YSU Statistical ML, Fall 2019

Lecture 01

Michael Poghosyan
YSU, AUA
michael@ysu.am, mpoghosyan@aua.am

12 November 2019

Supervised Learning

Contents

Supervised Learning

Everything starts from Data.

Supervised Learning

Everything starts from Data.

Here we assume we are given:

Supervised Learning

Everything starts from Data.

Here we assume we are given:

- The Input Space $\mathcal{X}$

Supervised Learning

Everything starts from Data.

Here we assume we are given:

- The Input Space $\mathcal{X}$
- The Output Space $\mathcal{Y}$

Supervised Learning

Everything starts from Data.

Here we assume we are given:

- The Input Space $\mathcal{X}$
- The Output Space $\mathcal{Y}$

We will assume $\mathcal{X} \subset \mathbb{R}^d$ (or, maybe, in other d-Dim Space), and a typical element $\mathbf{x}$ of $\mathcal{X}$ will have the form

$$\mathbf{x} = (x_1, x_2, ..., x_d)$$

Supervised Learning

Everything starts from Data.

Here we assume we are given:

- The Input Space $\mathcal{X}$
- The Output Space $\mathcal{Y}$

We will assume $\mathcal{X} \subset \mathbb{R}^d$ (or, maybe, in other d-Dim Space), and a typical element $\mathbf{x}$ of $\mathcal{X}$ will have the form

$$\mathbf{x} = (x_1, x_2, ..., x_d)$$

We will call x_k-s to be the **Features** of $\mathbf{x}$.

Supervised Learning

Everything starts from Data.

Here we assume we are given:

- The Input Space $\mathcal{X}$
- The Output Space $\mathcal{Y}$

We will assume $\mathcal{X} \subset \mathbb{R}^d$ (or, maybe, in other d-Dim Space), and a typical element $\mathbf{x}$ of $\mathcal{X}$ will have the form

$$\mathbf{x} = (x_1, x_2, ..., x_d)$$

We will call x_k-s to be the **Features** of $\mathbf{x}$.

We will assume also that $\mathcal{Y} \subset \mathbb{R}$, and we will call the elements of $\mathcal{Y}$ to be the **Labels**.

Supervised Learning

In the Supervised Learning Problems, we have a Data of the form:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n).$$

Supervised Learning

In the Supervised Learning Problems, we have a Data of the form:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n).$$

Here we interpret $\mathbf{x}_k$ and y_k as the Feature vector and the Label of the Observation (Object) k.

Supervised Learning

In the Supervised Learning Problems, we have a Data of the form:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n).$$

Here we interpret $\mathbf{x}_k$ and y_k as the Feature vector and the Label of the Observation (Object) k.

So we know the labels of our n Observations.

Supervised Learning

In the Supervised Learning Problems, we have a Data of the form:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n).$$

Here we interpret $\mathbf{x}_k$ and y_k as the Feature vector and the Label of the Observation (Object) k.

So we know the labels of our n Observations.

Problem: Given a Feature vector $\mathbf{x}$, other than $\mathbf{x}_k$, predict its Label y.

Features and Labels

Recall that our Feature Vector $\mathbf{x} \in \mathcal{X}$ has the form:

$$\mathbf{x} = (x_1, x_2, ..., x_d).$$

Features and Labels

Recall that our Feature Vector $\mathbf{x} \in \mathcal{X}$ has the form:

$$\mathbf{x} = (x_1, x_2, ..., x_d).$$

The Features x_k can be:

Features and Labels

Recall that our Feature Vector $\mathbf{x} \in \mathcal{X}$ has the form:

$$\mathbf{x} = (x_1, x_2, ..., x_d).$$

The Features x_k can be:

- Binary, if $x_k \in \{0, 1\}$ or $x_k \in \{-1, 1\}$

Features and Labels

Recall that our Feature Vector $\mathbf{x} \in \mathcal{X}$ has the form:

$$\mathbf{x} = (x_1, x_2, ..., x_d).$$

The Features x_k can be:

- Binary, if $x_k \in \{0, 1\}$ or $x_k \in \{-1, 1\}$
- Nominal/Categorical, if the set of possible values of x_k is finite, and no intrinsic order exists in that set

Features and Labels

Recall that our Feature Vector $\mathbf{x} \in \mathcal{X}$ has the form:

$$\mathbf{x} = (x_1, x_2, ..., x_d).$$

The Features x_k can be:

