Chapter 2
Introduction: Advanced Machine Learning

Abstract This chapter begins with an introduction to the traditional empirical risk minimization (ERM) framework, using standard label prediction tasks to illustrate its three core components: loss functions, optimization algorithms, and generalization analysis. We then explore advanced learning techniques including distributionally robust optimization (DRO) and group DRO that aim to enhance model robustness under distribution shifts. Building on this foundation, we introduce the empirical X-risk minimization (EXM) paradigm and discuss its applications in modern machine learning. Finally, we present the concept of data prediction for discriminative learning in foundation models. The goals of this chapter are threefold: (i) to provide a cohesive view of how discriminative principles inform objective function design; (ii) to highlight the role of optimization tools for objective design and model training; and (iii) to motivate the need for compositional optimization frameworks.

models fade, but principles endure!

Contents

2.1 Empirical Risk Minimization

What is Machine Learning (ML)?

In 1959, Arthur Samuel, a pioneer in the field of ML, defined Machine Learning as the *"field of study that gives computers the ability to learn without being explicitly programmed"* .

Nowadays, machine learning has become the foundation of AI. The essence of machine learning is to learn a model by optimizing an objective function on training data, with the goal of achieving strong generalization to unseen data. This relationship is captured by the formula:

$$\textbf{Machine Learning} = \textbf{Objective} \quad + \quad \textbf{Algorithm} \quad + \quad \textbf{Generalization}.$$

Optimization plays a fundamental role in machine learning, as it underpins (1) the formulation of objective functions, (2) the development of optimization algorithms, and (3) the analysis of generalization error of learned models. Below, we will use the traditional label prediction problem to illustrate the three components.

2.1.1 Discriminative Label Prediction

In supervised learning, the primary objective is often to learn a predictive model from a given set of supervised training data. Let us consider a classical label prediction problem. Denote by $(\mathbf{x}, y)$ a data-label pair, where $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^{d_0}$ denotes the input feature vector, and $y \in \mathcal{Y} = \{1, \ldots, K\}$ is the corresponding label. The goal is to learn a predictive model parameterized by $\mathbf{w} \in \mathcal{W} \subseteq \mathbb{R}^d$ (e.g., a deep neural network), which induces a scoring function $h(\mathbf{w}; \cdot) : \mathcal{X} \to \mathbb{R}^K$. Conceptually, the model can be expressed as $h(\mathbf{w}; \mathbf{x}) = W h_0(\mathbf{w}; \mathbf{x})$, where $h_0(\mathbf{w}; \cdot) : \mathcal{X} \to \mathbb{R}^{d_1}$ is the feature extraction component, and $W \in \mathbb{R}^{K \times d_1}$ is the classification head corresponding to the K classes.

A classical framework for learning such a model is the well-known empirical risk minimization (ERM), which minimizes the empirical risk over the training dataset. To this end, a pointwise loss function $\ell(h(\mathbf{w}; \mathbf{x}), y)$ is defined to measure the discrepancy between the model's prediction $h(\mathbf{w}; \mathbf{x})$ and the true label y. Given a training dataset $\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, the ERM problem is formulated as:

$$\min_{\mathbf{w} \in \mathcal{W}} \mathcal{R}_{\mathcal{S}}(\mathbf{w}) := \frac{1}{n} \sum_{i=1}^{n} \ell(h(\mathbf{w}; \mathbf{x}_i), y_i). \tag{2.1}$$

2.1.2 Discriminative Loss Functions

A major element of ERM is the design of the loss function. A common strategy of designing a loss function for label prediction is through a discriminative approach. Below, we introduce several discriminative loss functions.

Logistic Loss

A parameterized probabilistic model is defined to represent the probability of a class label for a given data point as

$$\Pr(y|\mathbf{x};\mathbf{w}) = \frac{\exp([h(\mathbf{w};\mathbf{x})]_y)}{\sum_{l=1}^{K}\exp([h(\mathbf{w};\mathbf{x})]_l)}, \tag{2.2}$$

where $[\cdot]_k$ denotes the k-th element of a vector. The associated loss is derived from the negative log-likelihood, resulting in the multi-class logistic loss, also known as the cross-entropy (CE) loss:

$$\ell(h(\mathbf{w};\mathbf{x}),y) = -\log\frac{\exp([h(\mathbf{w};\mathbf{x})]_y)}{\sum_{l=1}^{K}\exp([h(\mathbf{w};\mathbf{x})]_l)}. \tag{2.3}$$

The resulting method by ERM is commonly referred to as logistic regression. For binary classification, this loss becomes the binary logistic loss $\ell(h(\mathbf{w};\mathbf{x}),y) = \log(1+\exp(-yh(\mathbf{w};\mathbf{x})))$, where $h(\mathbf{w};\cdot)\in\mathbb{R}$ and $y\in\{1,-1\}$.

Max-Margin Loss

The max-margin loss, introduced by Crammer and Singer and commonly referred to as the Crammer-Singer (CS) loss (Crammer and Singer, 2002), is defined as:

$$\ell(h(\mathbf{w};\mathbf{x}),y) = \max\left(0, \max_{k\neq y}\left(c_{k,y} + [h(\mathbf{w};\mathbf{x})]_k - [h(\mathbf{w};\mathbf{x})]_y\right)\right), \tag{2.4}$$

where $c_{k,y}>0$ is a margin parameter. This loss seeks to ensure that the prediction score for the ground-truth label, $[h(\mathbf{w};\mathbf{x})]_y$, exceeds the scores of other class labels, $[h(\mathbf{w};\mathbf{x})]_k$ for $k\neq y$, by at least the margin $c_{k,y}$. This method is also known as the multi-class support vector machine. For binary classification, it reduces to the standard hinge loss $\ell(h(\mathbf{w};\mathbf{x}),\mathbf{y}) = \max(0, 1-yh(\mathbf{w};\mathbf{x}))$ for $h(\mathbf{w};\cdot)\in\mathbb{R}$ and $y\in\{1,-1\}$.

