FIRST-ORDER OPTIMIZATION FOR SUPERQUANTILE-BASED SUPERVISED LEARNING

MACHINE LEARNING FOR SIGNAL PROCESSING - 2020

Yassine LAGUEL[★] — Joint work with J. Malick[▲] and Z. Harchaoui[◆]

[★]Université Grenoble Alpes - [▲]CNRS - [◆]University of Washington

1 Safety in supervised ML

2 THE TOOLBOX SPQR

3 BEHIND SPQR

Supervised Learning

- Classical Supervised Machine Learning

$$\min_{w\in\mathbb{R}^d} \mathbb{E}_{(x,y)\sim\mathcal{P}}\left[\ell(y, \varphi(w, x))\right]$$

Supervised Learning

- Classical Supervised Machine Learning

$$\min_{w \in \mathbb{R}^d} \mathbb{E}_{(x,y)\sim\mathcal{P}} [\ell(y, \varphi(w, x))]$$

Model

Distribution

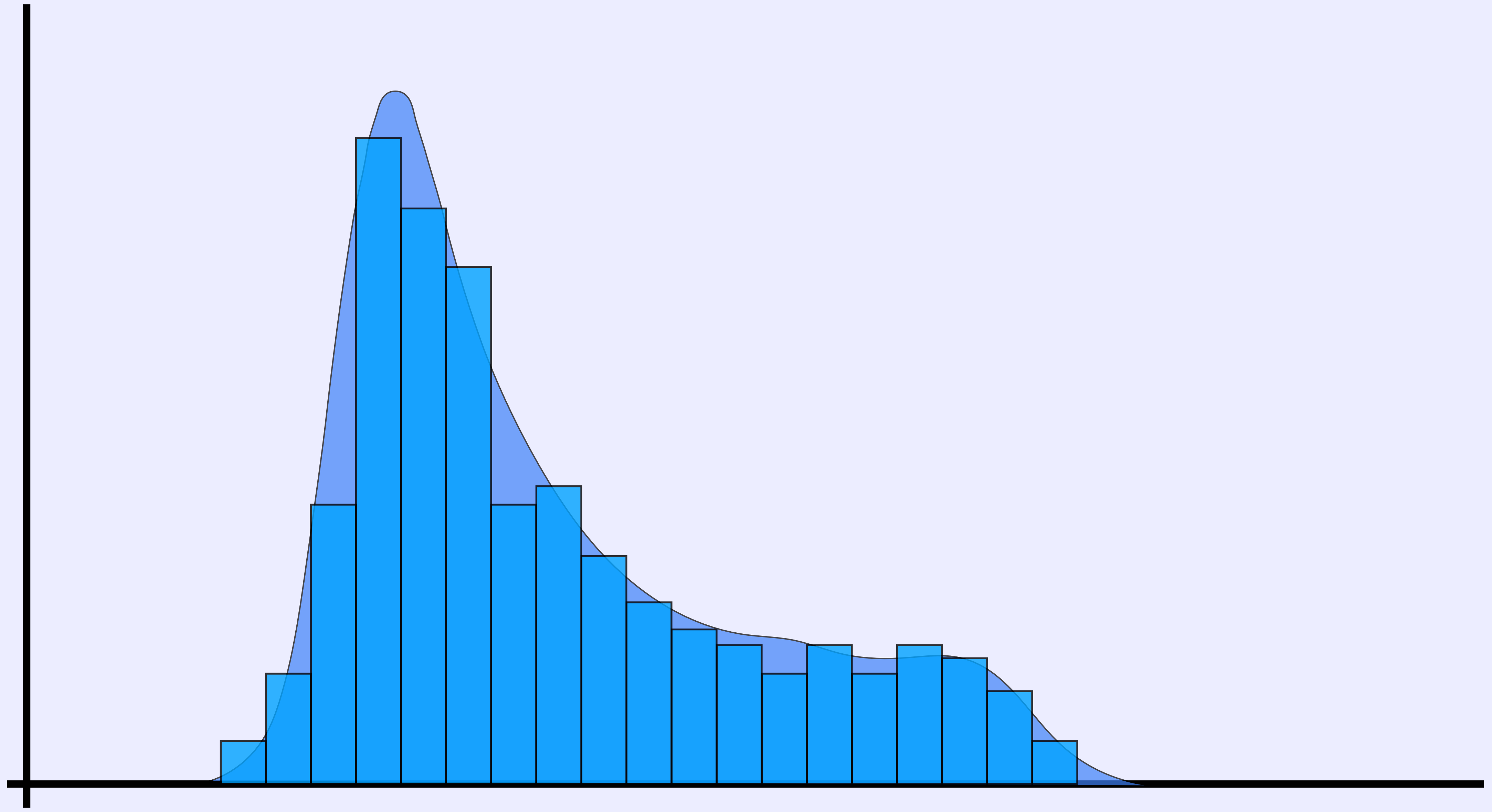
Loss

Prediction function

Supervised Learning

- Classical Supervised Machine Learning $(x_1, y_1), \ldots, (x_n, y_n) \sim \mathcal{P}$ Training Distribution

$$\min_{w \in \mathbb{R}^d} \mathbb{E}_{(x,y)\sim\mathcal{P}} \left[\ell(y, \varphi(w, x))\right]$$



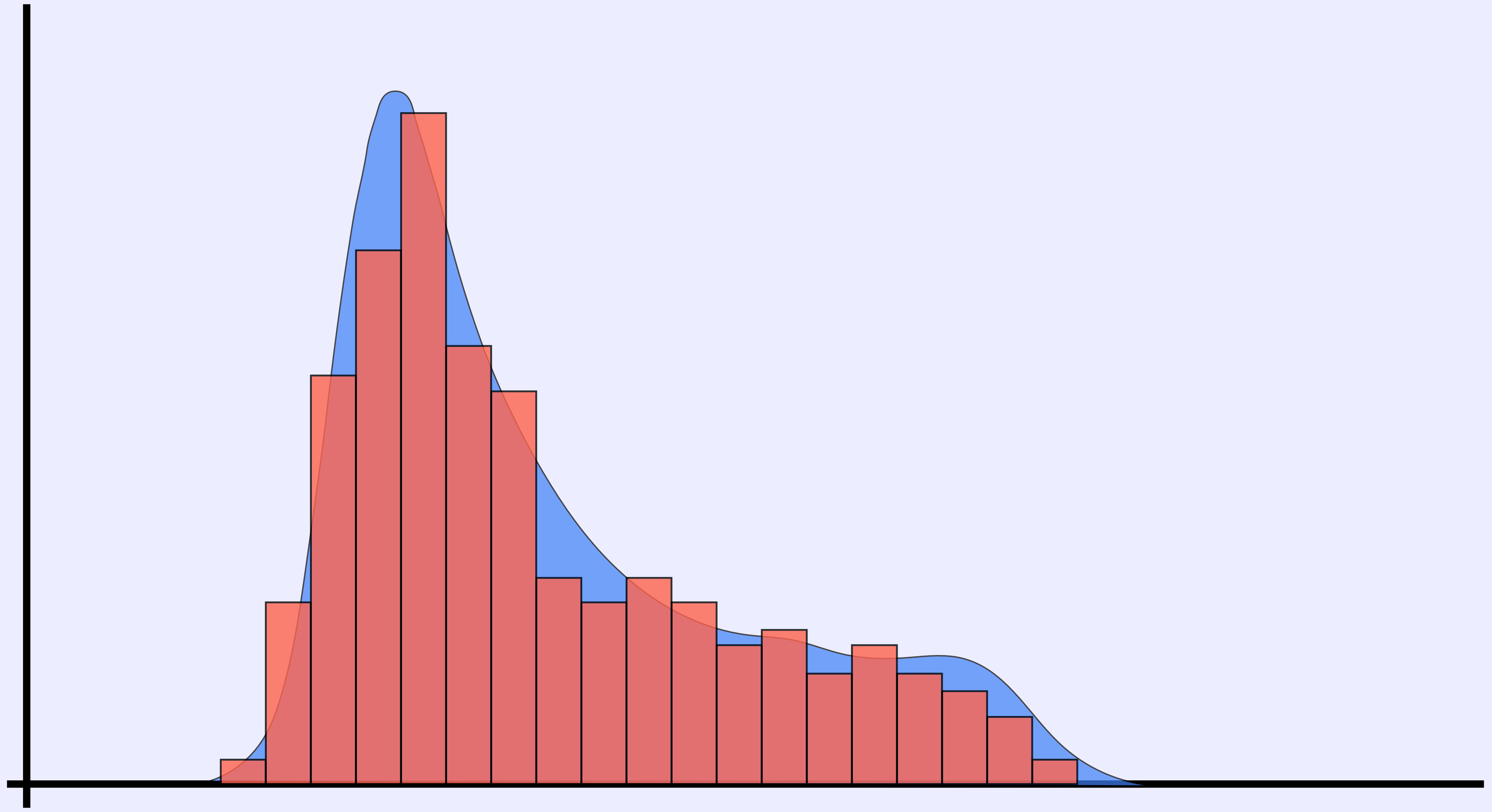
Supervised Learning

- Classical Supervised Machine Learning

$$\min_{w \in \mathbb{R}^d} \mathbb{E}_{(x,y)\sim\mathcal{P}} \left[\ell(y, \varphi(w, x))\right]$$

$(x_1, y_1), \ldots, (x_n, y_n) \sim \mathcal{P}$ Training Distribution

$(x'_1, y'_1), \ldots, (x'_n, y'_n) \sim \mathcal{P}'$ Testing Distribution



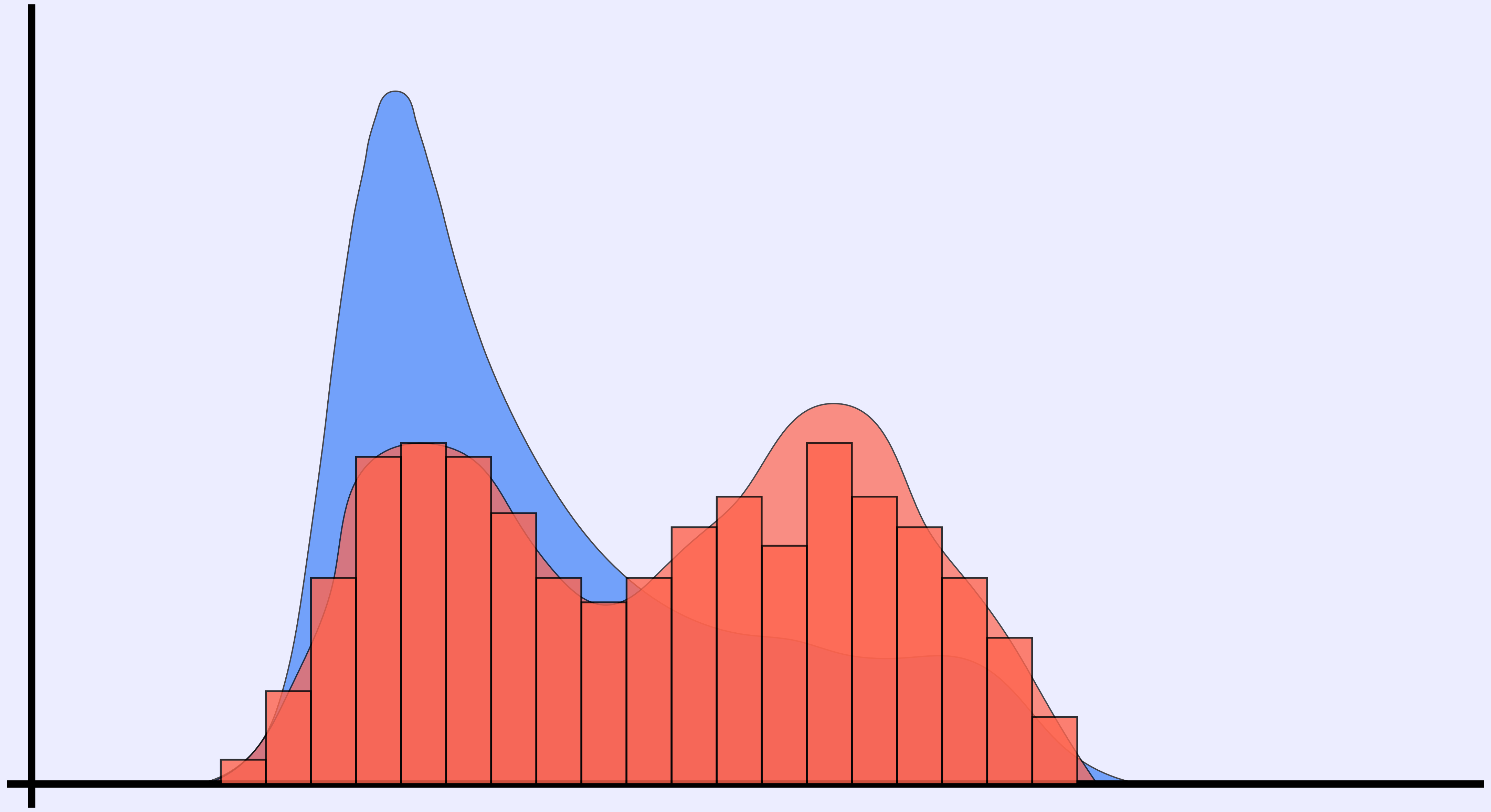
Supervised Learning

- Classical Supervised Machine Learning

$$\min_{w \in \mathbb{R}^d} \mathbb{E}_{(x,y)\sim\mathcal{P}} \left[\ell(y, \varphi(w, x))\right]$$

