution $\widehat{\boldsymbol{\theta}}$ satisfies

$$\boldsymbol{X}^\top \boldsymbol{X}\widehat{\boldsymbol{\theta}} = \boldsymbol{X}^\top \boldsymbol{y}.$$

- Case I: $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ has full column rank, then

$$\widehat{\boldsymbol{\theta}} = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y} = \boldsymbol{X}^\dagger y.$$

- Case II: $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ does not have full column rank. The typical case is, $n < d$. It means overfitting. LS has infinitely many solutions.

What We Have Learned about Supervised Learning

- Components of supervised learning.
- Linear classification and the perceptron algorithm.
- Linear regression and least squares.

Next:

- Let us take linear classification as example. The perceptron algorithm is learned based on the training dataset.
- How much it can say about the test dataset?
- The same question applies to linear regression as well.

$\rightsquigarrow$ This question is about training versus testing (also known as generalization).

Feasibility of Learning

Several Useful Probabilistic Inequalities

Setup for General Training versus Testing

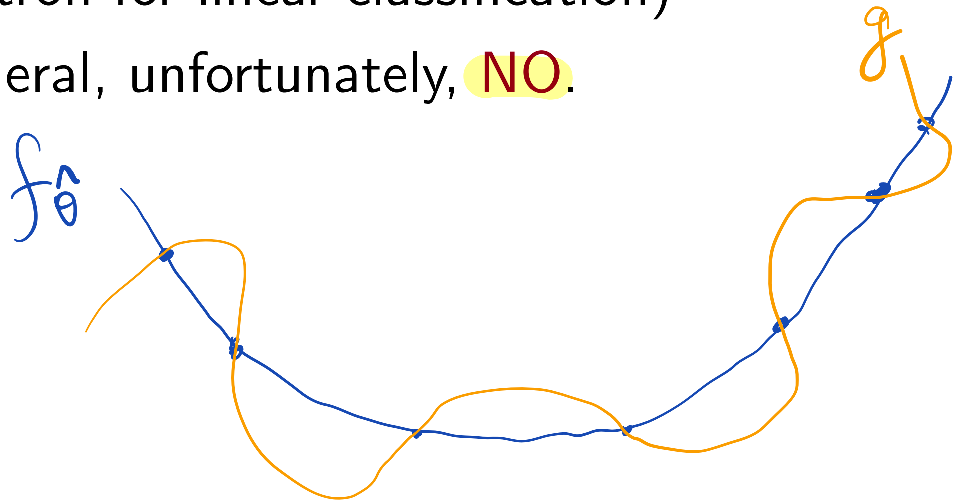
Is Learning Possible?

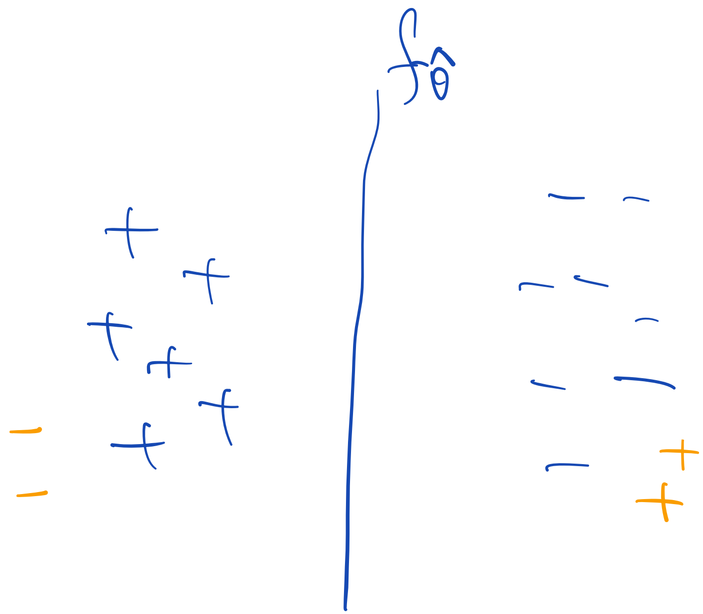
The question is: Does training say anything about testing? (Think about the perceptron for linear classification)

Is Learning Possible?

The question is: Does training say anything about testing? (Think about the perceptron for linear classification)

- In general, unfortunately, NO.





Is Learning Possible?

The question is: Does training say anything about testing? (Think about the perceptron for linear classification)

- In general, unfortunately, NO.
- We can even have adversarial training and testing data, in which training data says nothing (even worse) about testing data.