Label Distributionally Robust (LDR) Loss

Both the CS loss and the CE loss has it strengths and limitations. For example, the CS loss with the margin parameters is more flexible in controlling the discrimination between classes, while it is not consistent and non-smooth in terms of the prediction scores. The CE loss is smooth and consistent but lacks robustness to noise in class labels.

Consistency of a surrogate loss function

The consistency measures whether minimizing a surrogate loss with an infinite number of data also minimizes the Bayes error. More formally, a surrogate loss $\ell(h(\mathbf{x}), y)$ is said to be consistent if any sequence of measurable functions $h^{(n)}$ it holds

$$\mathcal{R}(h^{(n)}) \rightarrow \inf_{h\in\mathcal{H}} \mathcal{R}(h) \Rightarrow \mathcal{R}_{0-1}(h^{(n)}) \rightarrow \inf_{h\in\mathcal{H}} \mathcal{R}_{0-1}(h),$$

where $\mathcal{R}(h) = \mathbb{E}_{\mathbf{x},y}[\ell(h(\mathbf{x}), y)]$ is the expected risk, $\mathcal{R}_{0-1}(h) = \mathbb{E}_{\mathbf{x},y}[\mathbb{I}(y \neq h(\mathbf{x}))]$ is the Bayes error, and $\mathcal{H}$ is the set of any measurable functions.

In fact, the strengths and limitations of both the CE and CS losses can be better understood within a broader family known as the label-distributionally robust (LDR) loss:

$$\ell_\tau(h(\mathbf{w};\mathbf{x}), y) = \max_{\mathbf{p}\in\Delta_K} \sum_{k=1}^{K} p_k\big([h(\mathbf{w};\mathbf{x})]_k - [h(\mathbf{w};\mathbf{x})]_y + c_{k,y}\big) - \tau \sum_{k=1}^{K} p_k \log(p_k K), \tag{2.5}$$

where $\tau > 0$ is a hyperparameter, $c_{y,y} = 0$, $\mathbf{p} \in \mathbb{R}^K$ is referred to as the label distributional weight vector, and $\Delta_K = \{\mathbf{p} \in \mathbb{R}^K : p_k \geq 0, \sum_{k=1}^K p_k = 1\}$ is a simplex.

It is clear that the LDR loss is defined by solving an optimization problem. Indeed, the above optimization problem follows the distributionally robust optimization (DRO) principle, which is widely used at the level of data as discussed in section 2.2. By treating 'label' as a kind of data, we can unify the LDR loss with other losses discussed later in Section 2.4.

A closed-form solution for $\mathbf{p}$ can be derived using the KKT conditions (cf. Example 1.16), making the LDR loss equivalent to:

$$\ell_\tau(h(\mathbf{w};\mathbf{x}), y) = \tau \log\left[\frac{1}{K}\sum_{k=1}^{K} \exp\left(\frac{[h(\mathbf{w};\mathbf{x})]_k + c_{k,y} - [h(\mathbf{w};\mathbf{x})]_y}{\tau}\right)\right]. \tag{2.6}$$

From the perspective of DRO, we can define a more general family of LDR losses using different regularization functions on $\mathbf{p}$ and constrained domains Ω:

$$\bar{\ell}_\tau(h(\mathbf{w};\mathbf{x}), y) = \max_{\mathbf{p}\in\Omega} \sum_{k=1}^{K} p_k\left([h(\mathbf{w};\mathbf{x})]_k - [h(\mathbf{w};\mathbf{x})]_y + c_{k,y}\right) - \tau R(\mathbf{p}). \tag{2.7}$$

where $\Omega \subseteq \Delta_K$ and $R(\mathbf{p})$ is a strongly convex regularizer.

Why it matters:

- The LDR loss (2.6) unifies both the CS and CE losses as special cases. Specifically, the CE loss is recovered when $\tau = 1$ and $c_{k,y} = 0$ for all k, while the CS loss corresponds to the case $\tau = 0$.
  Moreover, the LDR loss encompasses the Label-Distribution-Aware Margin (LDAM) loss (Cao et al., 2019) when $\tau = 1$ and $c_{k,y} = c_y \propto 1/n_y^{1/4}$ for $k \neq y$, where n_y denotes the number of samples in class y:

$$\ell_{\text{LDAM}}(h(\mathbf{w};\mathbf{x}), y) = -\log\left(\frac{\exp\left([h(\mathbf{w};\mathbf{x})]_y - \frac{C}{n_y^{1/4}}\right)}{\exp\left([h(\mathbf{w};\mathbf{x})]_y - \frac{C}{n_y^{1/4}}\right) + \sum_{l\neq y}\exp([h(\mathbf{w};\mathbf{x})]_l)}\right),$$

  where C is a constant. For imbalanced datasets, this assigns larger margins c_y to minority classes, making it more suitable for handling class imbalance.
- The LDR loss provides insights into the strengths and limitations of CE and CS losses. The regularizer $R(\mathbf{p}) = \sum_{k=1}^K p_k \log(p_k K)$ is strongly convex in $\mathbf{p}$, which implies smoothness of the loss in terms of prediction scores due to the duality between smoothness and strong convexity (Lemma 1.9). This strong convexity also contributes to the statistical consistency of the loss (Zhu et al., 2023b). In contrast, the CS loss with $\tau = 0$ lacks this property, and hence suffer from non-smoothness and inconsistency.
- The LDR loss framework enables the design of new losses that are robust to label noise. For instance, when $\tau \to \infty$, the LDR loss reduces to:

$$\ell_\infty(\mathbf{w};\mathbf{x}, y) = \frac{1}{K}\sum_{k=1}^{K}\left([h(\mathbf{w};\mathbf{x})]_k - [h(\mathbf{w};\mathbf{x})]_y + c_{k,y}\right).$$

  A remarkable property of this loss is its symmetry: $\sum_{y=1}^K \ell_\infty(\mathbf{w};\mathbf{x}, y)$ is constant. This symmetry serves as a sufficient condition for robustness to uniform label noise (Ghosh et al., 2017). However, by treating all negative labels equally, it may limit the model's ability to focus on hard negative labels and potentially slow down the learning process. In practice, it is better to tune τ if there is label noise.