$(x_1, y_1), \ldots, (x_n, y_n) \sim \mathcal{P}$ Training Distribution

$(x'_1, y'_1), \ldots, (x'_n, y'_n) \sim \mathcal{P}'$ Testing Distribution



Supervised Learning

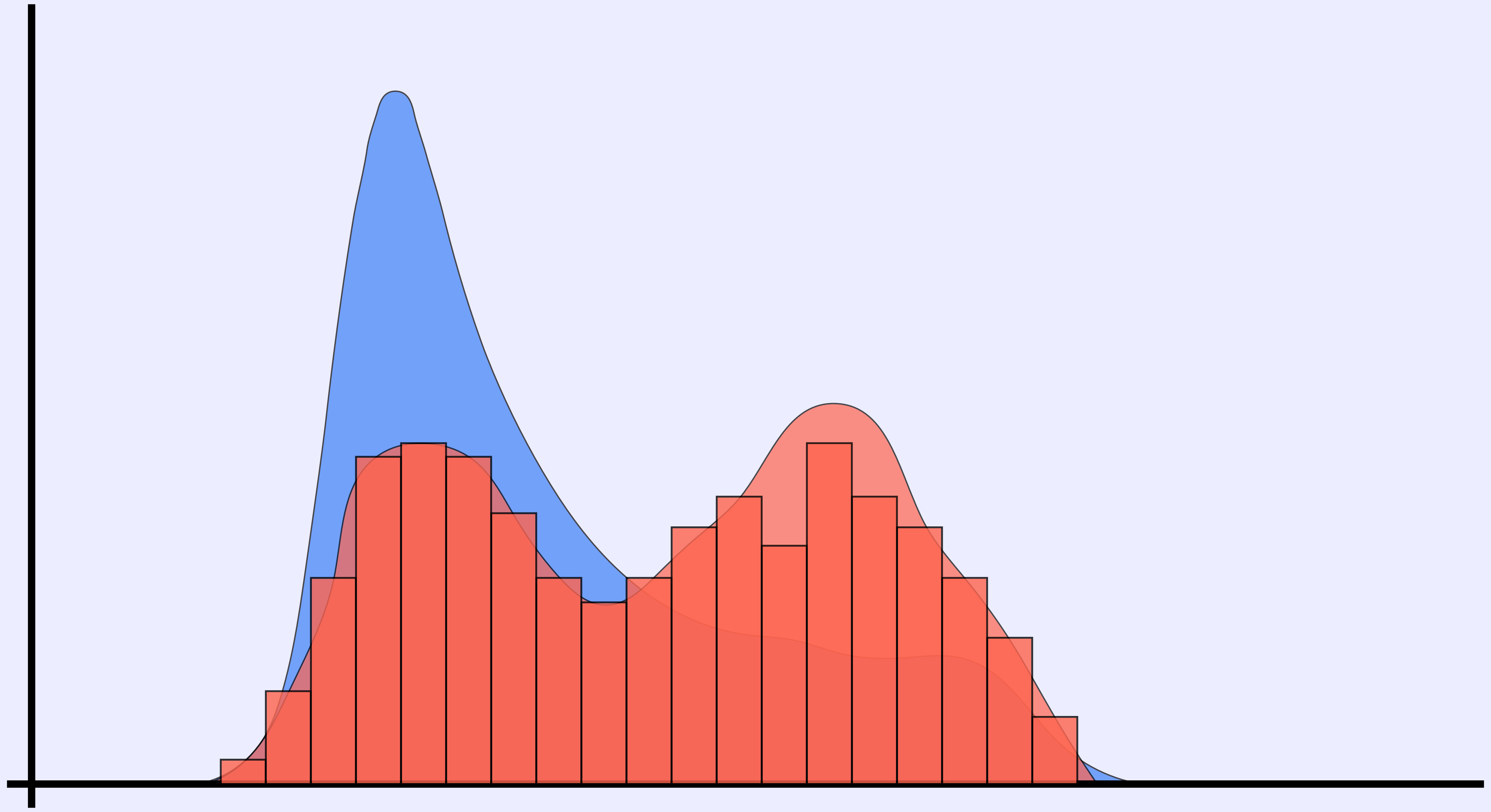
- Classical Supervised Machine Learning

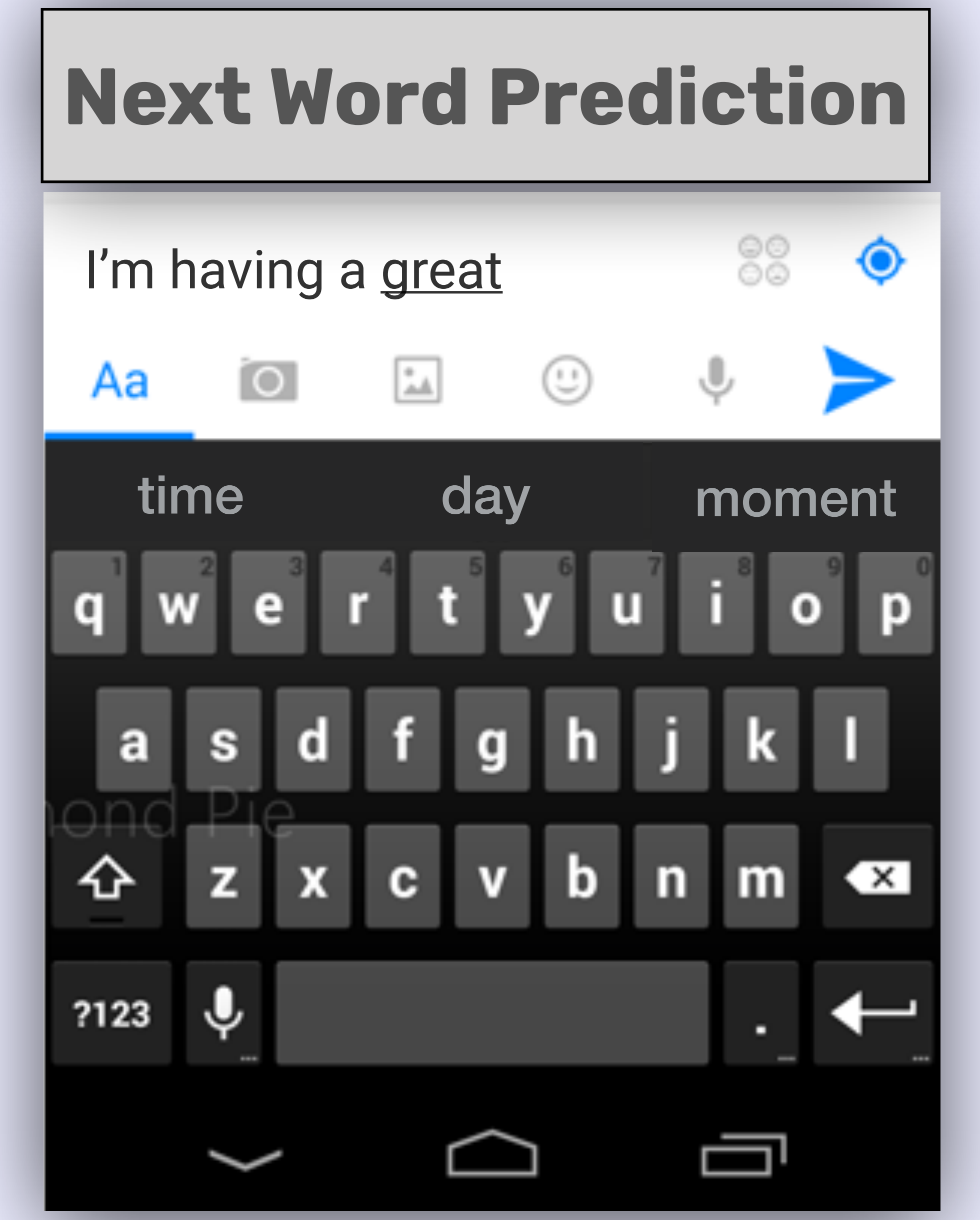
$$\min_{w \in \mathbb{R}^d} \mathbb{E}_{(x,y)\sim\mathcal{P}} \left[\ell(y, \varphi(w, x))\right]$$

$(x_1, y_1), \ldots, (x_n, y_n) \sim \mathcal{P}$ Training Distribution

$(x'_1, y'_1), \ldots, (x'_n, y'_n) \sim \mathcal{P}'$ Testing Distribution

- E.g.: Next word prediction on mobile phone - data distribution depends on the user.

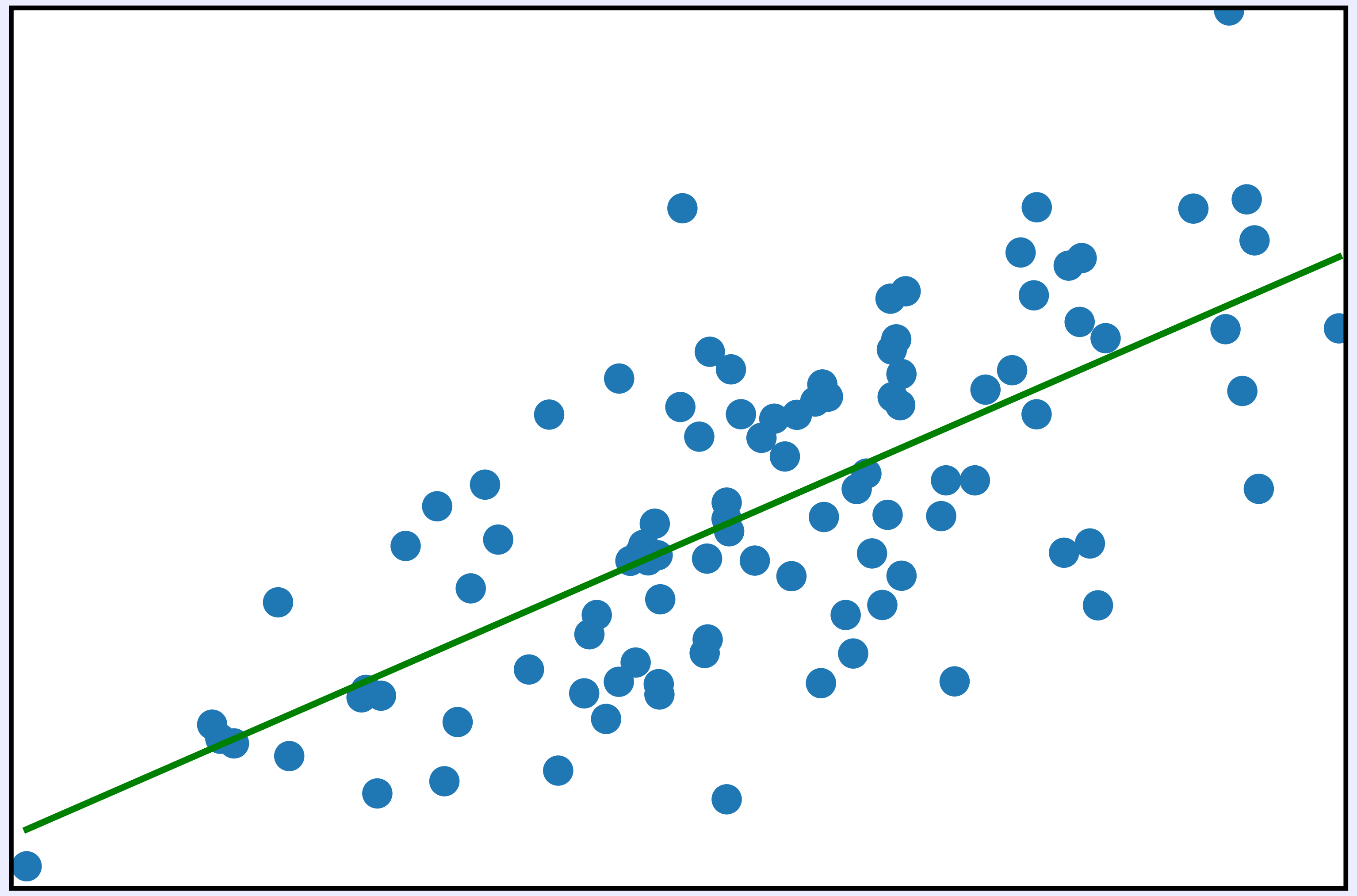




Safety for Ordinary Least Squares

- Ordinary Least Squares $\min_{w \in \mathbb{R}^d} \mathbb{E}\left[(Y - w^\top X)^2\right]$