Train

job	salary	approve ?
3	10	-1
0	0	$+1$

Test

job	salary	approve ?
3	10	$+1$
0	0	-1

Is Learning Possible?

The question is: Does training say anything about testing? (Think about the perceptron for linear classification)

- In general, unfortunately, NO.
- We can even have adversarial training and testing data, in which training data says nothing (even worse) about testing data.

Train

job	salary	approve ?
3	10	-1
0	0	+1

Test

job	salary	approve ?
3	10	+1
0	0	-1

- The (really) unknown target function g is the object of learning. Learning is all about to infer g outside of the seen training dataset.
- There are extreme cases, e.g., we know g on the training dataset, but nothing on other unseen (test) data points. Learning in this case is obviously infeasible.
- Know things we have already seen, this is not learning, it is memorizing.

When and How?

From the previous example, we know that learning is not feasible if we

- make no assumptions about the connections between the training and test data points, and
- want to firmly (deterministically) predict something about the test data.

When and How?

From the previous example, we know that learning is not feasible if we

- make no assumptions about the connections between the training and test data points, and
- want to firmly (deterministically) predict something about the test data.

Fortunately, learning will be possible if we

- make some assumptions that the training and testing data are related in some way. The most common assumption:

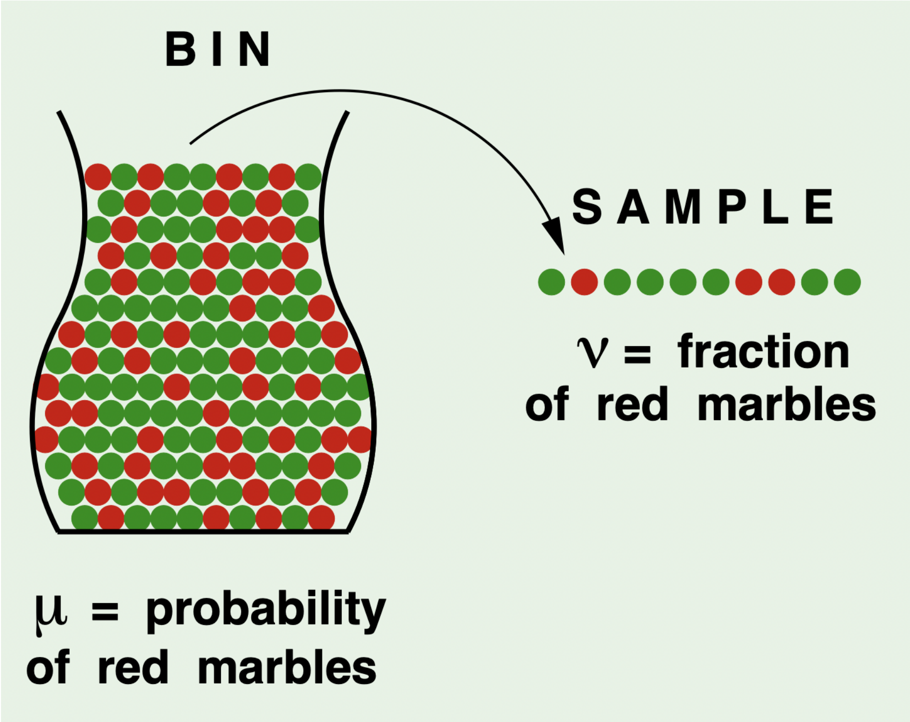
> The training and test data are independent and identically distributed (i.i.d.)

- predict something about the test data in a probabilistic way.

A Bin Sampling Example: i.i.d. and Randomness

- Consider a bin with red and green marbles.
  - $\Pr[\text{red marble}] = \mu$.
  - $\Pr[\text{green marble}] = 1 - \mu$.
- The learning task is to learn μ.
- μ is unknown to us.
- We pick n marbles randomly and independently (with replacement). This means i.i.d.

g



The fraction of red marbles is ν.

$f_{\hat{\theta}}$

A Bin Sampling Example: i.i.d. and Randomness

Does ν say anything about μ?

A Bin Sampling Example: i.i.d. and Randomness

Does ν say anything about μ?

- **No.** Sample can be mostly green while bin is mostly red — just like the previous adversarial example.
- **Yes.** By intuition, sample frequency ν is likely close to the bin frequency μ, once n is not too small.

A Bin Sampling Example: i.i.d. and Randomness

Does ν say anything about μ?