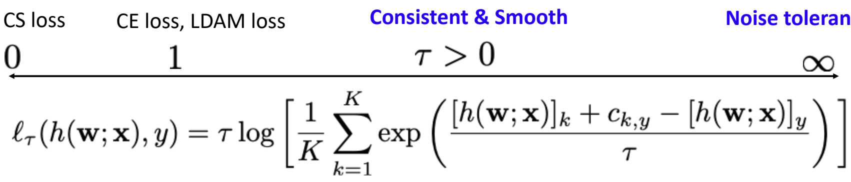
CS loss
CE loss, LDAM loss
Consistent & Smooth
Noise tolerant
0
1
$\tau > 0$
∞
$\ell_\tau(h(\mathbf{w};\mathbf{x}), y) = \tau \log \left[\frac{1}{K} \sum_{k=1}^{K} \exp\left(\frac{[h(\mathbf{w};\mathbf{x})]_k + c_{k,y} - [h(\mathbf{w};\mathbf{x})]_y}{\tau} \right) \right]$

Fig. 2.1: The LDR loss and its special cases by varying τ.

In conclusion, the LDR loss offers flexibility in achieving three desirable properties: max-margin, consistency, and symmetry. In practice, when tuning $\tau \in (0, \infty)$, it may be beneficial to normalize the prediction scores $h(\mathbf{w};\mathbf{x})$.

Critical: It is worth noting that all the discussed losses are discriminative in nature, aiming to increase the score $[h(\mathbf{w};\mathbf{x})]_y$ of the true label while decreasing the scores $[h(\mathbf{w};\mathbf{x})]_k$ of the negative labels ($k \neq y$).

2.1.3 Need of Optimization Algorithms

To address the ERM problem in the context of large-scale data (i.e., a substantial number of training examples), first-order stochastic algorithms are commonly employed. These include stochastic gradient descent (SGD), stochastic momentum methods, and adaptive gradient methods. For instance, the update rule of classical SGD for solving (2.1) with $\mathcal{W} = \mathbb{R}^d$ is given by:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \frac{1}{|\mathcal{B}_t|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{B}_t} \nabla \ell(h(\mathbf{w}_t; \mathbf{x}_i), y_i), \quad t = 1, \dots, T, \tag{2.8}$$

where $\eta_t \geq 0$ is the learning rate (or step size), and $\mathcal{B}_t$ denotes a random mini-batch data sampled from the full dataset. The concern of designing an optimization algorithm is how fast the algorithm can converge to a (near) optimal solution. We will discuss the design and analysis of classical stochastic optimization algorithms in Chapter 3.

Critical: A critical assumption in conventional stochastic optimization algorithms such as SGD is that the gradient $\nabla \ell(h(\mathbf{w};\mathbf{x}_i), y_i)$ of each individual loss, can be easily computed. This assumption will fail for the logistic loss when the number of classes K is gigantic, e.g. millions or even billions. This will be addressed in this book.

2.1.4 Generalization Analysis

To study the generalization of a model learned by solving ERM, we usually consider the expected risk defined as

$$\mathcal{R}(\mathbf{w}) = \mathbb{E}_{\mathbf{x},y\sim\mathbb{P}}[\ell(h(\mathbf{w};\mathbf{x}),y)]. \tag{2.9}$$

Let $\mathbf{w} = \mathcal{A}(\mathcal{S};\zeta)$ denote a learned model by a randomized algorithm $\mathcal{A}$ for solving ERM that depend on random variables ζ. A standard measure of generalization is given by the **excess risk** defined as $\mathcal{R}(\mathbf{w}) - \mathcal{R}(\mathbf{w}_*)$, where $\mathbf{w}_* \in \arg\min_{\mathbf{w}\in\mathcal{W}} \mathcal{R}(\mathbf{w})$. The following lemma decomposes the excess risk into the optimization error and the generalization error.

Lemma 2.1 *For a learned model* $\mathbf{w} = \mathcal{A}(\mathcal{S};\zeta)$*, we have*

$$\mathcal{R}(\mathbf{w}) - \mathcal{R}(\mathbf{w}_*) \leq \underbrace{\max(\mathcal{R}(\mathbf{w}) - \mathcal{R}_{\mathcal{S}}(\mathbf{w}), \mathcal{R}_{\mathcal{S}}(\mathbf{w}_*) - \mathcal{R}(\mathbf{w}_*))}_{\textit{generalization error}} + \underbrace{\mathcal{R}_{\mathcal{S}}(\mathbf{w}) - \min_{\mathbf{u}\in\mathcal{W}} \mathcal{R}_{\mathcal{S}}(\mathbf{u})}_{\textit{optimization error}},$$

and

$$\mathbb{E}_{\mathcal{S},\zeta}[\mathcal{R}(\mathbf{w}) - \mathcal{R}(\mathbf{w}_*)] = \mathbb{E}_{\mathcal{S},\zeta}[\mathcal{R}(\mathbf{w}) - \mathcal{R}_{\mathcal{S}}(\mathbf{w})] + \mathbb{E}_{\mathcal{S},\zeta}[\mathcal{R}_{\mathcal{S}}(\mathbf{w}) - \min_{\mathbf{u}\in\mathcal{W}} \mathcal{R}_{\mathcal{S}}(\mathbf{u})].$$

