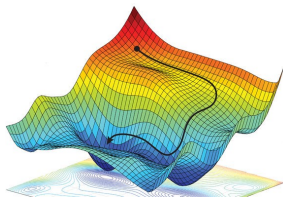
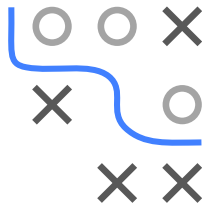


# Introduction to Machine Learning

## Advanced Risk Minimization

### Risk Minimization Basics



#### Learning goals

- Risk minimization and ERM recap
- Bayes optimal model, Bayes risk
- Bayes regret, estimation and approximation error
- Optimal constant model
- Consistency

# EMPIRICAL RISK MINIMIZATION

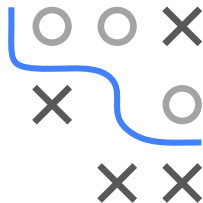
To learn a model, we usually do ERM:

$$\mathcal{R}_{\text{emp}}(f) = \sum_{i=1}^n L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right)$$

- observations  $(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{X} \times \mathcal{Y}$
- model  $f_{\mathcal{H}} : \mathcal{X} \rightarrow \mathbb{R}^g$ , from hypothesis space  $\mathcal{H}$ ; maps a feature vector to output score; often we omit  $\mathcal{H}$  in index
- loss  $L : \mathcal{Y} \times \mathbb{R}^g \rightarrow \mathbb{R}$ , measures error between label and prediction
- data generating process (DGP)  $\mathbb{P}_{xy}$ , we assume  $(\mathbf{x}^{(i)}, y^{(i)}) \stackrel{\text{i.i.d.}}{\sim} \mathbb{P}_{xy}$

Minimizing theoretical risk, so expected loss over DGP, is major goal:

$$\mathcal{R}(f) := \mathbb{E}_{xy}[L(y, f(\mathbf{x}))] = \int L(y, f(\mathbf{x})) d\mathbb{P}_{xy}$$

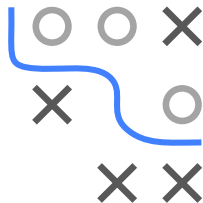


# TWO SHORT EXAMPLES

## Regression with linear model:

- Model:  $f(\mathbf{x}) = \boldsymbol{\theta}^\top \mathbf{x} + \theta_0$
- Squared loss:  $L(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$
- Hypothesis space:

$$\mathcal{H}_{\text{lin}} = \left\{ \mathbf{x} \mapsto \boldsymbol{\theta}^\top \mathbf{x} + \theta_0 : \boldsymbol{\theta} \in \mathbb{R}^d, \theta_0 \in \mathbb{R} \right\}$$



## Binary classification with shallow MLP:

- Model:  $f(\mathbf{x}) = \pi(\mathbf{x}) = \sigma(\mathbf{w}_2^\top \text{ReLU}(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + b_2)$
- Bernoulli / Log / Cross-Entropy loss:  
 $L(y, \pi(\mathbf{x})) = -(y \log(\pi(\mathbf{x})) + (1 - y) \log(1 - \pi(\mathbf{x})))$
- Hypothesis space:

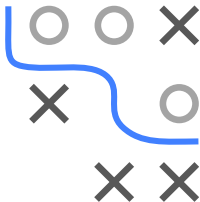
$$\mathcal{H}_{\text{MLP}} = \left\{ \mathbf{x} \mapsto \sigma(\mathbf{w}_2^\top \text{ReLU}(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + b_2) : \mathbf{W}_1 \in \mathbb{R}^{h \times d}, \mathbf{b}_1 \in \mathbb{R}^h, \mathbf{w}_2 \in \mathbb{R}^h, b_2 \in \mathbb{R} \right\}$$

# HYPOTHESIS SPACES AND PARAMETRIZATION

We often write  $\mathcal{R}(f)$ , but finding an optimal  $f$  is operationalized as finding optimal  $\theta \in \Theta$  among a family of parametrized curves:

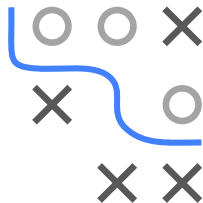
$$\mathcal{H} = \{f_{\theta} : f_{\theta} \text{ from functional family parametrized by } \theta\}$$

- Optimizing numeric vectors is more convenient than functions
- For some model classes, some parameters encode the same function (non-injective mapping, non-identifiability).  
We don't care here, now.