- Binary, if $x_k \in \{0, 1\}$ or $x_k \in \{-1, 1\}$
- Nominal/Categorical, if the set of possible values of x_k is finite, and no intrinsic order exists in that set
- Ordinal, if the set of possible values of x_k is finite, and there is a natural order in that set

Features and Labels

Recall that our Feature Vector $\mathbf{x} \in \mathcal{X}$ has the form:

$$\mathbf{x} = (x_1, x_2, ..., x_d).$$

The Features x_k can be:

- Binary, if $x_k \in \{0, 1\}$ or $x_k \in \{-1, 1\}$
- Nominal/Categorical, if the set of possible values of x_k is finite, and no intrinsic order exists in that set
- Ordinal, if the set of possible values of x_k is finite, and there is a natural order in that set
- Numerical/Quantitative, if $x_k \in \mathbb{R}$

Features and Labels

The Labels can be:

Features and Labels

The Labels can be:

Classification Problems:

- $\mathcal{Y} = \{-1, 1\}$ or $\mathcal{Y} = \{0, 1\}$ - **Binary Classification**

Features and Labels

The Labels can be:

Classification Problems:

- $\mathcal{Y} = \{-1, 1\}$ or $\mathcal{Y} = \{0, 1\}$ - **Binary Classification**
- $\mathcal{Y} = \{1, 2, ..., K\}$ - K-**class Classification**

Features and Labels

The Labels can be:

Classification Problems:

- $\mathcal{Y} = \{-1, 1\}$ or $\mathcal{Y} = \{0, 1\}$ - **Binary Classification**
- $\mathcal{Y} = \{1, 2, ..., K\}$ - K-**class Classification**

Regression Problems:

- $\mathcal{Y} = \mathbb{R}$ - **1D Regression**

Features and Labels

The Labels can be:

Classification Problems:

- $\mathcal{Y} = \{-1, 1\}$ or $\mathcal{Y} = \{0, 1\}$ - **Binary Classification**
- $\mathcal{Y} = \{1, 2, ..., K\}$ - K**-class Classification**

Regression Problems:

- $\mathcal{Y} = \mathbb{R}$ - **1D Regression**

Ranking Problems:

- $\mathcal{Y}$ is a finite ordered set

Examples

See Vorontsov's Lecture Slides

Supervised Learning

So, having a Dataset of Observations with Labels, we want to predict the Label for a new Observation. Of course, we cannot do this unless we will assume there is some structure, some relationship in the Data.

Supervised Learning

So, having a Dataset of Observations with Labels, we want to predict the Label for a new Observation. Of course, we cannot do this unless we will assume there is some structure, some relationship in the Data. Mathematically, we will assume that behind our Data we have a Probability Distribution, generating our Data.

Supervised Learning

So, having a Dataset of Observations with Labels, we want to predict the Label for a new Observation. Of course, we cannot do this unless we will assume there is some structure, some relationship in the Data. Mathematically, we will assume that behind our Data we have a Probability Distribution, generating our Data.

We will assume that the observation $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ is a realization of the pair of r.v.s $(\mathbf{X}, Y)$ that is coming from some unknown Distribution $\mathcal{F}$:

$$(\mathbf{X}, Y) \sim \mathcal{F}.$$

Supervised Learning

So, having a Dataset of Observations with Labels, we want to predict the Label for a new Observation. Of course, we cannot do this unless we will assume there is some structure, some relationship in the Data. Mathematically, we will assume that behind our Data we have a Probability Distribution, generating our Data.

We will assume that the observation $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ is a realization of the pair of r.v.s $(\mathbf{X}, Y)$ that is coming from some unknown Distribution $\mathcal{F}$:

$$(\mathbf{X}, Y) \sim \mathcal{F}.$$

So we "encode" our Data $(\mathbf{x}_k, y_k)$ as being a realisation of a r.v. $(\mathbf{X}_k, Y_k)$.

Supervised Learning

So, having a Dataset of Observations with Labels, we want to predict the Label for a new Observation. Of course, we cannot do this unless we will assume there is some structure, some relationship in the Data. Mathematically, we will assume that behind our Data we have a Probability Distribution, generating our Data.