Proof.

$$\begin{aligned}\mathcal{R}(\mathbf{w}) - \mathcal{R}(\mathbf{w}_*) &= \mathcal{R}(\mathbf{w}) - \mathcal{R}_{\mathcal{S}}(\mathbf{w}) + \mathcal{R}_{\mathcal{S}}(\mathbf{w}) - \min_{\mathbf{u}\in\mathcal{W}} \mathcal{R}_{\mathcal{S}}(\mathbf{u}) + \min_{\mathbf{u}\in\mathcal{W}} \mathcal{R}_{\mathcal{S}}(\mathbf{u}) - \mathcal{R}(\mathbf{w}_*)\\ &\leq \mathcal{R}(\mathbf{w}) - \mathcal{R}_{\mathcal{S}}(\mathbf{w}) + \mathcal{R}_{\mathcal{S}}(\mathbf{w}) - \min_{\mathbf{u}\in\mathcal{W}} \mathcal{R}_{\mathcal{S}}(\mathbf{u}) + \mathcal{R}_{\mathcal{S}}(\mathbf{w}_*) - \mathcal{R}(\mathbf{w}_*).\end{aligned}$$

This proves the first inequality. By taking expectation over $\mathcal{S}$, ζ, we finish the second inequality. □

Why it matters:

The excess risk can be decomposed into two components: the optimization error, given by $\mathcal{R}_{\mathcal{S}}(\mathbf{w}) - \min_{\mathbf{u}\in\mathcal{W}} \mathcal{R}_{\mathcal{S}}(\mathbf{u})$, and the generalization error which captures the difference between the expected risk and the empirical risk.
The generalization error can be bounded uniformly by $\sup_{\mathbf{w}\in\mathcal{W}} |\mathcal{R}(\mathbf{w}) - \mathcal{R}_{\mathcal{S}}(\mathbf{w})|$, which generally decreases as the training data size $|\mathcal{S}|$ increases. Bounding the (expected) optimization error is a central focus of this book, approached through the analysis of stochastic optimization algorithms. A brief discussion of the literature on generalization error analysis will be provided at the end of this chapter.

2.2 Robust Optimization

In this section, we introduce advanced machine learning methods based on the principle of robust optimization. Robust optimization is a framework designed to address uncertainty in data. It ensures that the solutions perform well even under worst-case scenarios of data within a specified set of uncertainties.

2.2.1 Distributionally Robust Optimization

Minimizing the average empirical risk often fails to yield a robust model in practice. For instance, the resulting model may perform poorly on minority data (e.g., patients with rare diseases) because the optimization predominantly focuses on majority class data.

> **Critical:** Empirical data may not fully represent the underlying data distribution, leading to generalization issues.

To address these challenges, distributionally robust optimization (DRO) has been extensively studied in machine learning as a means to improve robustness and generalization.

The core idea of DRO is to minimize a robust objective defined over the worst-case distribution of data, perturbed from the empirical distribution. Let us define a set of distributional weights, $\mathbf{p} = (p_1, \ldots, p_n) \in \Delta_n$, where $\Delta_n = \{\mathbf{p} \in \mathbb{R}^n : \sum_{i=1}^n p_i = 1,\ p_i \geq 0\}$, with each element p_i associated with a training sample $\mathbf{x}_i$.

Definition 2.1 (ϕ-divergence) Let $\phi(t) : \mathbb{R}^+ \to \mathbb{R}$ is a proper closed convex function and has a minimum value zero that is attained at $t = 1$. The ϕ-divergence is defined as:

$$D_\phi(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^n q_i \phi(p_i/q_i). \tag{2.10}$$

ϕ-divergence measures the discrepancy between two distributions $\mathbf{p}$ and $\mathbf{q}$ using the function ϕ. We present two common formulations of DRO based on the ϕ-divergence: regularized DRO and constrained DRO. They differ in how to define the uncertainty set of $\mathbf{p}$.

Below, we use the generic notation $\ell(\mathbf{w}; \mathbf{z})$ to denote the loss of a model $\mathbf{w}$ on a random data point $\mathbf{z}$ following a distribution denoted by $\mathbb{P}$. For supervised learning, this specializes to $\ell(\mathbf{w}; \mathbf{z}) = \ell(h(\mathbf{w}; \mathbf{x}), y)$, where $\mathbf{z} = (\mathbf{x}, y)$.

Definition 2.2 (Regularized DRO)

$$\min_{\mathbf{w}} \hat{\mathcal{R}}_S(\mathbf{w}) := \max_{\mathbf{p} \in \Delta_n} \sum_{i=1}^n p_i \ell(\mathbf{w}; \mathbf{z}_i) - \tau D_\phi\left(\mathbf{p}, \frac{\mathbf{1}}{n}\right). \tag{2.11}$$

Divergence	$\phi(t)$	$\phi^*(s)$	$D_\phi(\mathbf{p},\mathbf{q})$
KL	$t\log(t) - t + 1$	$\exp(s) - 1$	$\sum_{i=1}^n p_i \log \frac{p_i}{q_i}$
Burg entropy	$-\log t + t - 1$	$-\log(1-s), s < 1$	$\sum_{i=1}^n q_i \log \frac{q_i}{p_i}$
χ^2	$(t-1)^2$	$\begin{cases} \frac{1}{4}s^2 + s & \text{if } s \geq -2 \\ -1 & \text{o.w.} \end{cases}$	$\sum_{i=1}^n q_i(p_i/q_i - 1)^2$
Hellinger distance	$(\sqrt{t} - 1)^2$	$\frac{s}{1-s}, s < 1$	$\sum_i (\sqrt{p_i} - \sqrt{q_i})^2$
Variation distance	$\lvert t - 1 \rvert$	$\begin{cases} s & \text{if } s \in [-1, 1] \\ -1 & \text{if} s < -1 \end{cases}$	$\sum_i \lvert p_i - q_i \rvert$
CVaR	$\mathbb{I}_{0-\infty}(t \leq 1/\alpha)$	$\frac{[s]_+}{\alpha}$	$\begin{cases} 0 & \text{if } p_i \leq q_i/\alpha, \forall i \\ \infty & \text{o.w} \end{cases}$