- **No.** Sample can be mostly green while bin is mostly red — just like the previous adversarial example.
- **Yes.** By intuition, sample frequency ν is likely close to the bin frequency μ, once n is not too small.

It is "Possible versus Probable".

- We choose the "probable" characterization to describe feasibility of learning.
- We need several useful probabilistic inequalities to obtain the "probable" results.

Feasibility of Learning

Several Useful Probabilistic Inequalities

Setup for General Training versus Testing

Random Variable

Suppose X is a random variable, how to quantify the behavior of X?

- Probability density function $p(x)$

In principle a random variable X can possibly take any value in $(-\infty, +\infty)$

Random Variable

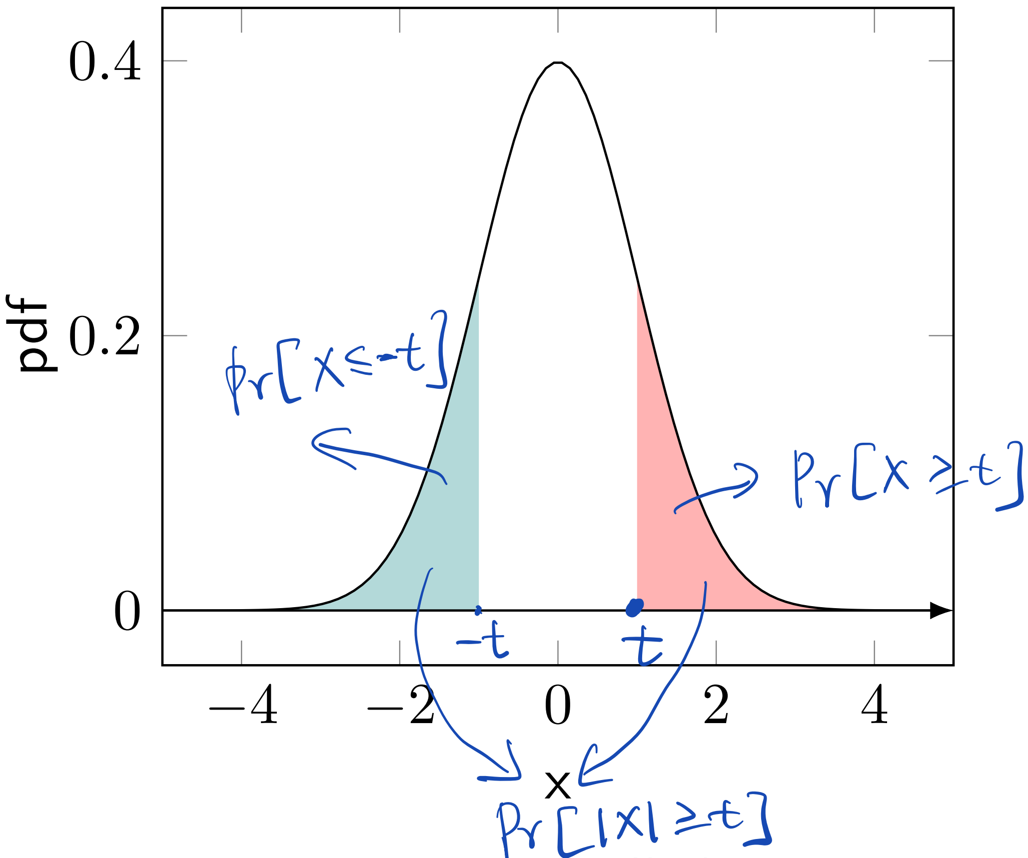
Suppose X is a random variable, how to quantify the behavior of X?

- Probability density function $p(x)$

In principle a random variable X can possibly take any value in $(-\infty, +\infty)$

We can say something like

$$\Pr[X \geq t]$$ → tail probality bound.



Concentration Inequality for Sub-Gaussian

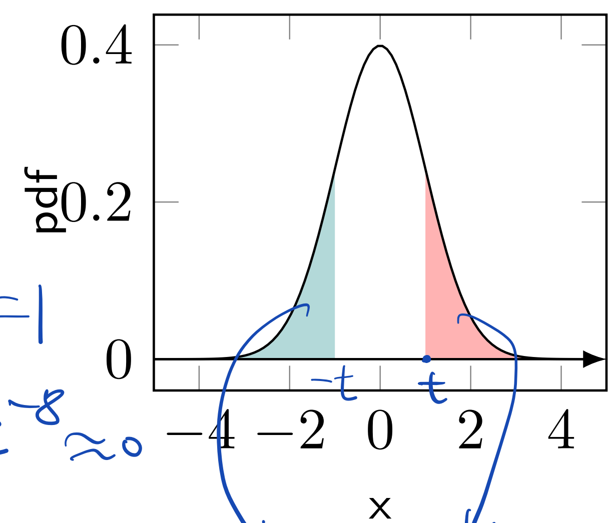
Theorem: Sub-Gaussian Concentration

Suppose X is a sub-Gaussian random variable with mean μ and parameter σ, then for any $t > 0$, we have