We will assume that the observation $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ is a realization of the pair of r.v.s $(\mathbf{X}, Y)$ that is coming from some unknown Distribution $\mathcal{F}$:

$$(\mathbf{X}, Y) \sim \mathcal{F}.$$

So we "encode" our Data $(\mathbf{x}_k, y_k)$ as being a realisation of a r.v. $(\mathbf{X}_k, Y_k)$.

The general idea/Problem is, having Data, to infer $\mathcal{F}$.

Supervised Learning

This is a very general problem, so we consider the following:

Supervised Learning

This is a very general problem, so we consider the following: Given a sequence of IID r.v.s

$$(\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), ..., (\mathbf{X}_n, Y_n)$$

construct a "good" Prediction Function

$$g : \mathcal{X} \to \mathcal{Y},$$

that will predict the Label of $\mathbf{X}$.

Supervised Learning, Loss Function

Here we need to talk about different things:

Supervised Learning, Loss Function

Here we need to talk about different things:

- What is the meaning that g is giving "good" labels, is a "good" Predictor, how to assess that?

Supervised Learning, Loss Function

Here we need to talk about different things:

- What is the meaning that g is giving "good" labels, is a "good" Predictor, how to assess that?
- How to construct good Predictors?

Supervised Learning, Loss Function

Here we need to talk about different things:

- What is the meaning that g is giving "good" labels, is a "good" Predictor, how to assess that?
- How to construct good Predictors?

Construction of g, using the Data we have, is called **Training**.

Supervised Learning, Loss Function

Here we need to talk about different things:

- What is the meaning that g is giving "good" labels, is a "good" Predictor, how to assess that?
- How to construct good Predictors?

Construction of g, using the Data we have, is called **Training**.
Predicting the values for new observations is called **Testing**.

Supervised Learning, Loss Function

Here we need to talk about different things:

- What is the meaning that g is giving "good" labels, is a "good" Predictor, how to assess that?
- How to construct good Predictors?

Construction of g, using the Data we have, is called **Training**. Predicting the values for new observations is called **Testing**.

To assess goodness of the Predictor g, we take a **Loss** function.

Supervised Learning, Loss Function

Here we need to talk about different things:

- What is the meaning that g is giving "good" labels, is a "good" Predictor, how to assess that?
- How to construct good Predictors?

Construction of g, using the Data we have, is called **Training**. Predicting the values for new observations is called **Testing**.

To assess goodness of the Predictor g, we take a **Loss** function. We will call any function of the form

$$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$

a **Loss** function, and we will assume that:

$$\ell(y_1, y_2) \geq 0, \qquad \forall y_1, y_2 \in \mathcal{Y}, \qquad \textit{and} \qquad \ell(y, y) = 0.$$

Loss Function

Some known Loss Functions are:

Loss Function

Some known Loss Functions are:

- For The Binary Classification:

$$\ell(y_1, y_2) = \begin{cases} 1, & y_1 \neq y_2 \\ 0, & y_1 = y_2 \end{cases}$$

[1] Here we use the Indicator, $\mathbf{1}(x)$ function with $\mathbf{1}(\mathit{True}) = 1$ and $\mathbf{1}(\mathit{False}) = 0$.

Loss Function

Some known Loss Functions are:

- For The Binary Classification:

$$\ell(y_1, y_2) = \begin{cases} 1, & y_1 \neq y_2 \\ 0, & y_1 = y_2 \end{cases}$$

We denote this[1] as $\mathbf{1}(y_1 \neq y_2)$.

[1] Here we use the Indicator, $\mathbf{1}(x)$ function with $\mathbf{1}(\mathit{True}) = 1$ and $\mathbf{1}(\mathit{False}) = 0$.

Loss Function

Some known Loss Functions are:

- For The Binary Classification:

$$\ell(y_1, y_2) = \begin{cases} 1, & y_1 \neq y_2 \\ 0, & y_1 = y_2 \end{cases}$$

  We denote this[1] as $\mathbf{1}(y_1 \neq y_2)$.

- For 1D Regression:
  - $\ell(y_1, y_2) = (y_1 - y_2)^2$ - Quadratic Loss;

[1] Here we use the Indicator, $\mathbf{1}(x)$ function with $\mathbf{1}(\mathit{True}) = 1$ and $\mathbf{1}(\mathit{False}) = 0$.