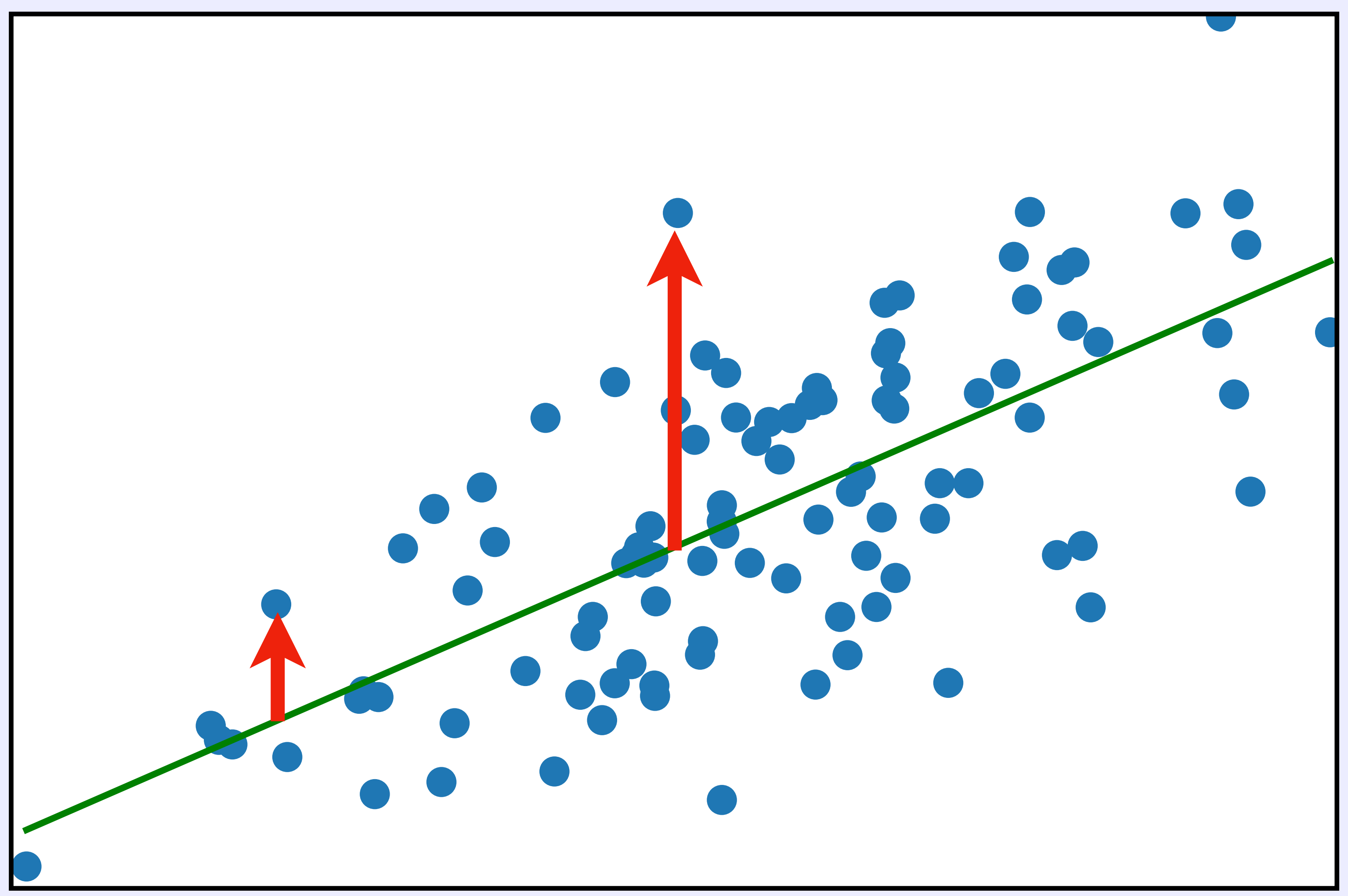
- Expectation is Risk Neutral



Safety for Ordinary Least Squares

- Ordinary Least Squares $\min_{w \in \mathbb{R}^d} \mathbb{E}\left[(Y - w^\top X)^2\right]$

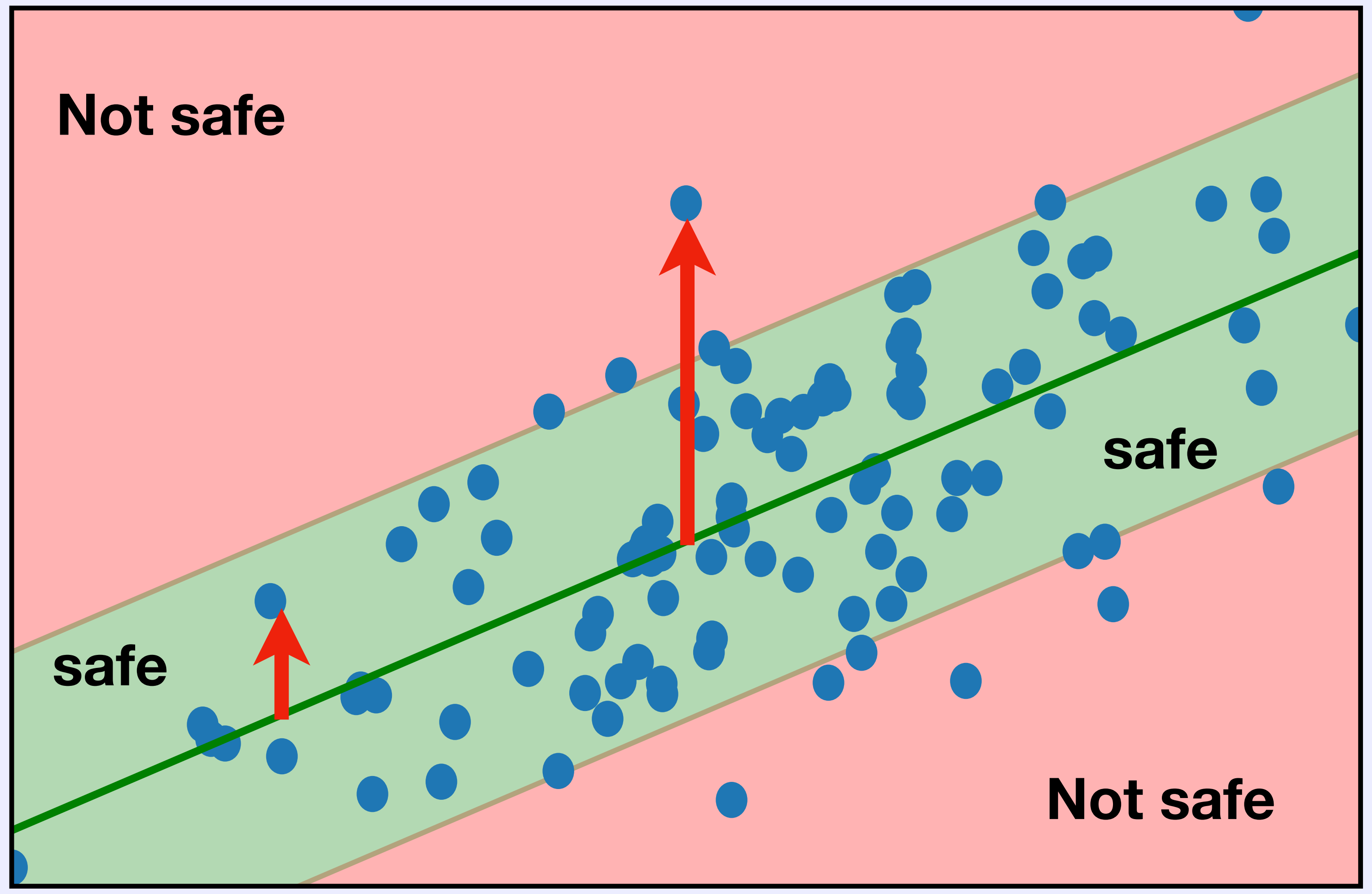
- Expectation is Risk Neutral



Safety for Ordinary Least Squares

- Ordinary Least Squares $\min_{w \in \mathbb{R}^d} \mathbb{E}\left[(Y - w^\top X)^2\right]$

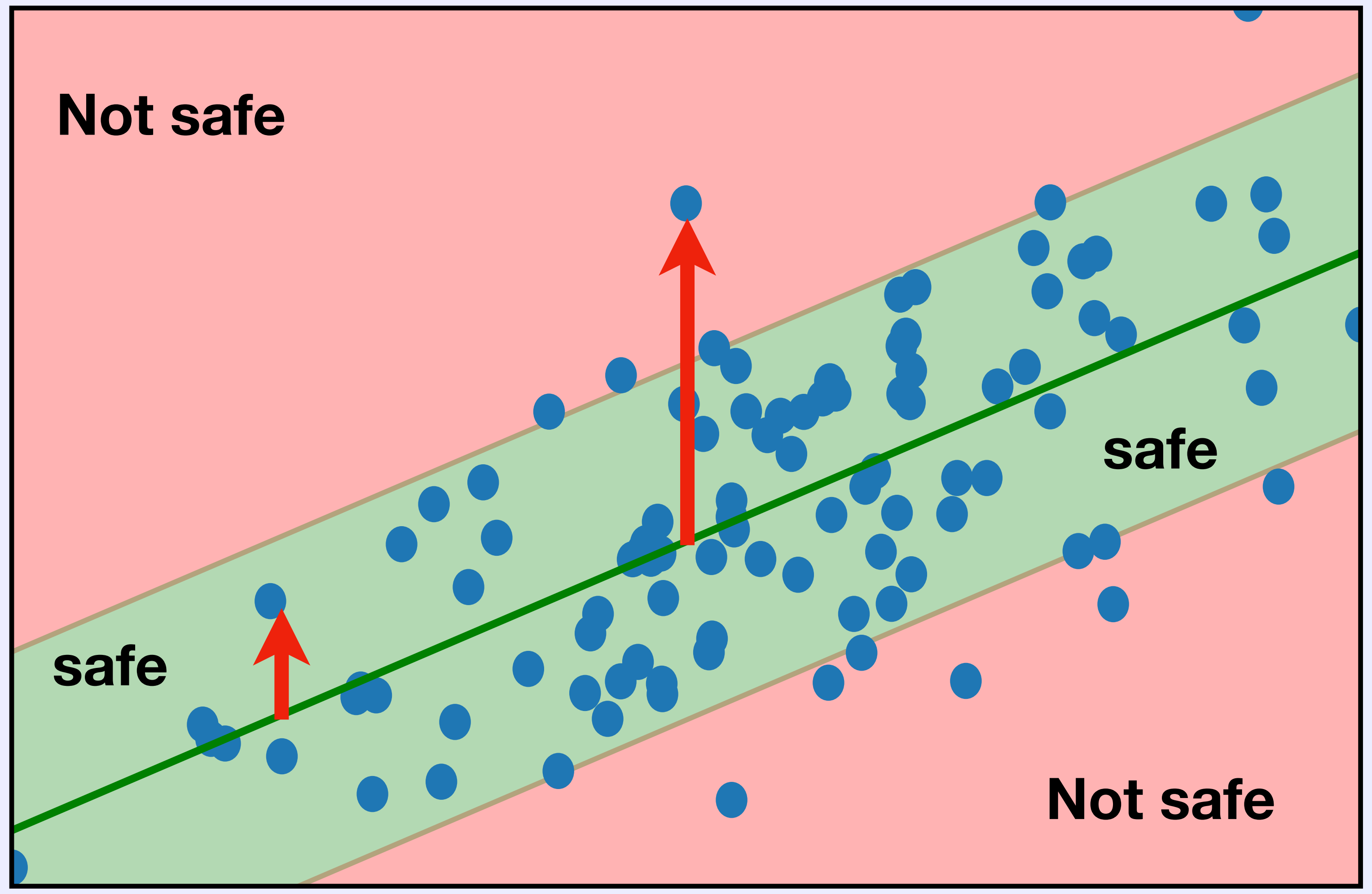
- Expectation is Risk Neutral



Safety for Ordinary Least Squares

- Ordinary Least Squares $\min_{w \in \mathbb{R}^d} \mathbb{E}\left[(Y - w^\top X)^2\right]$

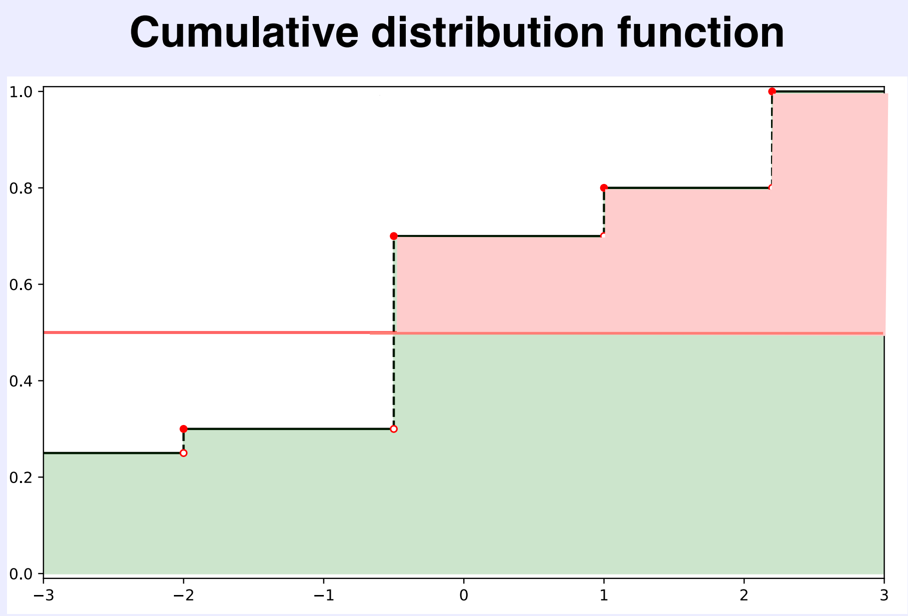
- Expectation is Risk Neutral

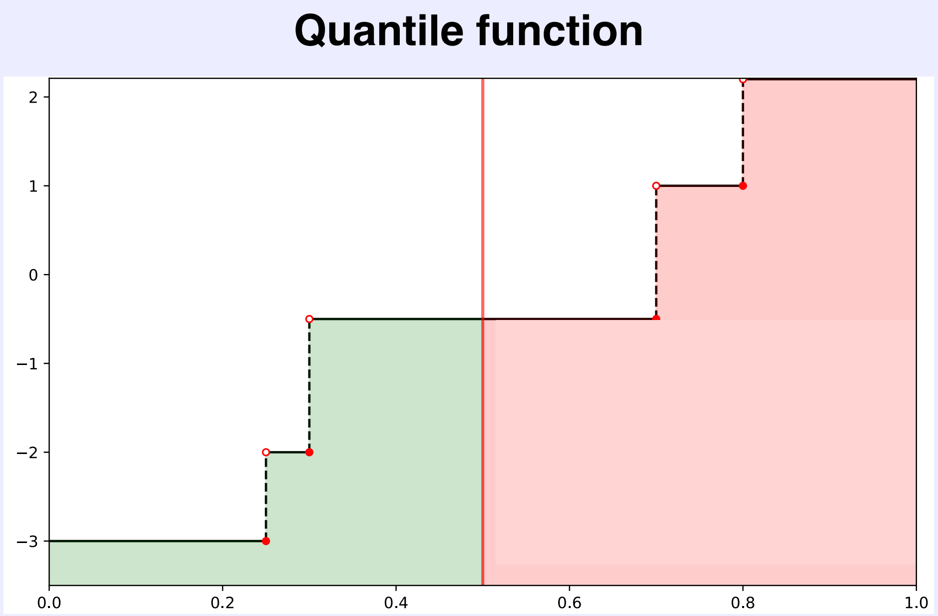


- Building a Risk-averse model

$$\varepsilon = (Y - w^\top X)^2$$

p-quantile	$Q_p(\varepsilon) = \min\{t \in \mathbb{R}, \mathbb{P}[\varepsilon \leq t] \geq p\}$