$$\boxed{\Pr[|X - \mu| \geq t] \leq 2e^{-\frac{t^2}{2\sigma^2}}}$$

- sub-Gaussian includes standard Gaussian and any bounded random variables. $\to X \in [a, b]$
- Tail probability exponentially decays with respect to t. $\to$ e.g. $t=4$, $\sigma^2=1$
  $\Rightarrow \Pr[\ \ ] \leq 2 \cdot e^{-8} \approx 0$
- Equivalently,

$$\Pr[|X - \mu| \leq t] \geq 1 - 2e^{-\frac{t^2}{2\sigma^2}}$$



$\Pr[|X - \mu| \geq t]$

Hoeffding's Inequality

- Fact: Any bounded random variable on $[a, b]$ are sub-Gaussian random variable with sub-Gaussian parameter $\sigma \leq \frac{b-a}{2}$.
- We can have the following Hoeffding's inequality for bounded random variables.

Corollary: Hoeffding's inequality for bounded random variables

Suppose X_i are independent random variables with mean μ_i and bounded on $[a_i, b_i]$ for $i = 1, \cdots, n$, then for any $t > 0$, we have

$\Rightarrow \sigma_i \leq \frac{b_i - a_i}{2}$

$$\Pr\left[\sum_{i=1}^{n}(X_i - \mu_i) \geq t\right] \leq e^{-\frac{2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}}$$

$\sigma^2 \leq \left(\frac{b-a}{2}\right)^2$

and

$$\Pr\left[\left|\sum_{i=1}^{n}(X_i - \mu_i)\right| \geq t\right] \leq 2e^{-\frac{2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}}$$

Back to Bin Sampling Example

Example: In the bin sampling example, the sampled red marble follows Binomial-(n, μ). What is lower bound of $\Pr\left[|\nu - \mu| \leq t\right]$?

$\nu \downarrow f_{\hat{\theta}}$ $\quad \mu \downarrow g$

red : $X_i = 1 \quad (\mu)$

green : $X_i = 0 \quad (1-\mu)$

$$\# \text{red} = X = \sum_{i=1}^{n} X_i$$

$$\Pr\left[\frac{\left|\overbrace{\sum_{i=1}^{n} X_i}^{X} - n\cdot\mu\right|}{n} \geq \frac{nt}{n}\right] \leq 2e^{-\frac{2(nt)^2}{n}}.$$

$$= 2e^{-2nt^2}$$

$$\Downarrow$$

$$\Pr\left[|\nu - \mu| \geq t\right] \leq 2e^{-2nt^2}$$

Back to Bin Sampling Example

Example: In the bin sampling example, the sampled red marble follows Binomial-(n, μ). What is lower bound of $\Pr[|\nu - \mu| \le t]$?

Suppose that X is the total number of red marbles out of n samples. Each sample is sub-Gaussian with parameter $\sigma \le \frac{b-a}{2} = \frac{1}{2}$ as each sample can either take value 1 (with prob. μ) or take value 0 (with prob. $1-\mu$).

Hence, by Hoeffding's inequality,

$$\Pr[|\nu - \mu| \ge t] = \Pr[|X - \mu n| \ge nt] \le 2e^{-2nt^2}.$$

- Let us replace $n = 500$ and $t = \frac{1}{10}$. We have

$$\Pr\left[|\nu - \mu| \le \frac{1}{10}\right] \ge 1 - 2e^{-10} \approx 1.$$

- That is, "ν learns μ" is probably and approximately correct (P.A.C.).
- Now, we turn to the general learning setting.

Feasibility of Learning

Several Useful Probabilistic Inequalities

Setup for General Training versus Testing

Notations

- $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subseteq \mathcal{X}$ are samples.
- $\{y_1, \ldots, y_n\} \subseteq \mathcal{Y}$ are corresponding labels generated by the target function g.
- $\mathcal{S} = \{\boldsymbol{x}_i\}_{i=1}^n \subseteq \mathcal{X}$ are training samples.
- Binary case: $y_i \in \{-1, +1\}$ and $\mathcal{H} \ni f_{\boldsymbol{\theta}} : \mathcal{X} \to \{-1, +1\}$.
- Example: Perceptron learning algorithm:

$$\operatorname{sign}(f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x}) \to \{-1, +1\}.$$

$\rightsquigarrow$ We will focus on binary linear classification. It can be generalized to real-valued y (i.e., regression) with the same conclusion. However, it is technical, and they do not add to the insight of our analysis of the binary case.