Table 2.1: Examples of ϕ-divergence

Definition 2.3 (Constrained DRO)

$$\min_{\mathbf{w}} \hat{\mathcal{R}}_{\mathcal{S}}(\mathbf{w}) := \max_{\mathbf{p}\in\Omega} \sum_{i=1}^{n} p_i \ell(\mathbf{w}; \mathbf{z}_i) \tag{2.12}$$
$$\text{where} \quad \Omega = \left\{\mathbf{p} \middle| \mathbf{p} \in \Delta_n, D_\phi\left(\mathbf{p}, \frac{\mathbf{1}}{n}\right) \leq \rho\right\}.$$

The regularized DRO uses a regularization on the **p** to implicitly define the uncertainty set, and the constrained DRO uses a constraint on **p** to explicitly define the uncertainty set.

The maximization over **p** in the DRO formulations simulates a worst-case scenario, thereby enhancing the model's robustness. The DRO objective interpolates between the maximal loss and the average loss:

- Without the ϕ-divergence regularization or constraint (i.e., $\tau = 0$ or $\rho = \infty$), the objective simplifies to the maximal loss among all samples, which is particularly beneficial for handling imbalanced data.
- Conversely, when $\rho = 0$ or $\tau = \infty$, the DRO objective reduces to the standard empirical risk.

Adding a tunable ϕ-divergence regularization or constraint (via τ or ρ) increases the model's robustness.

A list of ϕ-divergence is presented in Table 2.1. Two commonly used ones in machine learning are presented below:

- KL-Divergence: With $\phi(t) = t\log t - t + 1$, the ϕ-divergence becomes the KL divergence:
$$\mathrm{KL}(\mathbf{p},\mathbf{q}) = D_\phi(\mathbf{p},\mathbf{q}) = \sum_{i=1}^{n} p_i \log(p_i/q_i).$$
- Conditional Value-at-Risk (CVaR): With $\phi(t) = \mathbb{I}_{0-\infty}(t \leq 1/\alpha)$, where $\alpha \in (0, 1]$ and $\mathbb{I}_{0-\infty}$ is $0 - \infty$ indicator function, the divergence becomes $D_\phi(\mathbf{p},\mathbf{q}) = 0$ if

$p_i \leq q_i/\alpha \, \forall i$, otherwise $D_\phi(\mathbf{p}, \mathbf{q}) = \infty$. The resulting DRO formulation is also known as the empirical CVaR.

The Dual form of Regularized DRO

Solving the above DRO formulations requires dealing with a high-dimensional variable $\mathbf{p}$ from a simplex, which will incur additional overhead compared with solving ERM when the number of training data is large. The reason is that it requires performing a projection on the simplex Δ_n or the constrained simplex $\Omega = \{\mathbf{p} \in \Delta_n, D_\phi(\mathbf{p}, \frac{\mathbf{1}}{n}) \leq \rho\}$. To reduce this overhead, one approach is to convert the problem into unconstrained one using the Langrangian dual theory based on the convex conjugate of ϕ function.

Proposition 2.1 (Dual form of Regularized DRO). *Let $\phi^*(s) = \max_{t \geq 0} ts - \phi(t)$. Then we have*

$$\max_{\mathbf{p} \in \Delta_n} \sum_{i=1}^n p_i \ell(\mathbf{w}; \mathbf{z}_i) - \tau D_\phi\left(\mathbf{p}, \frac{\mathbf{1}}{n}\right) = \min_\nu \frac{\tau}{n} \sum_{i=1}^n \phi^*\left(\frac{\ell(\mathbf{w}; \mathbf{z}_i) - \nu}{\tau}\right) + \nu. \tag{2.13}$$

The proof can be found in Example 1.14 in Chapter 1.

Examples of Regularized DRO

Example 2.1. (KL-divergence Regularized DRO) *For the special case of using KL-divergence, we can further simplify the above objective function. Since $\phi(t) = t \log t - t + 1$, then $\phi^*(s) = \exp(s) - 1$ (see Example 1.15) and solving ν yields*

$$\nu = \tau \log\left(\frac{1}{n} \sum_{i=1}^n \exp(\ell(\mathbf{w}; \mathbf{z}_i)/\tau)\right).$$

Plugging it back into the objective, we can obtain a simplified form

$$\max_{\mathbf{p} \in \Delta_n} \sum_{i=1}^n p_i \ell(\mathbf{w}; \mathbf{z}_i) - \tau KL\left(\mathbf{p}, \frac{\mathbf{1}}{n}\right) = \tau \log\left(\frac{1}{n} \sum_{i=1}^n \exp(\ell(\mathbf{w}; \mathbf{z}_i)/\tau)\right).$$

As a result, with $\phi(t) = t \log t - t + 1$, the KL-divergence regularized DRO (2.11) is equivalent to

$$\min_{\mathbf{w}} \tau \log\left(\frac{1}{n} \sum_{i=1}^n \exp\left(\frac{\ell(\mathbf{w}; \mathbf{z}_i)}{\tau}\right)\right). \tag{2.14}$$

Example 2.2. (Empirical CVaR) *As another example, we derive the dual form of the empirical CVaR. With simple algebra, we can derive that $\phi^*(s) = \frac{[s]_+}{\alpha}$ (see Example 1.15) for $\phi(t) = \mathbb{I}_{0-\infty}(t \leq 1/\alpha)$.*

As a result, with $\phi(t) = \mathbb{I}_{0-\infty}(t \leq 1/\alpha)$, the empirical CVaR–$\alpha$ (2.11) is equivalent to

$$\min_{\mathbf{w},\nu} \frac{1}{n\alpha} \sum_{i=1}^{n} [\ell(\mathbf{w}; \mathbf{z}_i) - \nu]_+ + \nu. \tag{2.15}$$

When $k = n\alpha \in [1, n]$ is an integer, the above objective reduces to the average of top-k loss values when sorting them in descending order, as shown in the following lemma.