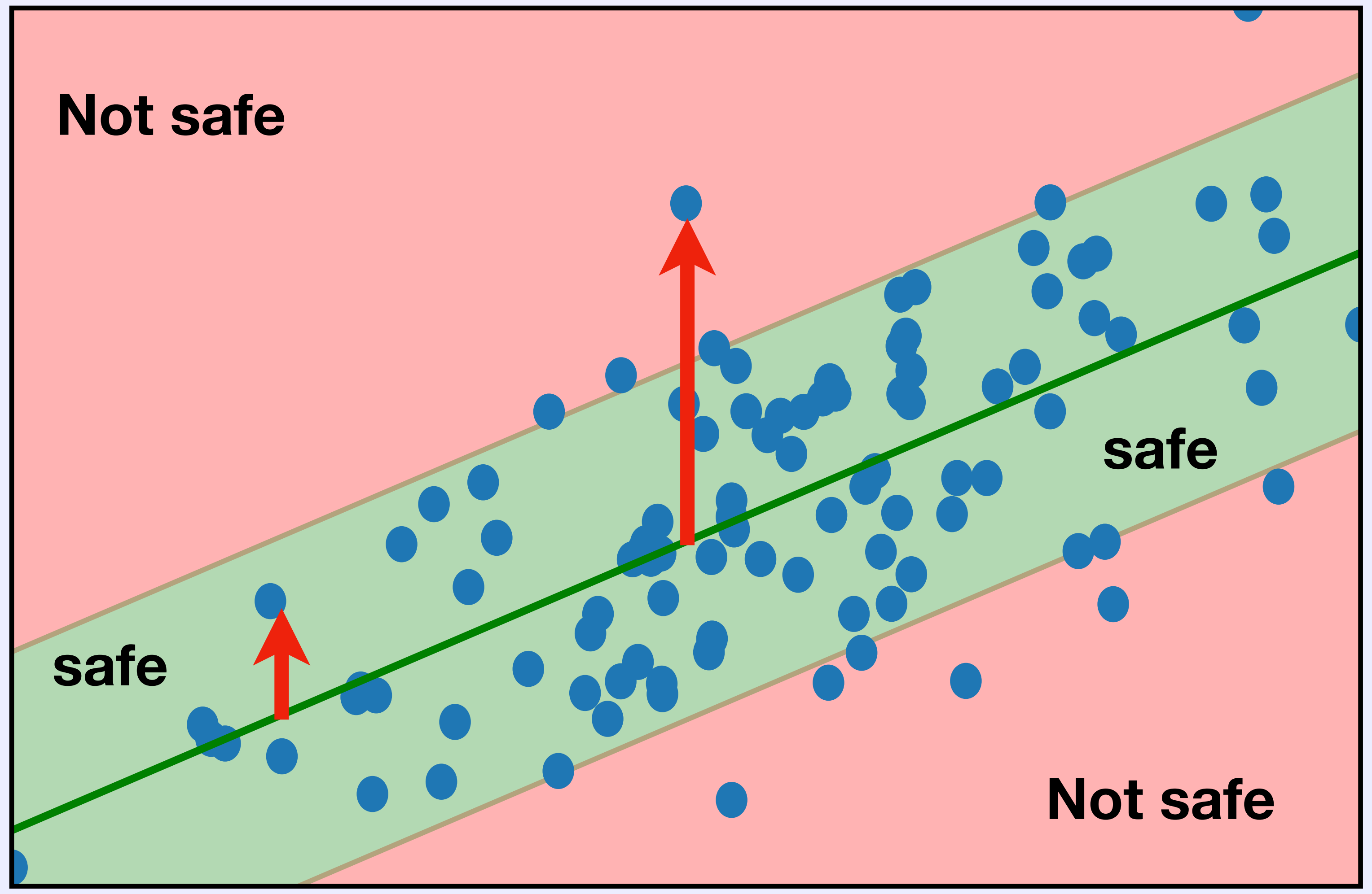




Safety for Ordinary Least Squares

- Ordinary Least Squares $\min_{w \in \mathbb{R}^d} \mathbb{E}\left[(Y - w^\top X)^2\right]$

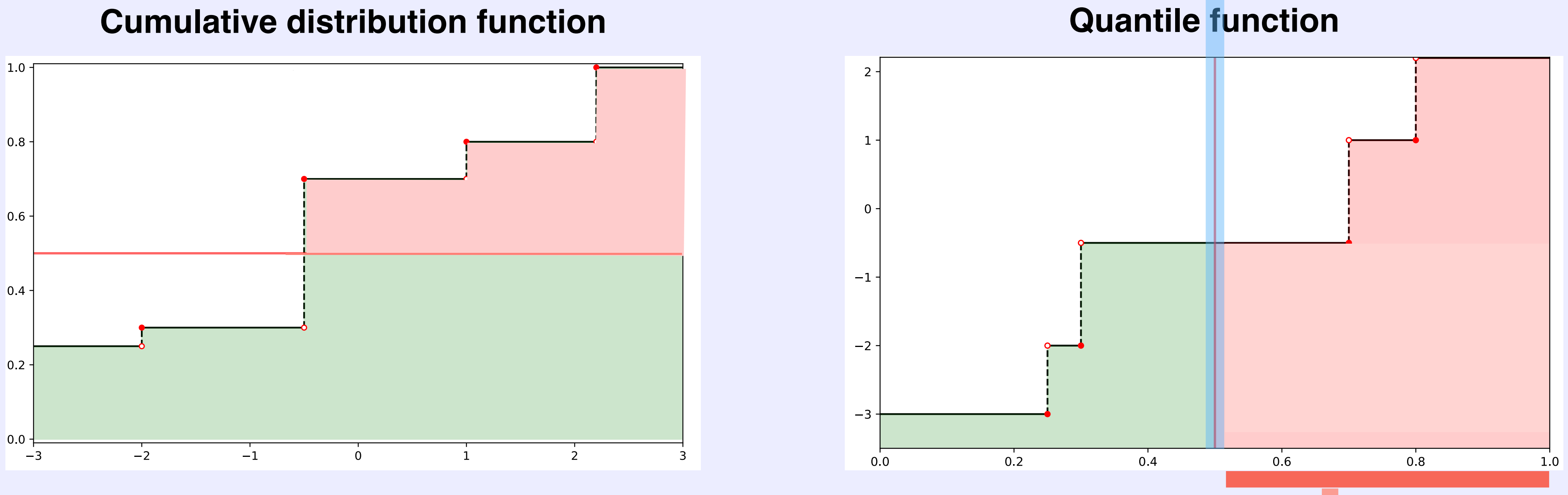
- Expectation is Risk Neutral



- Building a Risk-averse model

$$\varepsilon = (Y - w^\top X)^2$$

p-quantile $\quad Q_p(\varepsilon) = \min\{t \in \mathbb{R}, \mathbb{P}[\varepsilon \leq t] \geq p\}$



p-superquantile $\quad \bar{Q}_p(\varepsilon) = \frac{1}{1-p}\int_{p'=p}^{1} Q_{p'}(\varepsilon)dp'$

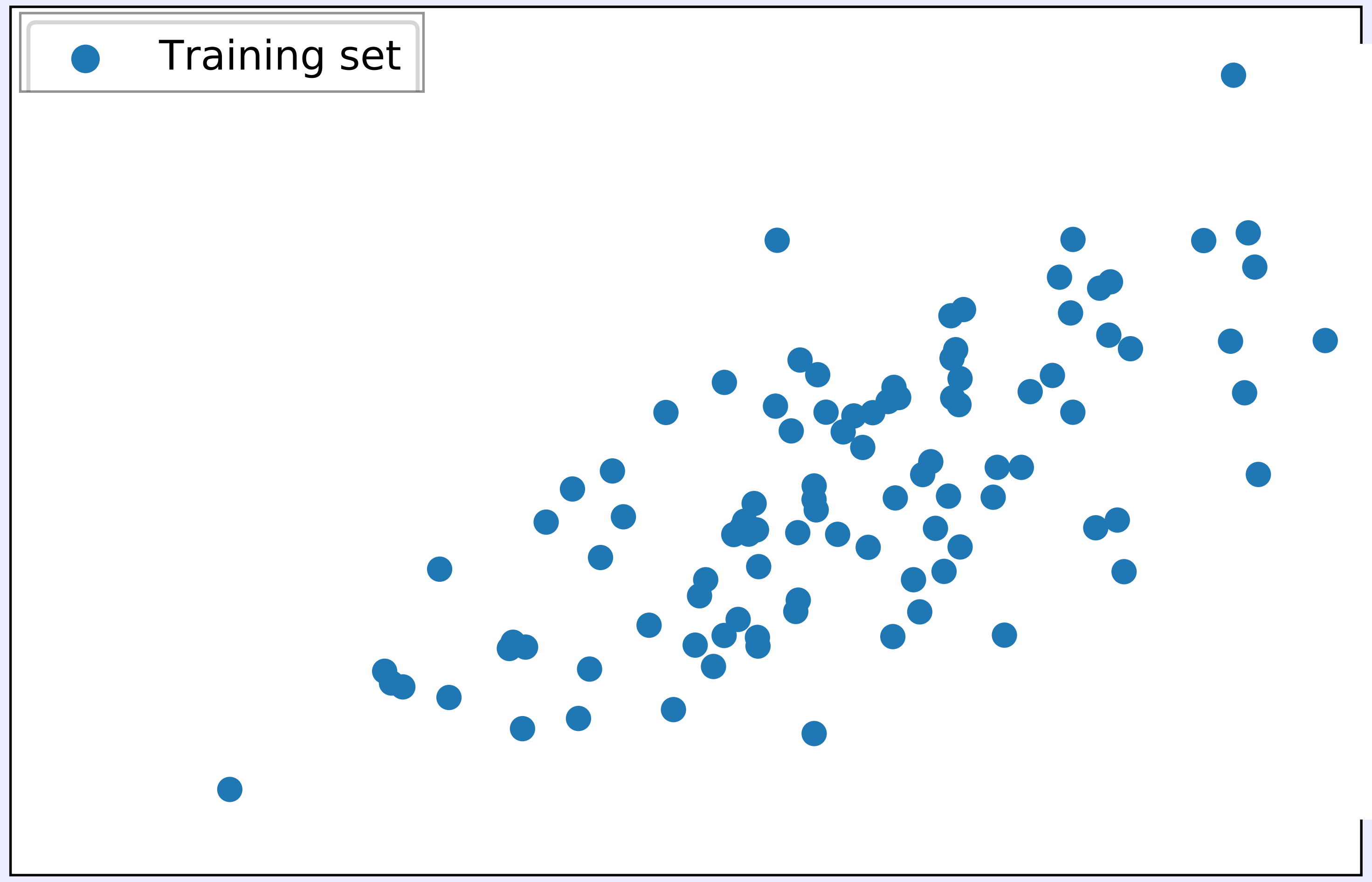
[Rockafellar, Uryasev 00']

Dual Formulation for the Superquantile

- Ordinary Least Squares $\min_{w \in \mathbb{R}^d} \mathbb{E}\left[(Y - w^\top X)^2\right]$

- Empirical Risk Minimization

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - w^\top x_i)^2 = \min_{w \in \mathbb{R}^d} \mathbb{E}_{\hat{\mathbb{P}}_n} (Y - w^\top X)^2$$