Loss Function

Some known Loss Functions are:

- For The Binary Classification:

$$\ell(y_1, y_2) = \begin{cases} 1, & y_1 \neq y_2 \\ 0, & y_1 = y_2 \end{cases}$$

  We denote this[1] as $\mathbf{1}(y_1 \neq y_2)$.

- For 1D Regression:
  - $\ell(y_1, y_2) = (y_1 - y_2)^2$ - Quadratic Loss;
  - $\ell(y_1, y_2) = |y_1 - y_2|$ - Absolute Error Loss;

[1] Here we use the Indicator, $\mathbf{1}(x)$ function with $\mathbf{1}(\mathit{True}) = 1$ and $\mathbf{1}(\mathit{False}) = 0$.

Learning Problem

Now assume we have a Predictor $g : \mathcal{X} \to \mathcal{Y}$, and a Loss Function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

Learning Problem

Now assume we have a Predictor $g : \mathcal{X} \to \mathcal{Y}$, and a Loss Function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

Assume $(\mathbf{X}, Y) \sim \mathcal{F}$.

Learning Problem

Now assume we have a Predictor $g : \mathcal{X} \to \mathcal{Y}$, and a Loss Function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

Assume $(\mathbf{X}, Y) \sim \mathcal{F}$. So $\mathbf{X}$ is our Feature Vector, and we Predict its label as $g(\mathbf{X})$.

Learning Problem

Now assume we have a Predictor $g : \mathcal{X} \to \mathcal{Y}$, and a Loss Function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

Assume $(\mathbf{X}, Y) \sim \mathcal{F}$. So $\mathbf{X}$ is our Feature Vector, and we Predict its label as $g(\mathbf{X})$. Then the Loss incurring will be

$$\ell(Y, g(\mathbf{X})).$$

Learning Problem

Now assume we have a Predictor $g : \mathcal{X} \to \mathcal{Y}$, and a Loss Function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

Assume $(\mathbf{X}, Y) \sim \mathcal{F}$. So $\mathbf{X}$ is our Feature Vector, and we Predict its label as $g(\mathbf{X})$. Then the Loss incurring will be

$$\ell(Y, g(\mathbf{X})).$$

We want to have this Loss as small as possible.

Learning Problem

Now assume we have a Predictor $g : \mathcal{X} \to \mathcal{Y}$, and a Loss Function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

Assume $(\mathbf{X}, Y) \sim \mathcal{F}$. So $\mathbf{X}$ is our Feature Vector, and we Predict its label as $g(\mathbf{X})$. Then the Loss incurring will be

$$\ell(Y, g(\mathbf{X})).$$

We want to have this Loss as small as possible. Well, under our setting, this will be a R.V., so we define the **Average Loss** or the **Risk** of the Predictor g to be

$$Risk(g) = \mathbb{E}\Big(\ell(Y, g(\mathbf{X}))\Big).$$

Learning Problem

Now assume we have a Predictor $g : \mathcal{X} \to \mathcal{Y}$, and a Loss Function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

Assume $(\mathbf{X}, Y) \sim \mathcal{F}$. So $\mathbf{X}$ is our Feature Vector, and we Predict its label as $g(\mathbf{X})$. Then the Loss incurring will be

$$\ell(Y, g(\mathbf{X})).$$

We want to have this Loss as small as possible. Well, under our setting, this will be a R.V., so we define the **Average Loss** or the **Risk** of the Predictor g to be

$$Risk(g) = \mathbb{E}\Big(\ell(Y, g(\mathbf{X}))\Big).$$

Here the Expectation is over the Distribution of $(\mathbf{X}, Y)$, i.e., $\mathcal{F}$.

Learning Problem

Now assume we have a Predictor $g : \mathcal{X} \to \mathcal{Y}$, and a Loss Function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

Assume $(\mathbf{X}, Y) \sim \mathcal{F}$. So $\mathbf{X}$ is our Feature Vector, and we Predict its label as $g(\mathbf{X})$. Then the Loss incurring will be

$$\ell(Y, g(\mathbf{X})).$$

We want to have this Loss as small as possible. Well, under our setting, this will be a R.V., so we define the **Average Loss** or the **Risk** of the Predictor g to be

$$Risk(g) = \mathbb{E}\Big(\ell(Y, g(\mathbf{X}))\Big).$$

Here the Expectation is over the Distribution of $(\mathbf{X}, Y)$, i.e., $\mathcal{F}$.