Lemma 2.2 *Let $\ell_{[i]}$ denote the i-th largest loss among $\{\ell(\mathbf{w}; \mathbf{z}_i), i = 1, \ldots, n\}$ ranked in descending order. If $\alpha = k/n$, we have*

$$\min_{\nu} \frac{1}{n\alpha} \sum_{i=1}^{n} [\ell(\mathbf{w}; \mathbf{z}_i) - \nu]_+ + \nu = \frac{1}{k} \sum_{i=1}^{k} \ell_{[i]}. \tag{2.16}$$

Proof. First, we have

$$\min_{\nu} \frac{1}{n\alpha} \sum_{i=1}^{n} [\ell(\mathbf{w}; \mathbf{z}_i) - \nu]_+ + \nu = \min_{\nu} \frac{1}{n\alpha} \sum_{i=1}^{n} [\ell_{[i]} - \nu]_+ + \nu.$$

Let ν_* be an optimal solution given $\mathbf{w}$. Due to the first-order optimality condition, we have

$$0 \in \frac{1}{k} \sum_{i=1}^{n} \partial_\nu [\ell_{[i]} - \nu_*]_+ + 1.$$

Hence,

$$-k \in \sum_{i=1}^{n} \partial_\nu [\ell_{[i]} - \nu_*]_+. \tag{2.17}$$

Let us first assume $\ell_{[k+1]} < \ell_{[k]}$. We will show hat $\nu_* \in (\ell_{[k+1]}, \ell_{[k]}]$ satisfy this condition. Since $-1 \in \partial_\nu [\ell_{[i]} - \nu_*]_+$ for $i = 1, \ldots, k$ due to $\ell_{[i]} \geq \nu_*$ and $\partial_\nu [\ell_{[i]} - \nu_*]_+ = 0$ for $i = k+1, \ldots, n$ due to $\ell_{[i]} < \nu_*$. Hence, it verifies that the condition (2.17) holds at such ν_*.

If $\ell_{[k+1]} = \ell_{[k]}$, we argue that $\nu_* = \ell_{[k]}$ can still satisfy (2.17). This is because $-1 \in \partial_\nu [\ell_{[i]} - \nu_*]_+$ for $i = 1, \ldots, k$ and $0 \in \partial_\nu [\ell_{[i]} - \nu_*]_+$ for $\ell_{[i]} = \ell_{[k+1]}, i \geq k+1$ and $\partial_\nu [\ell_{[i]} - \nu_*]_+ = 0$ for $\ell_{[i]} < \ell_{[k+1]}, i \geq k+1$. Then the conclusion follows. □

The Dual form of Constrained DRO

For transforming the constrained DRO, we can use the following proposition based on the Lagrangian duality theory.

Proposition 2.2 (Dual form of Constrained DRO)**.** *Let* $\phi^*(s) = \max_{t\geq 0} ts - \phi(t)$*. Then we have*

$$\max_{\mathbf{p}\in\Delta, D_\phi(\mathbf{p},\frac{1}{n})\leq\rho} \sum_{i=1}^n p_i \ell(\mathbf{w};\mathbf{z}_i) = \min_{\tau\geq 0,\nu} \frac{\tau}{n}\sum_{i=1}^n \phi^*\left(\frac{\ell(\mathbf{w};\mathbf{z}_i)-\nu}{\tau}\right) + \nu + \tau\rho. \tag{2.18}$$

The proof is similar to that of Proposition 2.1.

Examples of Constrained DRO

Example 2.3. *(***KL Constrained DRO***) With* $\phi(t) = t\log t - t + 1$*, the KL-divergence constrained DRO (2.12) is equivalent to:*

$$\min_{\mathbf{w},\tau\geq 0} \tau\log\left(\frac{1}{n}\sum_{i=1}^n \exp\left(\frac{\ell(\mathbf{w};\mathbf{z}_i)}{\tau}\right)\right) + \tau\rho. \tag{2.19}$$

KL-regularized DRO and KL-constrained DRO play important roles in many modern artificial intelligence applications. The LDR loss (2.5) can be interpreted as a form of KL-regularized DRO, except that the uncertainty is placed on the distribution of class labels for each individual data point. We will present additional applications in Section 2.4.

The Optimization Challenge

Although the transformed optimization problems do not involve dealing with a high-dimensional variable $\mathbf{p} \in \Delta_n$, the new optimization problems (2.14), (2.19) are not of the same form as ERM. The critical assumption that an unbiased gradient can be easily computed fails. We will cast them as instances of stochastic compositional optimization, which is topic of Chapter 3 of the book.

2.2.2 Optimized Certainty Equivalent

How to understand the generalization of DRO? One way is to still consider bounding the expected risk $\mathcal{R}(\mathbf{w})$ of the learned model. However, the expected risk may not be a good measure when the data distribution is skewed.