# OPTIMAL LOSS VALUES – M-ESTIMATORS

- Assume some RV  $z \sim Q, z \in \mathcal{Y}$  as target
- $z$  not the same as  $y$ , as we want to fiddle with its distribution
- We now consider  $\arg \min_c \mathbb{E}_{z \sim Q}[L(z, c)]$   
What is the constant that approximates  $z$  with minimal loss?



## 3 cases for $Q$ :

- $Q = P_y$ , distribution of labels  $y$ , marginal of  $\mathbb{P}_{xy}$   
optimal theoretical constant prediction
- $Q = P_n$ , the empirical product distribution for data  $y^{(1)}, \dots, y^{(n)}$   
optimal empirical constant prediction
- $Q = P_{y|\mathbf{x}=\tilde{\mathbf{x}}}$ , conditional label distribution at point  $\mathbf{x} = \tilde{\mathbf{x}}$   
Bayes optimal pointwise prediction / theoretical risk minimizer

# OPTIMAL UNCONDITIONAL VALUES

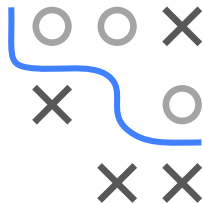
- Associating such a

$$c = \arg \min_{c \in \mathbb{R}} \mathbb{E}_{z \sim Q}[L(z, c)]$$

with a distribution is called a “statistical functional”

- Such a loss-minimizing version, and especially its empirical version below, is called an **M-estimator**
- “M” can be read as “max-likelihood type”, or “minimizing”, I prefer the latter
- If we look at the empirical counterpart, with the empirical distribution, this is the so-called “plug-in” estimator

$$\arg \min_{c \in \mathbb{R}} \sum_{i=1}^n L(y^{(i)}, c)$$

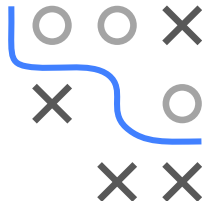
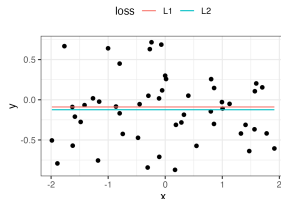


# OPTIMAL CONSTANT MODEL

- Goal: loss optimal, constant baseline predictor
- “constant”: featureless ML model, always predicts same value
- “baseline”: more complex model has to be better
- Also useful as optimal intercept

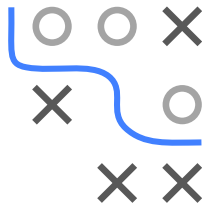
$$f_c^* = \arg \min_{c \in \mathbb{R}} \mathbb{E}_{xy} [L(y, c)] = \arg \min_{c \in \mathbb{R}} \mathbb{E}_y [L(y, c)]$$

- Estimation via ERM:  $\hat{f}_c = \arg \min_{c \in \mathbb{R}} \sum_{i=1}^n L(y^{(i)}, c)$



# RISK MINIMIZER

- Assume, hypothesis space  $\mathcal{H} = \mathcal{H}_{\text{all}}$  is unrestricted; contains any measurable  $f : \mathcal{X} \rightarrow \mathbb{R}^g$
- We know  $\mathbb{P}_{xy}$
- $f$  with minimal risk across  $\mathcal{H}_{\text{all}}$  is called **risk minimizer**, **population minimizer** or **Bayes optimal model**



$$\begin{aligned} f_{\mathcal{H}_{\text{all}}}^* &= \arg \min_{f \in \mathcal{H}_{\text{all}}} \mathcal{R}(f) = \arg \min_{f \in \mathcal{H}_{\text{all}}} \mathbb{E}_{xy} [L(y, f(\mathbf{x}))] \\ &= \arg \min_{f \in \mathcal{H}_{\text{all}}} \int L(y, f(\mathbf{x})) d\mathbb{P}_{xy} \end{aligned}$$