Now, we can state our Problem of finding a good Predictor: Find g minimizing the Risk, i.e., find

$$g^* \in \underset{g}{argmin}\, Risk(g).$$

Example

Toy Example: Assume $\mathcal{X} = \{1, 2, 3\}$, $\mathcal{Y} = \{0, 1\}$, and we have the Joint Distribution of (X, Y):

$Y \setminus X$	1	2	3
0	0.1	0.2	0.1
1	0.2	0.1	0.3

Assume

$$g(x) = \begin{cases} 0, & \text{if } x \text{ is even} \\ 1, & \text{otherwise} \end{cases}$$

and $\ell(y_1, y_2) = |y_1 - y_2|$. Calculate the $Risk(g)$.

Solution: OTB

Predictors

The above Minimization Problem is not complete: we need to specify the set of all possible Predictors g.

Predictors

The above Minimization Problem is not complete: we need to specify the set of all possible Predictors g.

The best case will be if we will take g to be **any measurable function from $\mathcal{X}$ to $\mathcal{Y}$**.

Predictors

The above Minimization Problem is not complete: we need to specify the set of all possible Predictors g.

The best case will be if we will take g to be **any measurable function from $\mathcal{X}$ to $\mathcal{Y}$**. But, unfortunately, this set is veery large to be able to solve the problem there.

Predictors

The above Minimization Problem is not complete: we need to specify the set of all possible Predictors g.

The best case will be if we will take g to be **any measurable function from $\mathcal{X}$ to $\mathcal{Y}$**. But, unfortunately, this set is veery large to be able to solve the problem there.

Usually, we assume that g comes from a Parametric Family of functions, which we call a Predictive Model:

$$g \in \mathcal{G} = \{g(\mathbf{x}|\theta),\ \theta \in \Theta\}, \qquad \textit{where} \quad g(\mathbf{x}|\theta) : \mathcal{X} \to \mathcal{Y}.$$

and Θ is some Parameter Set (1D or more).

Predictors, Examples

Say,

- In the 1D Linear Regression Problem, we consider

$$g(\mathbf{x}|\theta) = \theta_0 + \theta_1 \cdot x_1 + ... + \theta_d \cdot x_d;$$

Predictors, Examples

Say,

- In the 1D Linear Regression Problem, we consider

$$g(\mathbf{x}|\theta) = \theta_0 + \theta_1 \cdot x_1 + ... + \theta_d \cdot x_d;$$

- In the Binary Classification Problem, with $\mathcal{Y} = \{-1, 1\}$, we can consider, say

$$g(\mathbf{x}|\theta) = sgn(\theta_0 + \theta_1 \cdot x_1 + ... + \theta_d \cdot x_d);$$

Predictors, Examples

Say,

- In the 1D Linear Regression Problem, we consider

$$g(\mathbf{x}|\theta) = \theta_0 + \theta_1 \cdot x_1 + ... + \theta_d \cdot x_d;$$

- In the Binary Classification Problem, with $\mathcal{Y} = \{-1, 1\}$, we can consider, say

$$g(\mathbf{x}|\theta) = sgn(\theta_0 + \theta_1 \cdot x_1 + ... + \theta_d \cdot x_d);$$

- In the general Regression/Classification Problems, we can have $g(\mathbf{x}|\theta)$ to be a Neural Network, where $\mathbf{x}$ is our input, θ is the vector of all NN weights, and $g(\mathbf{x}|\theta)$ is the output of the NN.

The Learning Problem

Now we can finalize the statement of our Problem:

The Learning Problem

Now we can finalize the statement of our Problem: We are given

- A Dataset of Observations $(\mathbf{x}_k, y_k)$, $k = 1, .., n$, coming as a realization of $(\mathbf{X}_k, Y_k)$ from an unknown Distribution $\mathcal{F}$;

The Learning Problem

Now we can finalize the statement of our Problem: We are given

- A Dataset of Observations $(\mathbf{x}_k, y_k)$, $k = 1, .., n$, coming as a realization of $(\mathbf{X}_k, Y_k)$ from an unknown Distribution $\mathcal{F}$;
- A Loss Function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$;

The Learning Problem

Now we can finalize the statement of our Problem: We are given

- A Dataset of Observations $(\mathbf{x}_k, y_k)$, $k = 1, .., n$, coming as a realization of $(\mathbf{X}_k, Y_k)$ from an unknown Distribution $\mathcal{F}$;
- A Loss Function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$;
- A Predictive Model (set of Functions) $\mathcal{G}$