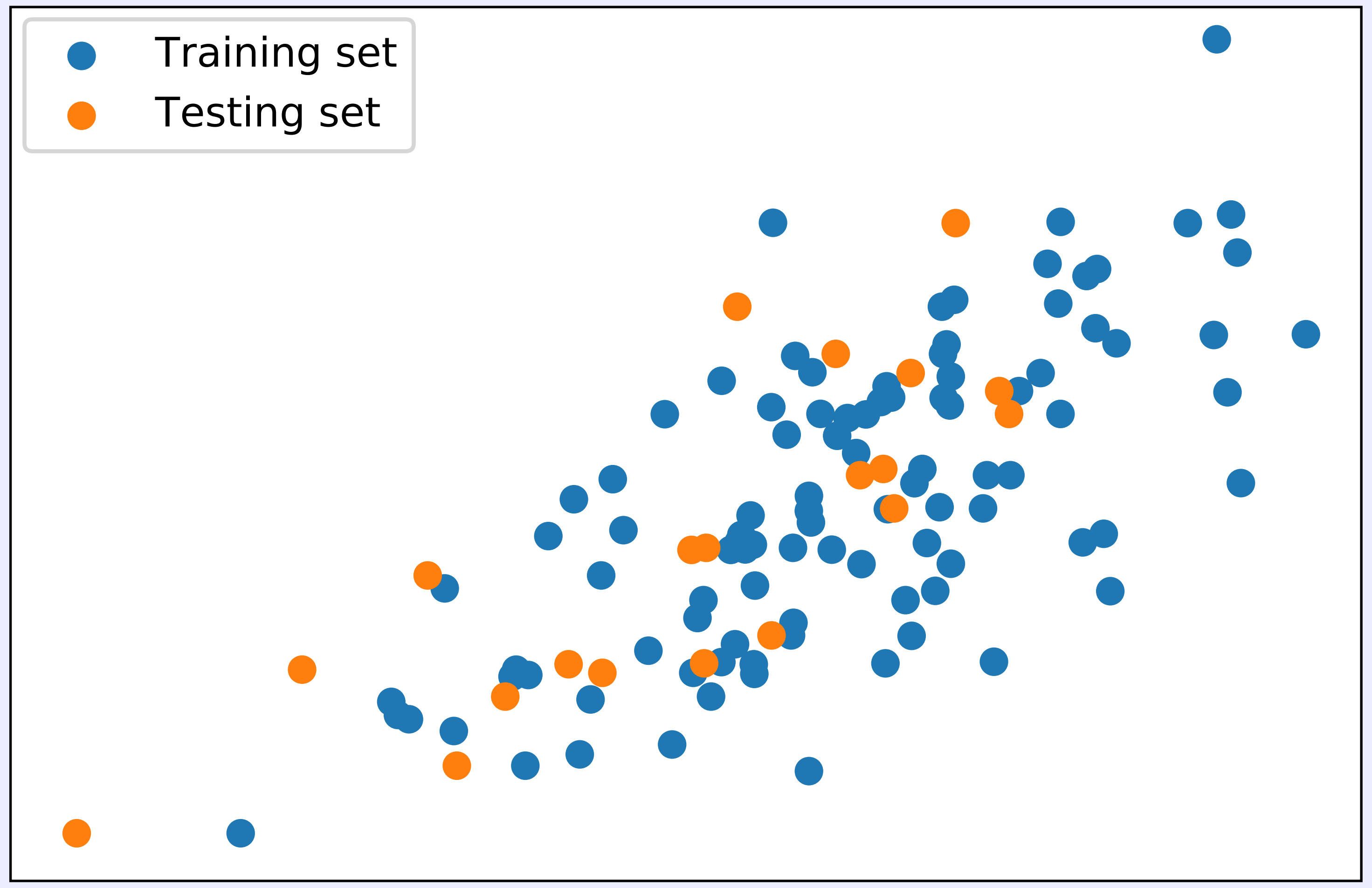


Dual Formulation for the Superquantile

- Ordinary Least Squares $\min_{w \in \mathbb{R}^d} \mathbb{E}\left[(Y - w^\top X)^2\right]$

- Empirical Risk Minimization

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - w^\top x_i)^2 = \min_{w \in \mathbb{R}^d} \mathbb{E}_{\hat{\mathbb{P}}_n} (Y - w^\top X)^2$$

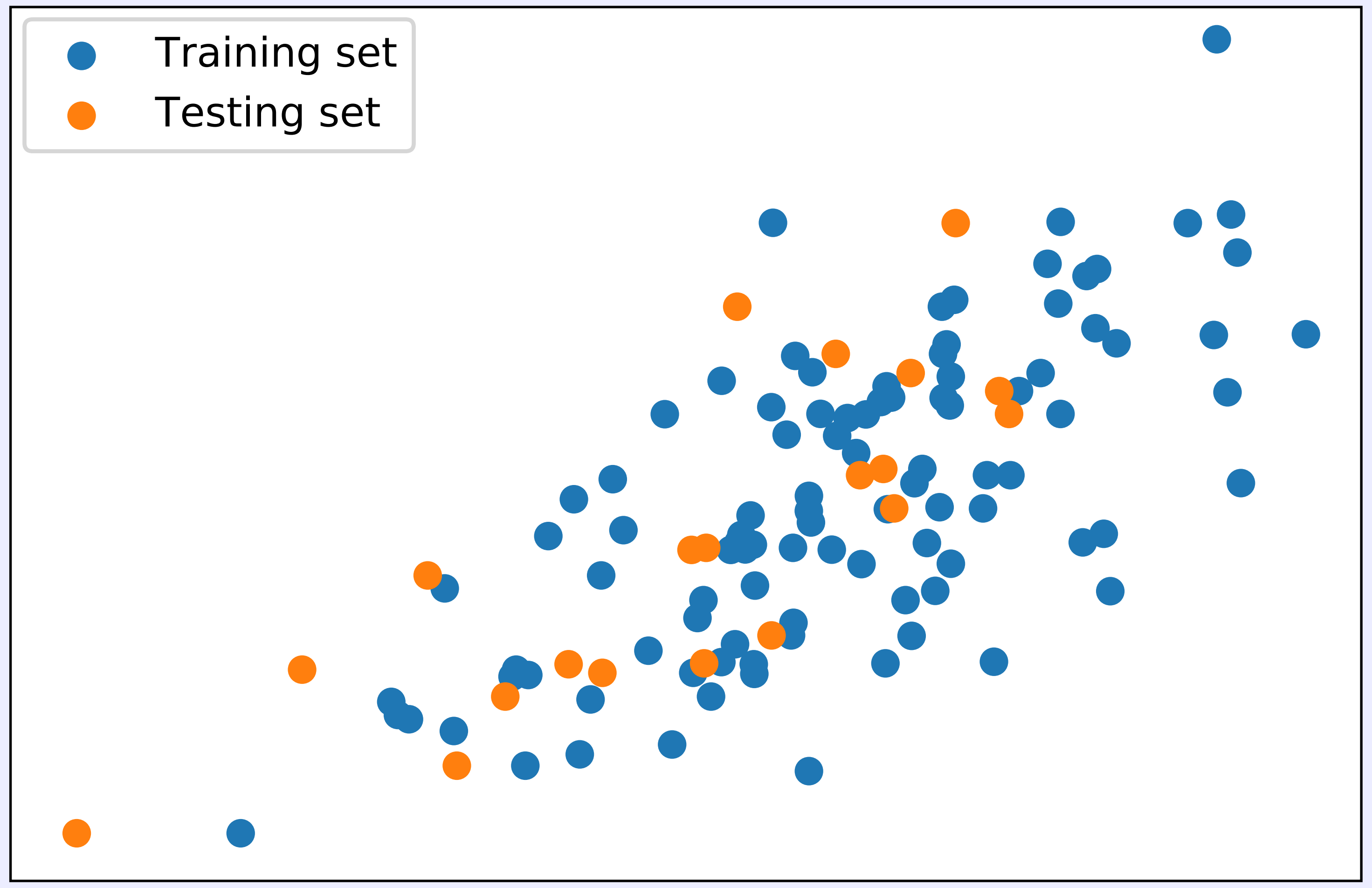


Dual Formulation for the Superquantile

- Ordinary Least Squares $\min_{w \in \mathbb{R}^d} \mathbb{E}\left[(Y - w^\top X)^2\right]$

- Empirical Risk Minimization

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - w^\top x_i)^2 = \min_{w \in \mathbb{R}^d} \mathbb{E}_{\hat{\mathbb{P}}_n} (Y - w^\top X)^2$$



- Distributionally Robust Optimization

$$\min_{w \in \mathbb{R}^d} \max_{Q \in \mathcal{A}_p} \mathbb{E}_Q[(Y - w^\top X)^2]$$

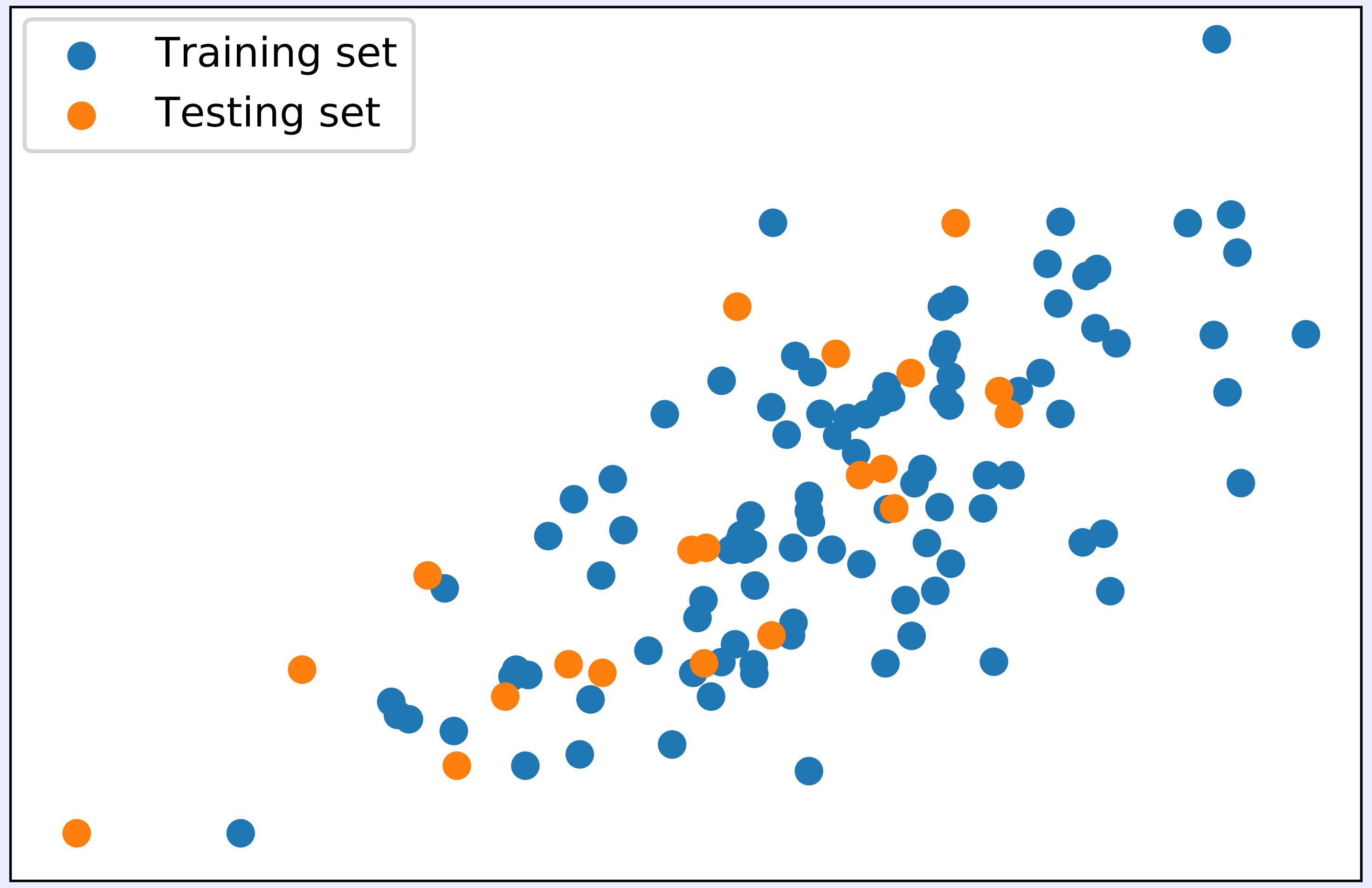
Ambiguity Set

Dual Formulation for the Superquantile

- Ordinary Least Squares $\min_{w \in \mathbb{R}^d} \mathbb{E}\left[(Y - w^\top X)^2\right]$

- Empirical Risk Minimization

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - w^\top x_i)^2 = \min_{w \in \mathbb{R}^d} \mathbb{E}_{\hat{\mathbb{P}}_n} (Y - w^\top X)^2$$

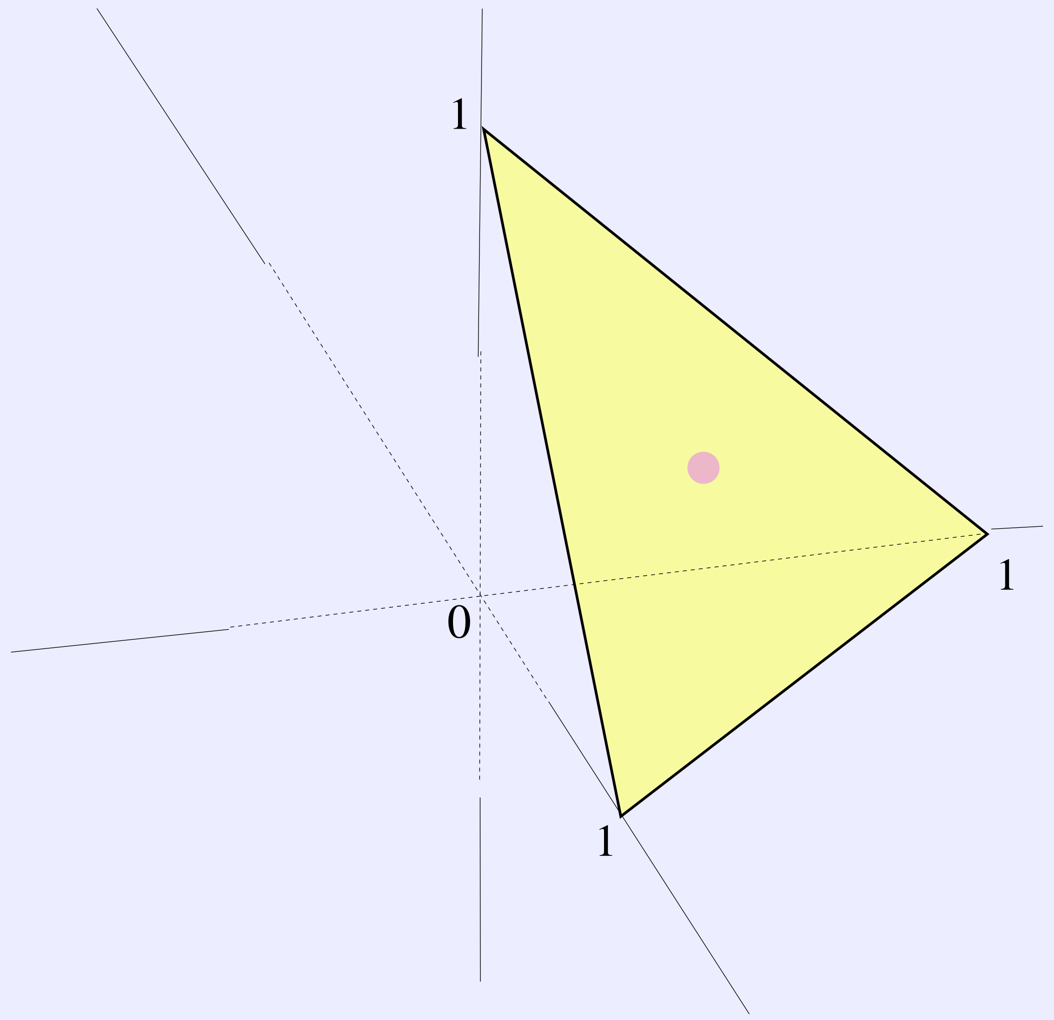


- Distributionally Robust Optimization

$$\min_{w \in \mathbb{R}^d} \max_{Q \in \mathcal{A}_p} \mathbb{E}_Q[(Y - w^\top X)^2]$$