- The resulting risk is called **Bayes risk**:  $\mathcal{R}^* = \mathcal{R}(f_{\mathcal{H}_{\text{all}}}^*)$
- **Risk minimizer within**  $\mathcal{H} \subset \mathcal{H}_{\text{all}}$  is  $f_{\mathcal{H}}^* = \arg \min_{f \in \mathcal{H}} \mathcal{R}(f)$



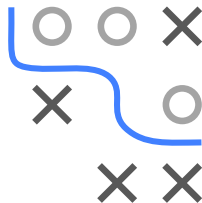
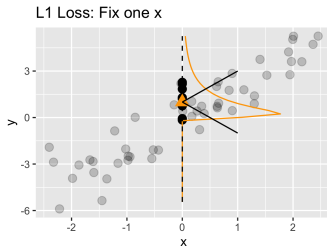
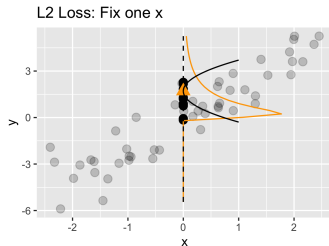
# OPTIMAL POINT-WISE PREDICTIONS

- To derive the RM, by law of total expectation

$$\mathcal{R}(f) = \mathbb{E}_{xy} [L(y, f(\mathbf{x}))] = \mathbb{E}_x [\mathbb{E}_{y|x} [L(y, f(\mathbf{x})) \mid \mathbf{x}]]$$

- We can choose  $f(\mathbf{x})$  as we want from  $\mathcal{H}_{\text{all}}$
- Hence, for fixed feature vector  $\tilde{\mathbf{x}}$  we can select **any** value  $c$  to predict. So we construct the **point-wise optimizer**

$$f^*(\tilde{\mathbf{x}}) = \arg \min_c \mathbb{E}_{y|x} [L(y, c) \mid \mathbf{x} = \tilde{\mathbf{x}}]$$



# THEORETICAL AND EMPIRICAL RISK

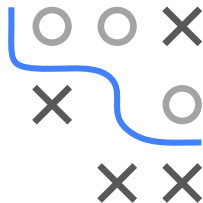
- Bayes risk minimizer is mainly a theoretical tool
- In practice, need to restrict  $\mathcal{H}$  for efficient search
- We don't normally know  $\mathbb{P}_{xy}$ . Instead, use ERM.

$$\hat{f}_{\mathcal{H}} = \arg \min_{f \in \mathcal{H}} \mathcal{R}_{\text{emp}}(f) = \arg \min_{f \in \mathcal{H}} \sum_{i=1}^n L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right)$$

- Due to **law of large numbers**, empirical risk for fixed model converges to true risk, so consistent estimator

$$\bar{\mathcal{R}}_{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^n L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right) \xrightarrow{n \rightarrow \infty} \mathcal{R}(f)$$

- Still, that does not imply that the selected ERM minimizer converges to  $f^*$ , due to overfitting or lack of uniform convergence
- Would need more assumptions / math. machinery for this, will not pursue this here



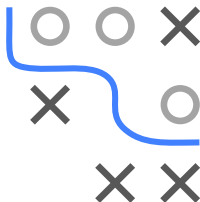
# ESTIMATION AND APPROXIMATION ERROR

- Goal: Train model  $\hat{f}_{\mathcal{H}}$  with risk  $\mathcal{R}(\hat{f}_{\mathcal{H}})$  close to Bayes risk  $\mathcal{R}^*$
- Minimize **Bayes regret** or **excess risk**