Error Measure

The learning goal (analogous to $\nu \approx \mu$) is

$$\boxed{f_{\boldsymbol{\theta}} \approx g}$$

How to measure such an approximate equality?

- Point-wise error measure:

$$e(f_{\boldsymbol{\theta}}(\boldsymbol{x}), g(\boldsymbol{x})) \text{ is small for all possible data } \boldsymbol{x}$$

- Examples:

$$\text{squared error: } e(f_{\boldsymbol{\theta}}(\boldsymbol{x}), g(\boldsymbol{x})) = (f_{\boldsymbol{\theta}}(\boldsymbol{x}) - g(\boldsymbol{x}))^2$$

$$\text{binary error: } e(f_{\boldsymbol{\theta}}(\boldsymbol{x}), g(\boldsymbol{x})) = 1_{[f_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq g(\boldsymbol{x})]} = \begin{cases} 1, & f_\theta(x) \neq g(x) \\ 0, & \text{o.w.} \end{cases}$$

The squared error measure is mainly used for regression, while the zero-one measure is tailored for classification.

In-sample Error versus Out-of-sample Error

- In-sample Error: Given a set of training samples $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$,

$$\boxed{\mathrm{Er}_{\mathrm{in}} = \frac{1}{n}\sum_{i=1}^{n} e(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \underbrace{g(\boldsymbol{x}_i)}_{y_i})}$$

- Out-of-sample Error: Suppose data $\boldsymbol{x}$ follows a certain distribution $\mathcal{D}$ in an i.i.d. manner,

$$\boxed{\mathrm{Er}_{\mathrm{out}} = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[e(f_{\boldsymbol{\theta}}(\boldsymbol{x}), g(\boldsymbol{x}))\right]}$$

Remarks:

- The In-sample error $\mathrm{Er}_{\mathrm{in}}$ is also known as the training error.
- The out-of-sample error $\mathrm{Er}_{\mathrm{out}}$ is more general than the test error. Fortunately, we can use the test error to approximate $\mathrm{Er}_{\mathrm{out}}$ very well when the test dataset is large enough.

Learning g is to Make $\mathrm{Er}_{\mathrm{out}}$ Small

Recall that learning is all about to infer g outside of the seen training dataset, i.e.,

Make the out-of-sample error small

Final exam analogy

- The in-sample error/training error is the sample final.
- The out-of-sample error/test error is the actual final.
- Goal: do well on actual final.

Memorization vs learning

- Do well on training data by memorizing it (overfitting).
- Learning means you have to do well with new data (generalization).

But, $\mathrm{Er}_{\mathrm{out}}$ is even not computable ;-(

The Concept of Training versus Testing / Generalization

The goal of generalization is to

explore how out-of-sample error is related to in-sample error

The reason to explore this relationship is that $\mathrm{Er}_{\mathrm{in}}$ is computable, checkable, and even amenable.

The Concept of Training versus Testing / Generalization

The goal of generalization is to

explore how out-of-sample error is related to in-sample error

The reason to explore this relationship is that $\mathrm{Er}_{\mathrm{in}}$ is computable, checkable, and even amenable.

Recall that the goal of machine learning is to make the out-of-sample error small.

Expected:

In-sample error is small implies out-of-sample error is small

Then:

It only remains to make the in-sample error small (just choose a very complex hypothesis $\mathcal{H}$)

The Concept of Training versus Testing / Generalization

The goal of generalization is to

explore how out-of-sample error is related to in-sample error

The reason to explore this relationship is that $\mathrm{Er}_{\mathrm{in}}$ is computable, checkable, and even amenable.

Recall that the goal of machine learning is to make the out-of-sample error small.

Expected:

In-sample error is small implies out-of-sample error is small

Then:

It only remains to make the in-sample error small (just choose a very complex hypothesis $\mathcal{H}$)

However, this intuition is not true in general.