Ambiguity Set

$$\mathcal{A} = \{\hat{\mathbb{P}}_n\}$$

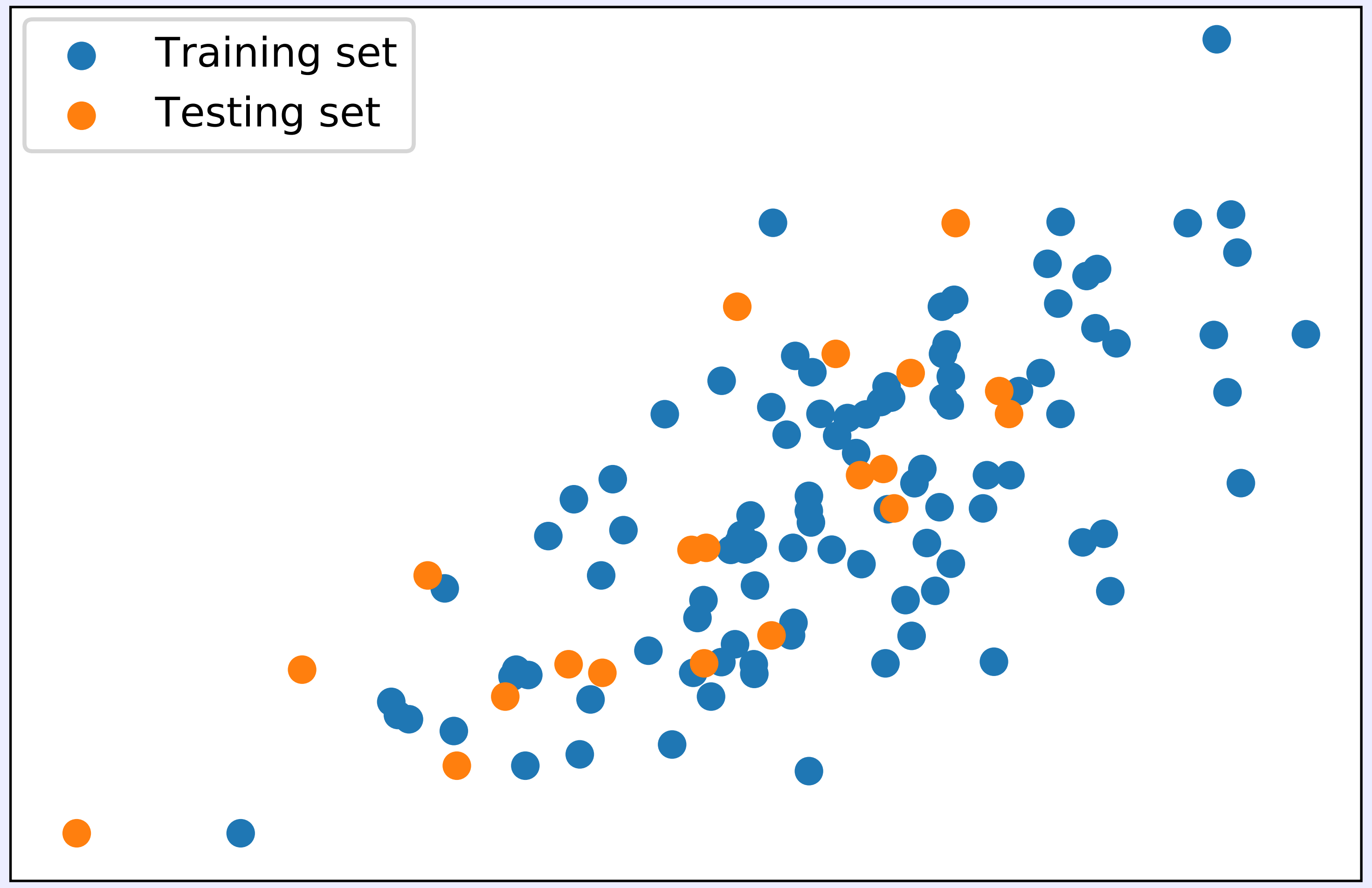


Dual Formulation for the Superquantile

- Ordinary Least Squares $\min_{w \in \mathbb{R}^d} \mathbb{E}\left[(Y - w^\top X)^2\right]$

- Empirical Risk Minimization

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - w^\top x_i)^2 = \min_{w \in \mathbb{R}^d} \mathbb{E}_{\hat{\mathbb{P}}_n} (Y - w^\top X)^2$$

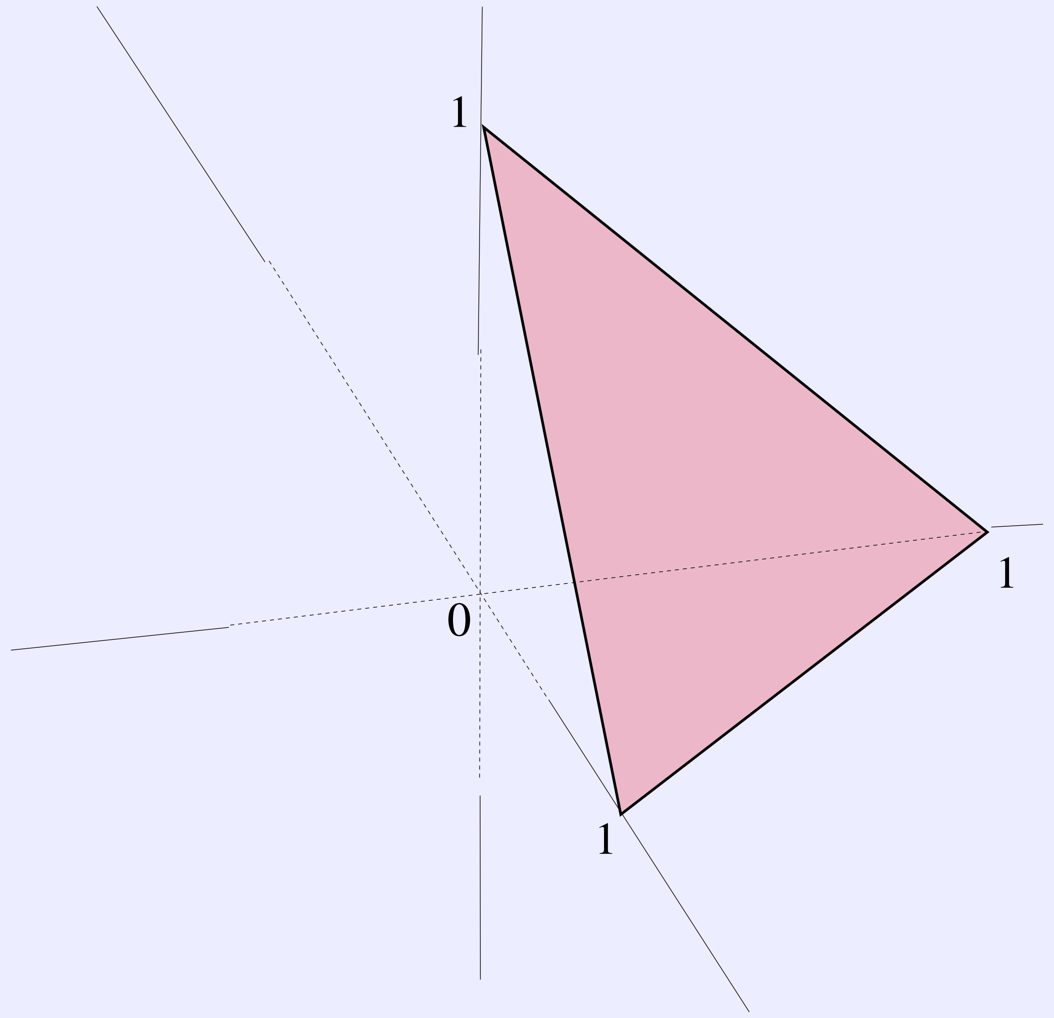


- Distributionally Robust Optimization

$$\min_{w \in \mathbb{R}^d} \max_{Q \in \mathcal{A}_p} \mathbb{E}_Q[(Y - w^\top X)^2]$$

Ambiguity Set

$$\mathcal{A} = \Delta_{n-1}$$

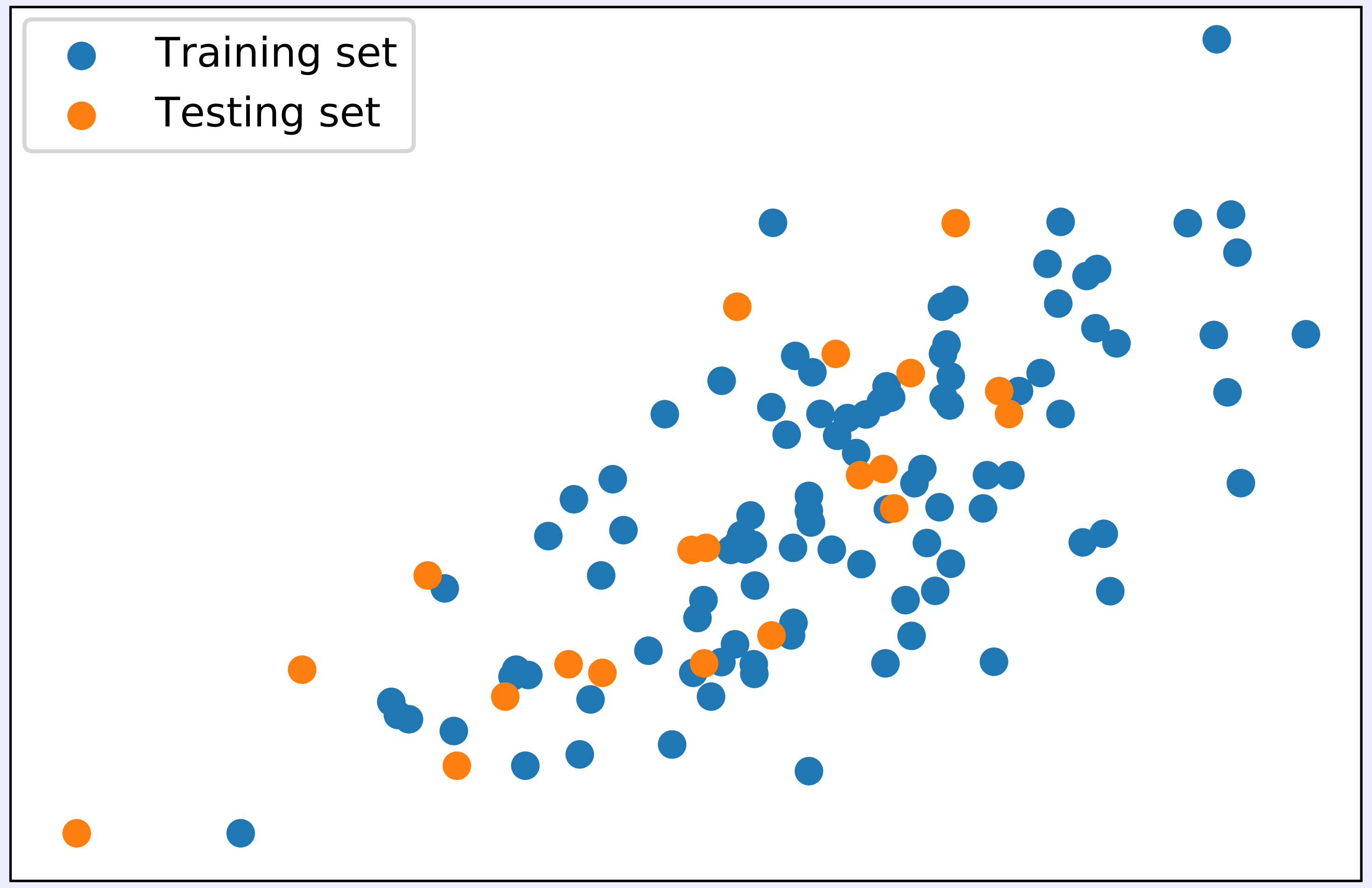


Dual Formulation for the Superquantile

- Ordinary Least Squares $\min_{w \in \mathbb{R}^d} \mathbb{E}\left[(Y - w^\top X)^2\right]$

- Empirical Risk Minimization

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - w^\top x_i)^2 = \min_{w \in \mathbb{R}^d} \mathbb{E}_{\hat{\mathbb{P}}_n} (Y - w^\top X)^2$$