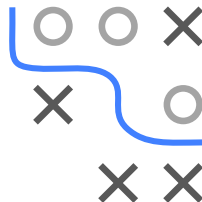
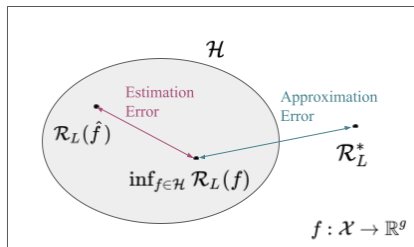
$$\mathcal{R}(\hat{f}_{\mathcal{H}}) - \mathcal{R}^*$$

- Decompose:

$$\begin{aligned}\mathcal{R}(\hat{f}_{\mathcal{H}}) - \mathcal{R}^* &= \underbrace{\left[ \mathcal{R}(\hat{f}_{\mathcal{H}}) - \inf_{f \in \mathcal{H}} \mathcal{R}(f) \right]}_{\text{estimation error}} + \underbrace{\left[ \inf_{f \in \mathcal{H}} \mathcal{R}(f) - \mathcal{R}^* \right]}_{\text{approximation error}} \\ &= \left[ \mathcal{R}(\hat{f}_{\mathcal{H}}) - \mathcal{R}(f_{\mathcal{H}}^*) \right] + \left[ \mathcal{R}(f_{\mathcal{H}}^*) - \mathcal{R}(f_{\mathcal{H}_{\text{all}}}^*) \right]\end{aligned}$$



# ESTIMATION AND APPROXIMATION ERROR



$$\mathcal{R}(\hat{f}_{\mathcal{H}}) - \mathcal{R}^* = \underbrace{\left[ \mathcal{R}(\hat{f}_{\mathcal{H}}) - \inf_{f \in \mathcal{H}} \mathcal{R}(f) \right]}_{\text{estimation error}} + \underbrace{\left[ \inf_{f \in \mathcal{H}} \mathcal{R}(f) - \mathcal{R}^* \right]}_{\text{approximation error}}$$

- **Estimation error:** We fit  $\hat{f}_{\mathcal{H}}$  via ERM on finite data, so we don't find best  $f \in \mathcal{H}$
- **Approximation error:**  $\mathcal{H}$  will often not contain Bayes optimal  $f^*$

## (UNIVERSALLY) CONSISTENT LEARNERS ▶ Stone 1977

**Consistency** is an asymptotic property of a learning algorithm, which ensures the algorithm returns **the correct model** when given **unlimited data**.

Let  $\mathcal{I} : \mathbb{D} \rightarrow \mathcal{H}$  be a learning algorithm that takes a training set  $\mathcal{D}_{\text{train}} \sim \mathbb{P}_{\mathbf{xy}}$  of size  $n_{\text{train}}$  and estimates a model  $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}^g$ .

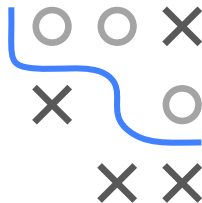
The learning method  $\mathcal{I}$  is said to be **consistent** w.r.t. a certain distribution  $\mathbb{P}_{xy}$  if the risk of the estimated model  $\hat{f}$  converges in probability ( " $\xrightarrow{P}$ " ) to the Bayes risk  $\mathcal{R}^*$  when  $n_{\text{train}}$  goes to  $\infty$ :

$$\mathcal{R}(\mathcal{I}(\mathcal{D}_{\text{train}})) \xrightarrow{p} \mathcal{R}^* \quad \text{for } n_{\text{train}} \rightarrow \infty$$

# (UNIVERSALLY) CONSISTENT LEARNERS

► Stone 1977

Consistency is defined w.r.t. a particular distribution  $\mathbb{P}_{xy}$ . But since we usually don't know  $\mathbb{P}_{xy}$ , consistency does not offer much help to choose an algorithm for a specific task.



More interesting is the stronger concept of **universal consistency**: An algorithm is universally consistent if it is consistent for **any** distribution.

In Stone's famous consistency theorem (1977), the universal consistency of a weighted average estimator such as KNN was proven. Many other ML models have since then been proven to be universally consistent (SVMs, ANNs, etc.).