The Fundamental Trade-off in Learning

Here is a simple decomposition:

$$\mathrm{Er}_{\mathrm{out}} = \underbrace{\mathrm{Er}_{\mathrm{out}} - \mathrm{Er}_{\mathrm{in}}}_{generalization\ error} + \underbrace{\mathrm{Er}_{\mathrm{in}}}_{traning\ error}$$

Simple observation is that we have to simultaneously make generalization error and training error small, in order to make $\mathrm{Er}_{\mathrm{out}}$ small.

The Fundamental Trade-off in Learning

Here is a simple decomposition: Training versus testing

$$\mathrm{Er}_{\mathrm{out}} = \underbrace{\mathrm{Er}_{\mathrm{out}} - \mathrm{Er}_{\mathrm{in}}}_{generalization\ error} + \underbrace{\mathrm{Er}_{\mathrm{in}}}_{traning\ error}$$

Simple observation is that we have to simultaneously make generalization error and training error small, in order to make $\mathrm{Er}_{\mathrm{out}}$ small.

On the generalization side, we need:

less complex hypothesis $\mathcal{H}$

On the training side, we need:

more complex hypothesis $\mathcal{H}$

$\rightsquigarrow$ We will study the above conclusions later.

i.i.d. Assumption

We make the following assumption to connect training and test date:

- All data (training + testing) come from the same distribution $\mathcal{D}$ (identically distributed).
- The data are sampled independently.

i.i.d. interpretation

	age	gender	salary	citizenship	years in job
applicant 1 (train)	2.5	1	10	3	1
applicant 2 (train)	2.8	0	8	6	5
applicant 3 (train)	1.6	0	0	4	0
applicant 4 (test)	1.5	1	4	5	3

- Rows 1-4 follow the same same distribution $\mathcal{D}$.
- Rows 1-4 are mutually independent.

i.i.d. is an ideal assumption, but a good approximation of practice.

The Starting Point

Given training samples $\mathcal{S} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$.

In expectation for fixed $f \in \mathcal{H}$: (we omit $\boldsymbol{\theta}$ in $f_{\boldsymbol{\theta}}$ for simplicity)

$$\boxed{\mathbb{E}_{\mathcal{S}\sim_{i.i.d.}\mathcal{D}}\left[\mathrm{Er}_{\mathrm{in}}(f)\right] = \mathrm{Er}_{\mathrm{out}}(f).}$$

Proof: Recall that all samples are i.i.d. according to $\mathcal{D}$, we have

$$\mathbb{E}_{\mathcal{S}\sim_{i.i.d.}\mathcal{D}}\left[\mathrm{Er}_{\mathrm{in}}(f)\right] = \mathbb{E}_{\mathcal{S}\sim_{i.i.d.}\mathcal{D}}\left[\frac{1}{n}\sum_{i=1}^{n} e(f(\boldsymbol{x}_i), g(\boldsymbol{x}_i))\right]$$

$$\mathbb{E}_{x_i\sim D}\, e(f(x_i), g(x_i)) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[e(f(\boldsymbol{x}), g(\boldsymbol{x}))\right]}_{\mathrm{Erout}(f)} \quad \text{(since i.i.d.)}$$

$$= \mathbb{E}_{x\sim D}\, e(f(x), g(x)). = \mathrm{Er}_{\mathrm{out}}(f)$$

Interpretation

$$\boxed{\mathbb{E}_{\mathcal{S}\sim_{i.i.d.}\mathcal{D}}\left[\mathrm{Er}_{\mathrm{in}}(f)\right] = \mathrm{Er}_{\mathrm{out}}(f).}$$

- $\mathrm{Er}_{\mathrm{in}}(f)$ is an unbiased estimator for $\mathrm{Er}_{\mathrm{out}}(f)$.
- Law of large numbers: When $n \to \infty$, we have in-sample error estimates the out-of-sample error accurately.
- However, this is an asymptotic (infinite n) result. In practice, we can never deal with infinite samples.

Interpretation

$$\boxed{\mathbb{E}_{\mathcal{S}\sim_{i.i.d.}\mathcal{D}}\left[\mathrm{Er}_{\mathrm{in}}(f)\right] = \mathrm{Er}_{\mathrm{out}}(f).}$$

- $\mathrm{Er}_{\mathrm{in}}(f)$ is an unbiased estimator for $\mathrm{Er}_{\mathrm{out}}(f)$.
- Law of large numbers: When $n \to \infty$, we have in-sample error estimates the out-of-sample error accurately.
- However, this is an asymptotic (infinite n) result. In practice, we can never deal with infinite samples.

Solution: Derive non-asymptotic (finite n) results using concentration inequalities. $\rightsquigarrow$Next lecture.