For simplicity, let us consider a binary classification problem with $\Pr(\mathbf{x}, y = 1) = \pi_+ \Pr(\mathbf{x}|y = 1)$ and $\Pr(\mathbf{x}, y = -1) = \pi_- \Pr(\mathbf{x}|y = -1)$, where $\pi_+ = \Pr(y = 1), \pi_- = \Pr(y = -1)$. Let $\mathbb{P}_+$ and $\mathbb{P}_-$ the distributions of $\mathbf{x}$ conditioned on $y = 1$ and $y = -1$, respectively. If $\pi_+ \ll \pi_-$, then by the Law of total expectation we have

$$\mathcal{R}(\mathbf{w}) = \mathbb{E}_{\mathbf{x},y}\ell(h(\mathbf{w};\mathbf{x}), y) = \pi_+\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_+}[\ell(h(\mathbf{w};\mathbf{x}), 1)] + \pi_-\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_-}[\ell(h(\mathbf{w};\mathbf{x}), -1)]. \tag{2.20}$$

Since $\pi_- \gg \pi_+$, the expected risk would be dominated by the expected loss of data from the negative class. As a result, a small $\mathcal{R}(\mathbf{w})$ does not necessarily indicate a small $\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_+}[\ell(\mathbf{w};\mathbf{x},1)]$.

Instead, we consider the population risk of DRO as the target measure. A formal definition of the population risk for the regularized DRO (2.11) is given below.

Definition 2.4 (Population risk of DRO) Given a data distribution $\mathbb{P}$, for any $\tau > 0$, we define the population risk of regularized DRO (2.11) as:

$$\hat{\mathcal{R}}_\tau(\mathbf{w}) := \max_{\mathbb{Q}\in\mathcal{Q}} \mathbb{E}_{\mathbf{z}'\sim\mathbb{Q}}\ell(\mathbf{w};\mathbf{z}') - \tau\mathbb{E}_{\mathbb{P}}\phi\left(\frac{d\mathbb{Q}}{d\mathbb{P}}\right) \tag{2.21}$$

$$= \min_{\nu} \tau\mathbb{E}_{\mathbf{z}\sim\mathbb{P}}\phi^*\left(\frac{\ell(\mathbf{w};\mathbf{z})-\nu}{\tau}\right) + \nu, \tag{2.22}$$

where $\phi^*(s) = \max_{t\geq 0} ts - \phi(t)$.

In the definition above, $\mathcal{Q} = \{\mathbb{Q} \mid \mathbb{Q} \ll \mathbb{P}\}$ denotes the set of probability measures that are absolutely continuous with respect to $\mathbb{P}$. A probability measure $\mathbb{Q}$ is said to be absolutely continuous with respect to $\mathbb{P}$, denoted $\mathbb{Q} \ll \mathbb{P}$, if every event that has probability 0 under $\mathbb{P}$ also has probability 0 under $\mathbb{Q}$. If $\mathbb{P}$ and $\mathbb{Q}$ admit densities $p(z)$ and $q(z)$ with respect to a common dominating measure on $\mathcal{Z}$, and $\mathbb{Q} \ll \mathbb{P}$, then

$$\mathbb{E}_{\mathbb{P}}\left[\phi\left(\frac{d\mathbb{Q}}{d\mathbb{P}}\right)\right] = \int_{\mathcal{Z}} p(z)\,\phi\left(\frac{q(z)}{p(z)}\right)\,dz.$$

The equivalent counterpart in (2.22) is a risk measure originates from the **optimized certainty equivalent (OCE)**, a concept popularized in mathematical economics (Ben-Tal and Teboulle, 1986a). Minimizing OCE has an effect of so-called **risk-aversion**, which discourages models from having rare but catastrophic errors. Two special cases are discussed below:

- When $\phi(t) = \mathbb{I}_{0-\infty}(t \leq 1/\alpha)$, the OCE becomes the CVaR-α, i.e.,

  $$\hat{\mathcal{R}}_\tau(\mathbf{w}) = \mathbb{E}_{\mathbf{z}}[\ell(\mathbf{w};\mathbf{z})|\ell(\mathbf{w};\mathbf{z}) \geq \mathrm{VAR}_\alpha(\ell(\mathbf{w};\mathbf{z}))],$$

  where $\mathrm{VAR}_\alpha(\ell(\mathbf{w};\mathbf{z})) = \sup_s[\Pr(\ell(\mathbf{w};\mathbf{z}) \geq s) \geq \alpha]$ is the α-quantile or "value-at-risk" of the random loss values.
- When $\phi(t) = t\log t - t + 1$, OCE becomes the entropic risk:

  $$\hat{\mathcal{R}}_\tau(\mathbf{w}) = \tau\log\left(\mathbb{E}_{\mathbf{z}}\exp\left(\frac{\ell(\mathbf{w};\mathbf{z})}{\tau}\right)\right).$$

We present two properties of OCE below.

Lemma 2.3 *Let $\partial\phi^*(t) = \{s : \phi'^*_-(t) \leq s \leq \phi'^*_+(t)\}$. If $a < b$, then $0 \leq \phi'^*_+(a) \leq \phi'^*_-(b)$.*

Proof. Due to the definition $\phi^*(s) = \max_{t\geq 0} ts - \phi(t)$, we have $\partial\phi^*(s) \geq 0$, which indicates that ϕ^* is non-decreasing. Since ϕ^* is also convex, the conclusion follows from the convex analysis (Rockafellar, 1970b)[Section 24]. □

Lemma 2.4 *For any $\tau > 0, \mathbf{w} \in \mathbb{R}^d$, it holds that $\hat{\mathcal{R}}_\tau(\mathbf{w}) \geq \mathcal{R}(\mathbf{w})$.*

Proof. Since $\phi(1) = 0$, then $\phi^*(s) = \max_{t\geq 0} ts - \phi(t) \geq s - \phi(1) = s$. Hence,

$$\begin{aligned}\hat{\mathcal{R}}_\tau(\mathbf{w}) &= \min_\nu \tau\mathbb{E}_{\mathbf{z}}\phi^*\left(\frac{\ell(\mathbf{w};\mathbf{z}) - \nu}{\tau}\right) + \nu \\ &\geq \min_\nu \tau\mathbb{E}_{\mathbf{z}}\left(\frac{\ell(\mathbf{w};\mathbf{z}) - \nu}{\tau}\right) + \nu = \mathcal{R}(\mathbf{w}).\end{aligned}$$

□

Why it matters:

Lemma 2.3 implies that a data with a larger loss $\ell(h(\mathbf{w};\mathbf{x}), y)$ will have a higher weight in the gradient calculation in terms of $\mathbf{w}$.
Lemma 2.4 indicates that OCE is a stronger measure than the expected risk. A small OCE will imply a small expected risk, while the reverse is not necessarily true.

