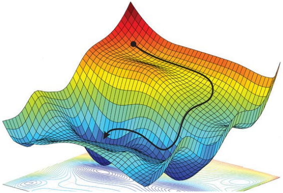
Introduction to Machine Learning

Advanced Risk Minimization

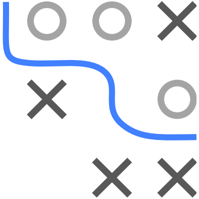
Risk Minimization Basics



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:

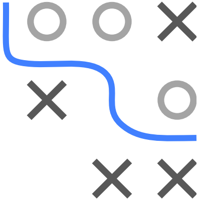
$$\mathcal{R}_{\text{emp}}(f) = \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right)$$

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

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

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

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:

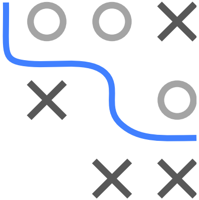
$$\mathcal{H}_{\text{lin}} = \left\{ \mathbf{x} \mapsto \boldsymbol{\theta}^\top \mathbf{x} + \theta_0 : \boldsymbol{\theta} \in \mathbb{R}^d, \theta_0 \in \mathbb{R} \right\}$$

Binary classification with shallow MLP:

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

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

HYPOTHESIS SPACES AND PARAMETRIZATION

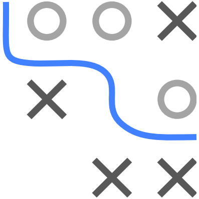


We often write $\mathcal{R}(f)$, but finding an optimal f is operationalized as finding optimal $\boldsymbol{\theta} \in \Theta$ among a family of parametrized curves:

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

- 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

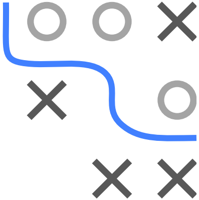


- Assume some RV $z \sim Q, z \in \mathcal{Y}$ as target
- z not the same as y, as we want to fiddle with its distribution
- We now consider $\arg\min_c \mathbb{E}_{z \sim Q}[L(z, c)]$
  What is the constant that approximates z with minimal loss?

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 $y^{(1)}, \ldots, y^{(n)}$
  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

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

  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

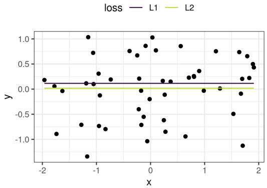
$$\operatorname*{arg\,min}_{c \in \mathbb{R}} \sum_{i=1}^{n} L(y^{(i)}, c)$$

OPTIMAL CONSTANT MODEL

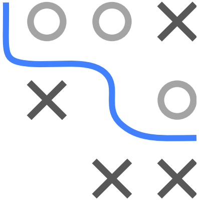
- Goal: loss optimal, constant baseline predictor
- “constant”: featureless ML model, always predicts same value
- “baseline”: more complex model has to be better
- Also useful as optimal intercept

$$f_c^* = \underset{c \in \mathbb{R}}{\arg\min}\, \mathbb{E}_{xy}\left[L(y, c)\right] = \underset{c \in \mathbb{R}}{\arg\min}\, \mathbb{E}_{y}\left[L(y, c)\right]$$

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



RISK MINIMIZER



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

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

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

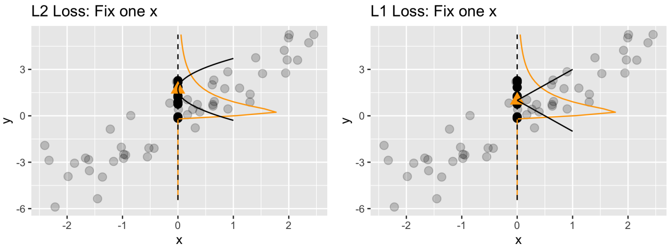
OPTIMAL POINT-WISE PREDICTIONS

- To derive the RM, by law of total expectation

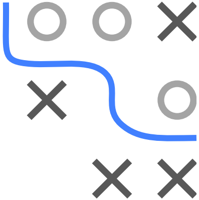
$$\mathcal{R}(f) = \mathbb{E}_{xy}\left[L\left(y, f(\mathbf{x})\right)\right] = \mathbb{E}_x\left[\mathbb{E}_{y|x}\left[L\left(y, f(\mathbf{x})\right) \mid \mathbf{x}\right]\right]$$

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

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



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.

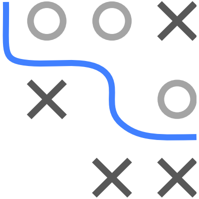
$$\hat{f}_{\mathcal{H}} = \underset{f \in \mathcal{H}}{\arg\min}\, \mathcal{R}_{\text{emp}}(f) = \underset{f \in \mathcal{H}}{\arg\min} \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right)$$

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

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

- 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



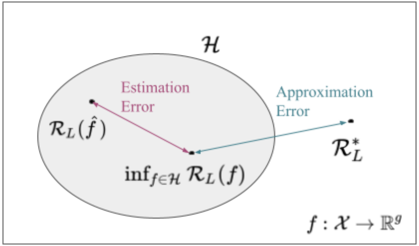
- Goal: Train model $\hat{f}_{\mathcal{H}}$ with risk $\mathcal{R}\left(\hat{f}_{\mathcal{H}}\right)$ close to Bayes risk $\mathcal{R}^*$
- Minimize **Bayes regret** or **excess risk**

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

- Decompose:

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

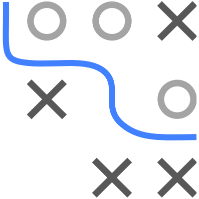
ESTIMATION AND APPROXIMATION ERROR



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

- **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



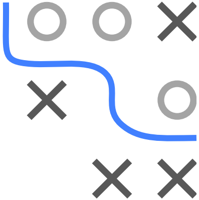
Consistency is an asymptotic property of a learning algorithm, which ensures the algorithm returns **the correct model** when given **unlimited data**.

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

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_{train} goes to ∞:

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

(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.).