- Distributionally Robust Optimization

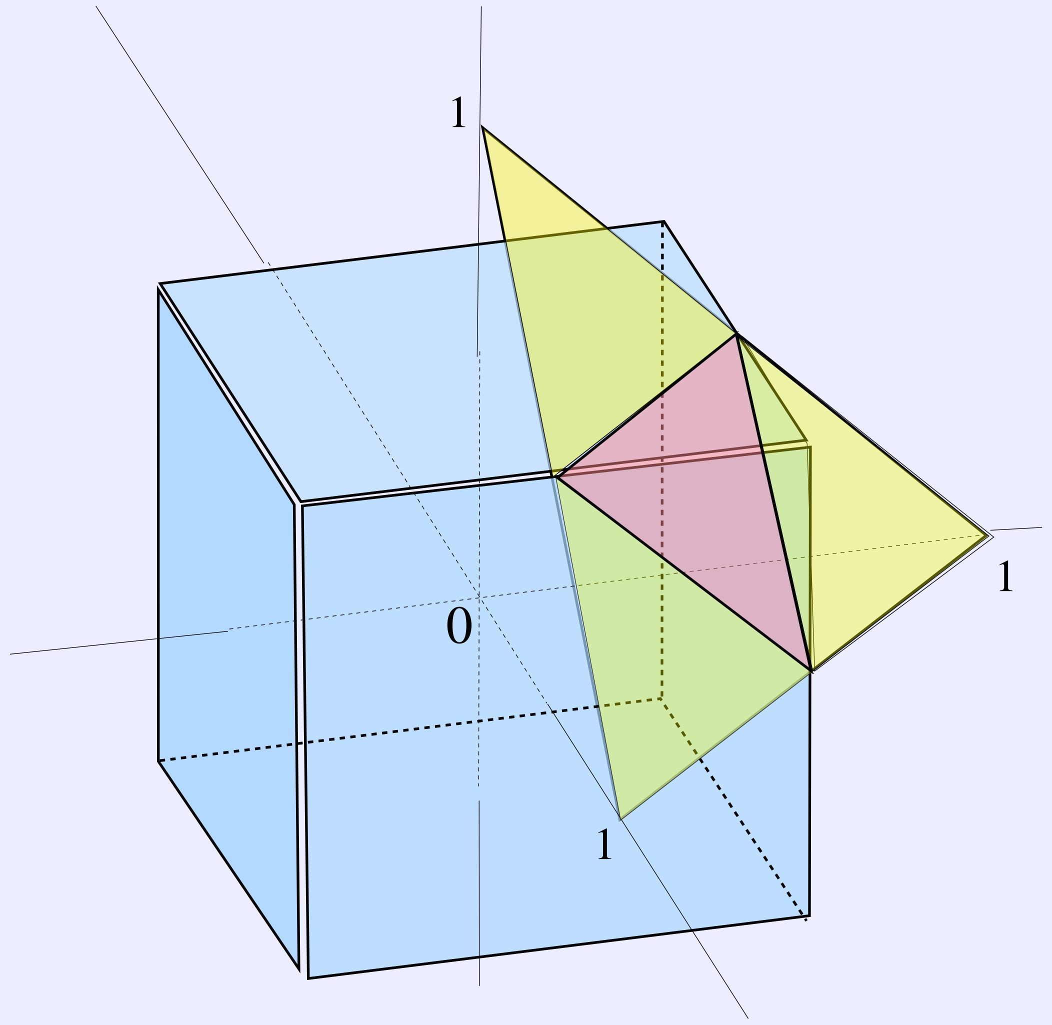
$$\min_{w \in \mathbb{R}^d} \max_{Q \in \mathcal{A}_p} \mathbb{E}_Q[(Y - w^\top X)^2]$$

Ambiguity Set

[Ben-Tal, Teboulle 07']

$$\mathcal{A}_p = \Delta_{n-1} \cap B\left(0, \frac{1}{n(1-p)}\right)$$

$$\max_{Q \in \mathcal{A}_p} \mathbb{E}_Q[(Y - w^\top X)^2] = \bar{Q}_p[(Y - w^\top X)^2]$$



1 Safety in Supervised ML

3 BEHIND SPQR

2 THE TOOLBOX SPQR

USAGE

Solving $\min_{w \in \mathbb{R}^d} \bar{Q}_p(L(w))$

- **Input**
  - **Oracle**
    ```
    L(w,x,y)
    L_prime(w,x,y)
    ```
  - **Dataset**
    ```
    X,Y
    ```

USAGE

How to solve: $\min_{w \in \mathbb{R}^d} \bar{Q}_p(L(w))$

- **Input**
  - **Oracle**
    - `L(w,x,y)`
    - `L_prime(w,x,y)`
  - **Dataset**
    - `X,Y`

Example : Least squares regression

In[1]:

```
import numpy as np
# Define the loss and derivative
def L(w, x, y):
    return (y - np.dot(x,w))**2
def L_prime(w, x, y):
    return -2.0 * (y - np.dot(x,w)) * x
```

In[2]:

```
# The dataset
X = np.random.rand(100,2)
alpha = np.array([1.,2.])
Y = np.dot(X, alpha) + np.random.rand(100)
```

USAGE

How to solve: $\min_{w \in \mathbb{R}^d} \bar{Q}_p(L(w))$

- **Input**
  - **Oracle**
    - `L(w,x,y)`
    - `L_prime(w,x,y)`
  - **Dataset**
    - `X,Y`
- **Classical Algorithms**
  - **If L is convex non-smooth**

    Subgradient method, dual averaging
  - **If L is smooth**

    Gradient descent, Nesterov Accelerated Gradient, Quasi-Newton

Built on top of Scikit-Learn

The RiskOptimizer Object

In[3]:

```
from spqr import RiskOptimizer
# Instantiate a risk optimiser object
optimiser = RiskOptimizer(L, L_prime, p=0.9)
```

In[4]:

```
# Running the algorithm
optimiser.fit(X,Y)
```

USAGE

How to solve: $\min_{w \in \mathbb{R}^d} \bar{Q}_p(L(w))$

Built on top of Scikit-Learn

- **Input**
  - **Oracle**
    - `L(w,x,y)`
    - `L_prime(w,x,y)`
  - **Dataset**
    - `X,Y`
- **Algorithms**
  - **If L is convex non-smooth**
    Subgradient method, dual averaging
  - **If L is smooth**
    Gradient descent, Nesterov Accelerated Gradient, Quasi-Newton

The RiskOptimizer Object

In[3]:
```
from spqr import RiskOptimizer
# Instantiate a risk optimiser object
optimiser = RiskOptimizer(L, L_prime, p=0.9)
```

In[4]:
```
# Running the algorithm
optimiser.fit(X,Y)
```

The Output

In[5]:
```
# Solution provided
sol = optimiser.solution
```

DOCUMENTATION

https://yassine-laguel.github.io/spqr/

SPQR

Search docs

TABLE OF CONTENTS

Docs » SPQR

View page source

SPQR

SPQR is a python toolbox for optimization of superquantile-based risk measures.For more details, we refer to the companion paper “First Order Algorithms for Minimization of superquantile-based Risk Measures”.

Overview

For a couple of features and labels (X, y),this toolbox is aimed at minimizing functions of the form :

$$\phi(w) = \mathrm{CVAR}_p \circ L_{X,y}(w),$$

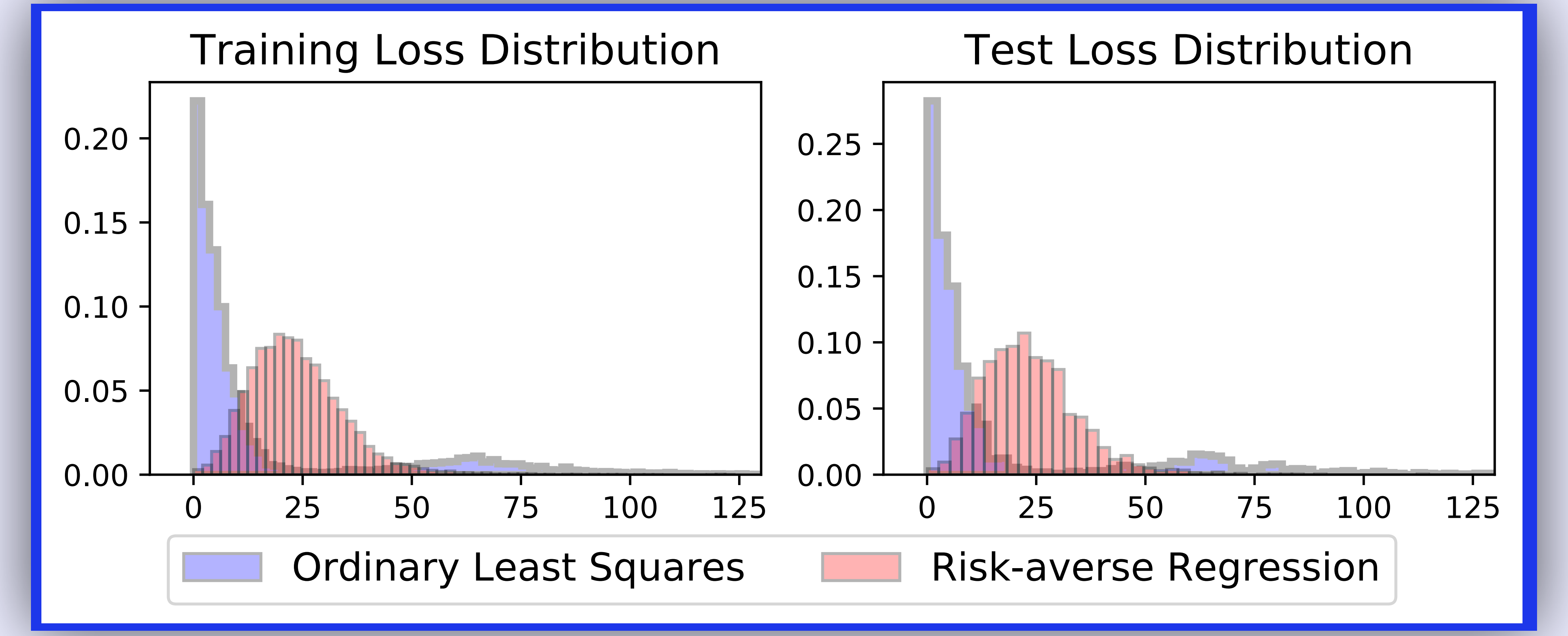
where CVAR denotes the superquantile, also called “conditional value at risk”, “average value at risk” or “expected shortfall” and loss function L is assumed to be provided by the user together with the dataset (X, y).

We build oracles for the nonsmooth function ϕ and for a smoothed counterpart ϕ_μ . Various first-order algorithms are proposed to minimise these 2 functions. Among these first order algorithms, one can find the Dual Averaging Method, Nesterov Accelerated Method or BFGS. For instance, quantile regression and superquantile regression can be performed with this toolbox :

NUMERICAL EXPERIMENTS

- **On a synthetic dataset**
  - **Data Generation** $y_i = w^\top x_i + \varepsilon_i$
  - **Noise Modeling** $\varepsilon_i \sim \beta \varepsilon_{\mathcal{N}} + (1-\beta)\varepsilon_{\mathcal{L}}$

    β: $Bernoulli\ (0.8)$, $\varepsilon_{\mathcal{N}}$: $Normal\ (0,1)$, $\varepsilon_{\mathcal{L}}$: $Laplace\ (10,1)$
  - **Squared Residuals** $r_i^2 = (y_i - w^\top x_i)^2$
  - **Safety parameter** p=0.9

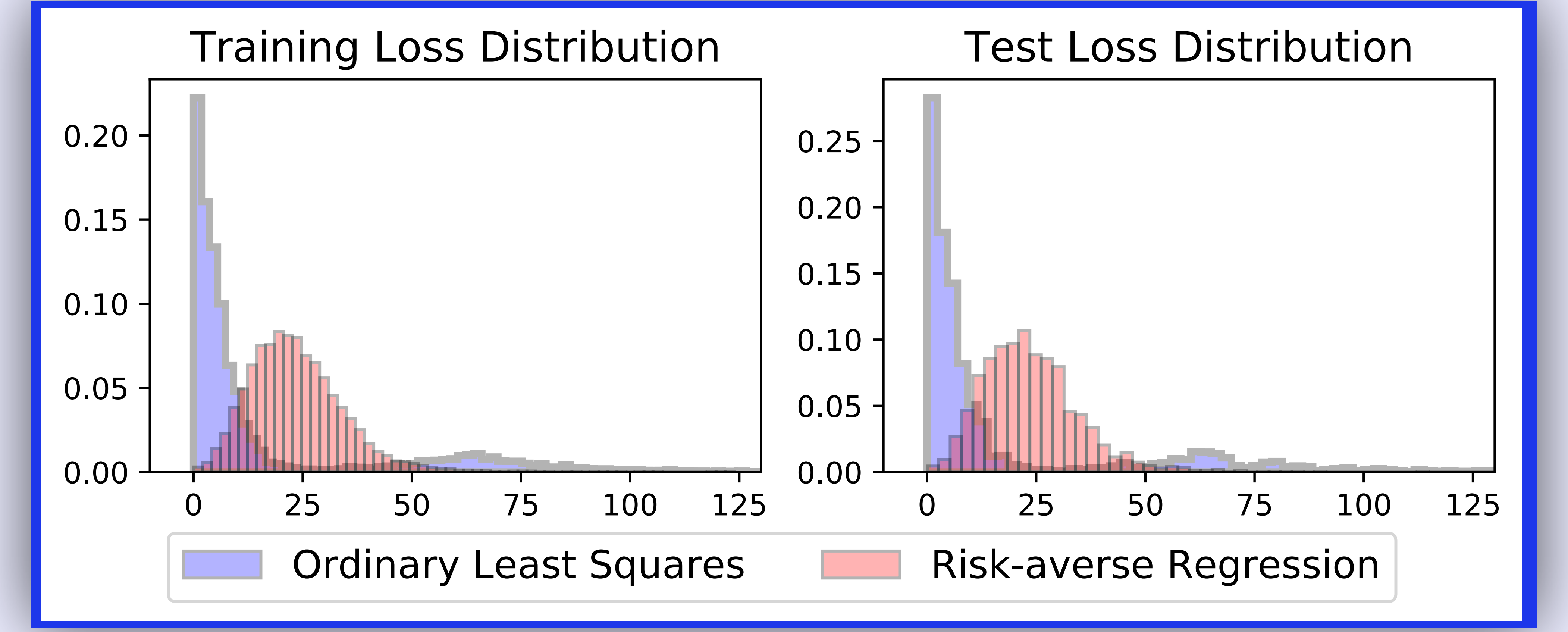


NUMERICAL EXPERIMENTS

On a synthetic dataset

- **Data Generation** $y_i = w^\top x_i + \varepsilon_i$
- **Noise Modeling** $\varepsilon_i \sim \beta \varepsilon_{\mathcal{N}} + (1-\beta)\varepsilon_{\mathcal{L}}$