The Learning Problem

Now we can finalize the statement of our Problem: We are given

- A Dataset of Observations $(\mathbf{x}_k, y_k)$, $k = 1, .., n$, coming as a realization of $(\mathbf{X}_k, Y_k)$ from an unknown Distribution $\mathcal{F}$;
- A Loss Function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$;
- A Predictive Model (set of Functions) $\mathcal{G}$

and we want to find $g^* \in \mathcal{G}$ such that

$$g^* \in \underset{g \in \mathcal{G}}{argmin}\, Risk(g).$$

The Learning Problem

Now we can finalize the statement of our Problem: We are given

- A Dataset of Observations $(\mathbf{x}_k, y_k)$, $k = 1, .., n$, coming as a realization of $(\mathbf{X}_k, Y_k)$ from an unknown Distribution $\mathcal{F}$;
- A Loss Function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$;
- A Predictive Model (set of Functions) $\mathcal{G}$

and we want to find $g^* \in \mathcal{G}$ such that

$$g^* \in \underset{g \in \mathcal{G}}{argmin}\, Risk(g).$$

If $\mathcal{G}$ coincides with the set of all measurable functions, thet g^*, if exists, is called the **Bayes Predictor**.

Risk Minimization vs Empirical Risk Minimization

OK, nice. But, unfortunately, we usually cannot solve the above problem, since we **do not have the Distribution** $\mathcal{F}$ to calculate the Risk.

Risk Minimization vs Empirical Risk Minimization

OK, nice. But, unfortunately, we usually cannot solve the above problem, since we **do not have the Distribution** $\mathcal{F}$ to calculate the Risk.

So we do the following: recall that, by the LLN,

$$\frac{1}{n} \cdot \sum_{k=1}^{n} \ell(Y_k, g(\mathbf{X}_k)) \to$$

Risk Minimization vs Empirical Risk Minimization

OK, nice. But, unfortunately, we usually cannot solve the above problem, since we **do not have the Distribution** $\mathcal{F}$ to calculate the Risk.

So we do the following: recall that, by the LLN,

$$\frac{1}{n} \cdot \sum_{k=1}^{n} \ell(Y_k, g(\mathbf{X}_k)) \to \mathbb{E}\Big(\ell(Y, g(\mathbf{X}))\Big) = Risk(g) \qquad a.s.$$

Risk Minimization vs Empirical Risk Minimization

OK, nice. But, unfortunately, we usually cannot solve the above problem, since we **do not have the Distribution $\mathcal{F}$** to calculate the Risk.

So we do the following: recall that, by the LLN,

$$\frac{1}{n} \cdot \sum_{k=1}^{n} \ell(Y_k, g(\mathbf{X}_k)) \to \mathbb{E}\Big(\ell(Y, g(\mathbf{X}))\Big) = Risk(g) \qquad a.s.$$

So, instead of trying to minimize $Risk(g)$, we can try to minimize

$$ERM(g) = \frac{1}{n} \cdot \sum_{k=1}^{n} \ell(Y_k, g(\mathbf{X}_k)),$$

which is called the **Empirical Risk Measure of** g.

The Learning Problem, Empirical Version

Now we change the statement of our Problem like this:

The Learning Problem, Empirical Version

Now we change the statement of our Problem like this: We are given

- A Dataset of Observations $(\mathbf{x}_k, y_k)$, $k = 1, .., n$, coming as a realization of $(\mathbf{X}_k, Y_k)$ from an unknown Distribution $\mathcal{F}$;
- A Loss Function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$;
- A Predictive Model (set of Functions) $\mathcal{G}$

The Learning Problem, Empirical Version

Now we change the statement of our Problem like this: We are given

- A Dataset of Observations $(\mathbf{x}_k, y_k)$, $k = 1, .., n$, coming as a realization of $(\mathbf{X}_k, Y_k)$ from an unknown Distribution $\mathcal{F}$;
- A Loss Function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$;
- A Predictive Model (set of Functions) $\mathcal{G}$

and we want to find $g^* \in \mathcal{G}$ such that

$$g^* \in \underset{g \in \mathcal{G}}{argmin}\, ERM(g).$$