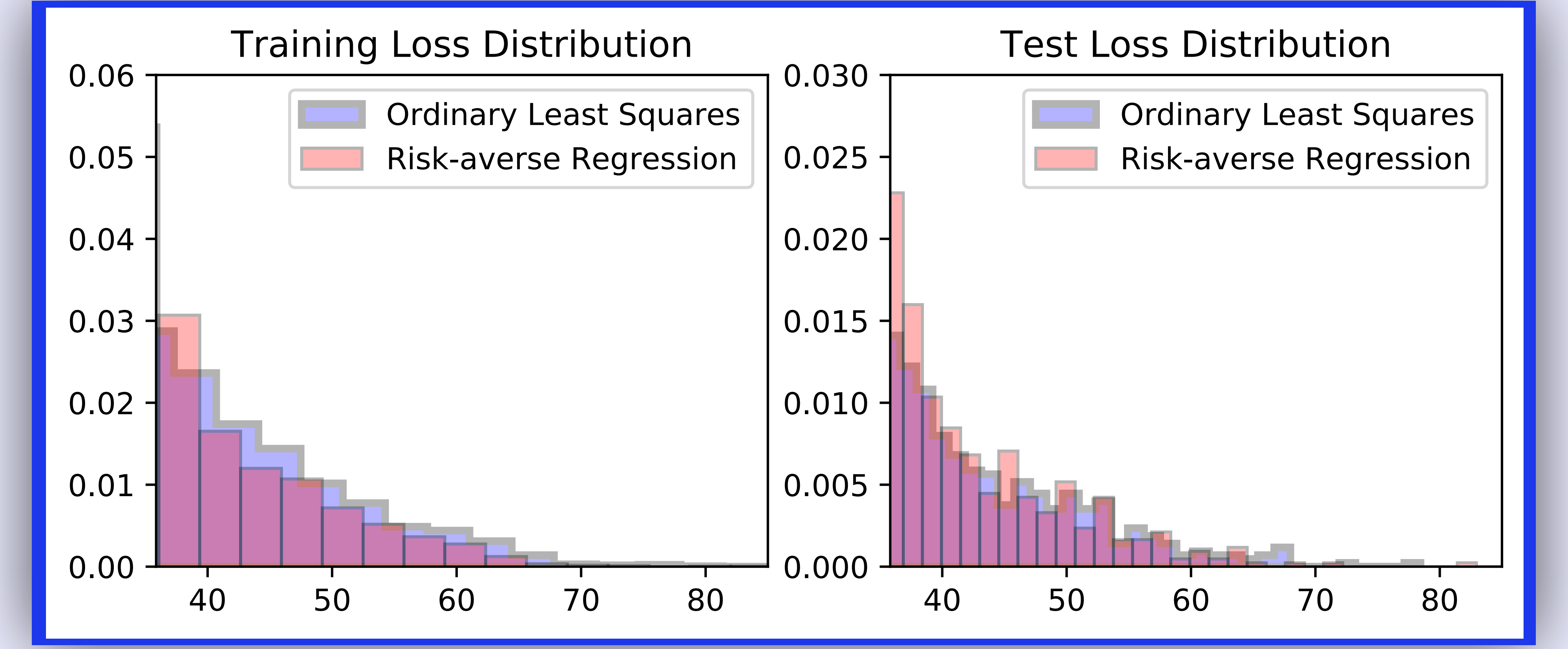
  β: *Bernoulli* (0.8); $\varepsilon_{\mathcal{N}}$: *Normal* (0,1); $\varepsilon_{\mathcal{L}}$: *Laplace* (10,1)
- **Squared Residuals** $r_i^2 = (y_i - w^\top x_i)^2$
- **Safety parameter** p=0.9



On the superconductivity dataset

- **Learning Task :** predict the critical temperature of a superconductor from 10 given features

Method	Mean	*p*-quantile of the Loss		
		p=0.90	p=0.95	p=0.99
$\mathbb{E}$	**16.5**	35.8	42.7	55.7
$\bar{Q}_p, p = 0.8$	17.4	**34.7**	**41.0**	53.8
$\bar{Q}_p, p = 0.9$	18.1	35.6	**41.0**	**53.6**
$\bar{Q}_p, p = 0.95$	18.9	36.5	41.4	**53.6**



1 Safety in Supervised ML

2 THE TOOLBOX SPQR

3 Behind SPQR

SUBGRADIENT ORACLE

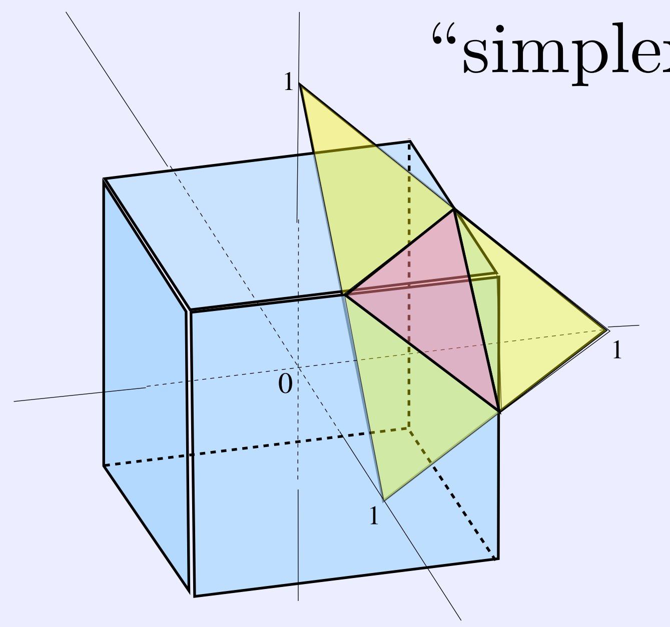
- **Dual Formulation of Superquantiles**

$$\bar{Q}_p(U) = \sup_{Q \in \mathcal{A}_p} \mathbb{E}_Q[U] \overset{(\star)}{=} \sup_{Q \in \mathcal{A}_p} \langle Q | U \rangle$$

$$\mathcal{A}_p = \left\{ Q \in \mathbb{R}^n, \underbrace{\sum_{i=1}^n q_i = 1}, \underbrace{0 \leq q_i \leq \frac{1}{n(1-p)}} \right\}$$

"simplex constraint"

"∞-norm constraint"



SUBGRADIENT ORACLE

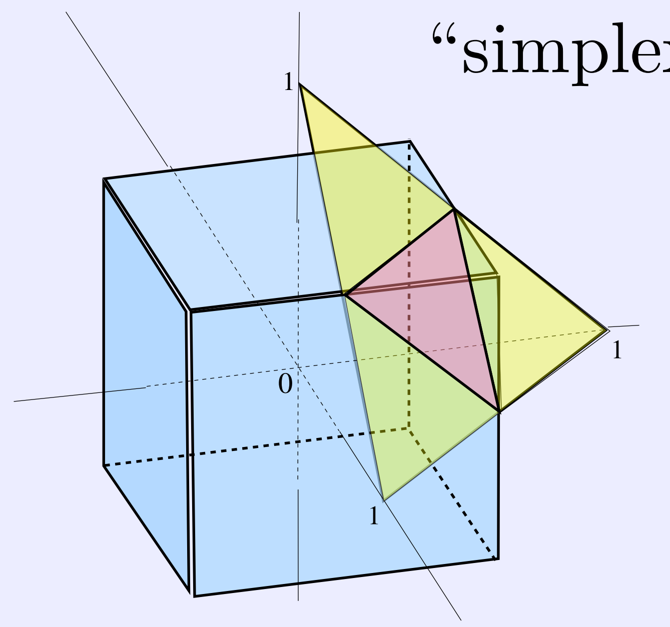
Dual Formulation of Superquantiles

$$\bar{Q}_p(U) = \sup_{Q \in \mathcal{A}_p} \mathbb{E}_Q[U] = \sup_{Q \in \mathcal{A}_p} \langle Q | U \rangle \quad (\star)$$

$$\mathcal{A}_p = \left\{ Q \in \mathbb{R}^n, \underbrace{\sum_{i=1}^n q_i = 1}, \underbrace{0 \le q_i \le \frac{1}{n(1-p)}} \right\}$$

"simplex constraint"

"∞-norm constraint"



Subgradient Formula

- Assuming $w \mapsto L(w, x_i, y_i)$ is convex

$$\partial(\bar{Q}_p \circ L)(w) = \left\{ \sum_{i=1}^n Q_i, \partial_w L(w, x_i, y_i), \underbrace{Q \in \operatorname{argmax}(\star)} \right\}$$

Not Reduced to a singleton!

- Computational complexity $\mathcal{O}(n)$

SMOOTHED GRADIENT ORACLE

Dual Formulation of Superquantiles

Strongly convex

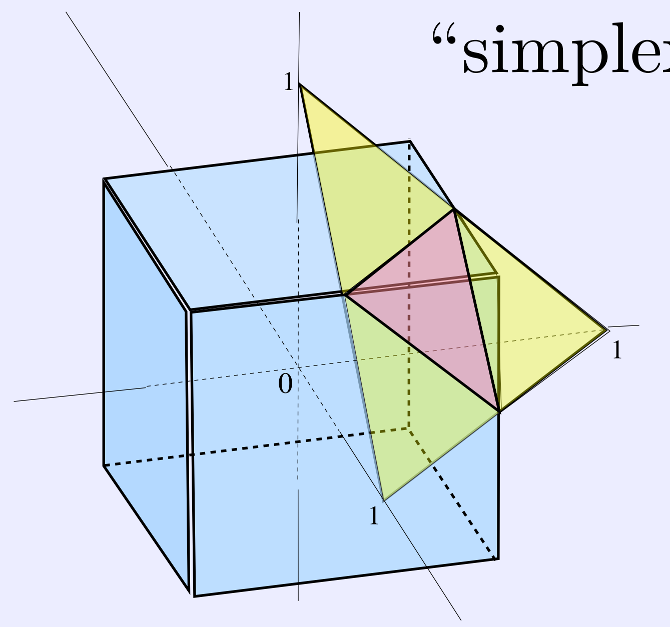
$$\bar{Q}_p(U) = \sup_{Q \in \mathcal{A}_p} \mathbb{E}_Q[U] \simeq \sup_{Q \in \mathcal{A}_p} \langle Q | U \rangle - \mu\, d(q)$$

Nesterov's Smoothing

$$\mathcal{A}_p = \left\{ Q \in \mathbb{R}^n, \sum_{i=1}^{n} q_i = 1, 0 \leq q_i \leq \frac{1}{n(1-p)} \right\}$$

"simplex constraint"

"∞-norm constraint"



SMOOTHED GRADIENT ORACLE

Dual Formulation of Superquantiles

Strongly convex

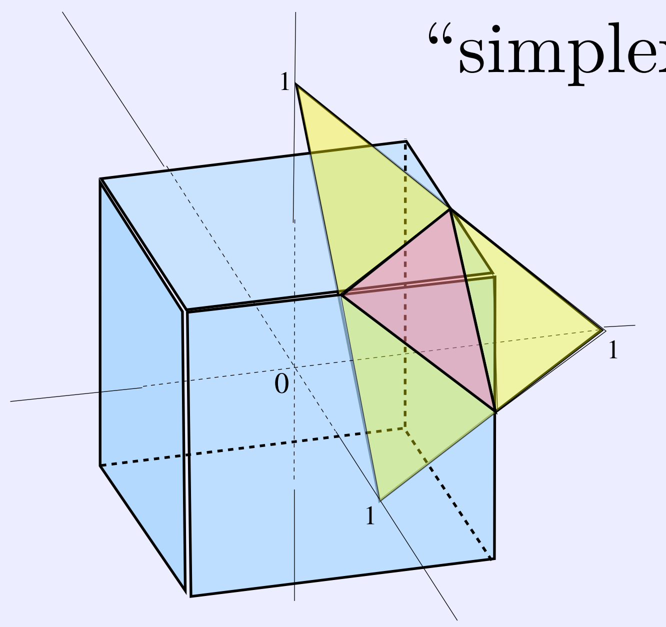
$$\bar{Q}_p(U) = \sup_{Q \in \mathcal{A}_p} \mathbb{E}_Q[U] \simeq \sup_{Q \in \mathcal{A}_p} \langle Q | U \rangle - \mu \, d(q)$$

Nesterov's Smoothing

$$\mathcal{A}_p = \left\{ Q \in \mathbb{R}^n, \sum_{i=1}^{n} q_i = 1, 0 \leq q_i \leq \frac{1}{n(1-p)} \right\}$$

"simplex constraint"

"∞-norm constraint"



Smoothing Procedure

- Based on Lagrangian Duality.
- Comes back to the computation of the p-quantile of $L(w, X, Y)$.
- Choice of the prox-function

$$d(q) = \left\| q - \frac{(1, \ldots, 1)^\top}{n} \right\|_2^2 \quad \text{(quadratic)}$$

$$d(q) = \sum_{i=1}^{n} q_i \log(n \, q_i) \quad \text{(entropic)}$$

CONCLUSION & PERSPECTIVES

CONCLUSION & PERSPECTIVES

- First-order oracle for Safe Supervised Machine Learning
- Smoothing with a fast computation procedure
- A Toolbox for effective minimization of superquantiles
  https://yassine-laguel.github.io/spqr/
- Potential Applications in Distributed Settings including Federated Learning

Feel free to ask questions : yassine.laguel@univ-grenoble-alpes.fr