Based on OCE, we can define the excess risk $\hat{\mathcal{R}}_\tau(\mathbf{w}) - \min_{\mathbf{w}\in\mathcal{W}} \hat{\mathcal{R}}_\tau(\mathbf{w})$ and decompose it into an optimization error and a generalization error.

Lemma 2.5 *For a learned model $\mathbf{w} = \mathcal{A}(\mathcal{S};\zeta)$ for solving empirical DRO (2.11), we have*

$$\begin{aligned}\hat{\mathcal{R}}_\tau(\mathbf{w}) - \min_{\mathbf{u}\in\mathcal{W}} \hat{\mathcal{R}}_\tau(\mathbf{w}) \leq{}& \underbrace{\max\{\hat{\mathcal{R}}_\tau(\mathbf{w}) - \hat{\mathcal{R}}_{\mathcal{S}}(\mathbf{w}), \hat{\mathcal{R}}_{\mathcal{S}}(\mathbf{w}_*) - \hat{\mathcal{R}}_\tau(\mathbf{w}_*)\}}_{\text{generalization error}} \\ &+ \underbrace{\hat{\mathcal{R}}_{\mathcal{S}}(\mathbf{w}) - \min_{\mathbf{u}\in\mathcal{W}} \hat{\mathcal{R}}_{\mathcal{S}}(\mathbf{w})}_{\text{optimization error}},\end{aligned}$$

where $\mathbf{w}_* = \arg\min_{\mathbf{u}\in\mathcal{W}} \hat{\mathcal{R}}_\tau(\mathbf{u})$.

2.2.3 Group Distributionally Robust Optimization

Group DRO is an extension of DRO by aggregating data into groups and using DRO on the group level to formulate a robust risk function. It is helpful to promote equity of the learned model and mitigating the impact of spurious correlations that exist between the label and some features, by using prior knowledge to group the data.

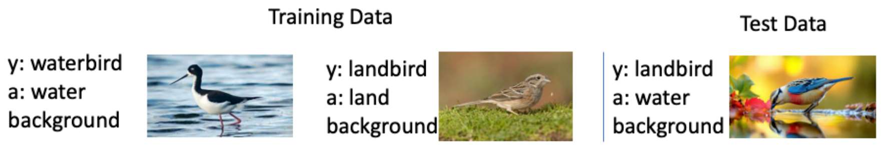

Fig. 2.2: Illustrative of spurious correlation between the class label and some feature: waterbird images mostly have water background and landbird images mostly have land background.

Let us consider an illustrative example of classifying waterbird images from landbird images (see Figure 2.2). The training data may have the same number of waterbird images and landbird images. However, most waterbird images may have water in the background and most landbird images may have land in the background. Standard empirical risk minimization may learn spurious correlation between the class labels (e.g., waterbird) and the specific value of some attribute (e.g., the water background). As a consequence, the model may perform poorly on waterbird images with land background.

Critical: Data may exhibit imbalance not in the marginal distribution of class label but some joint distribution of the class label and some attributes, which causes the spurious correlation.

GDRO can be used to mitigate this issue by leveraging prior knowledge of spurious correlations to define groups over the training data. Let the training data be divided into multiple groups $\mathcal{G}_1, \ldots, \mathcal{G}_K$, where $\mathcal{G}_j = \{(\mathbf{x}_1^j, y_1^j), \ldots (\mathbf{x}_{n_k}^j, y_{n_k}^j)\}$ includes a set of examples for the j-th group. We define an averaged loss over examples from each group $L_j(\mathbf{w}) = \frac{1}{n_j}\sum_{i=1}^{n_j} \ell(\mathbf{w}; \mathbf{x}_i^j, y_i^j)$. Then a regularized group DRO can be defined as

$$\min_{\mathbf{w}} \max_{\mathbf{p}\in\Delta_K} \sum_{j=1}^{K} p_j L_j(\mathbf{w}) - \tau D_\phi\left(\mathbf{p}, \frac{\mathbf{1}}{K}\right), \tag{2.23}$$

and a constrained group DRO is given by:

$$\min_{\mathbf{w}} \max_{\mathbf{p}\in\Delta_K, D_\phi(\mathbf{p}, \frac{\mathbf{1}}{K})\leq\rho} \sum_{j=1}^{K} p_j L_j(\mathbf{w}). \tag{2.24}$$

By doing so, the learning process is less likely to be dominated by the majority group associated with the spurious correlation between the label and a particular feature. If the model only captures the spurious correlation, the loss for the minority group will be large, which in turn drives the learning process to reduce this loss and thereby mitigate the spurious correlation.

Examples and Reformulations

Similar to before, we can convert the min-max problem into minimization problem to reduce additional overhead of dealing with a large number of groups. We give two examples of using KL-divergence constraint of **p** and CVaR-α.

With $\phi(t) = t \log t - t + 1$, the KL-divergence constrained group DRO (2.24) is equivalent to

$$\min_{\mathbf{w}, \tau \geq 0} \tau \log \left(\frac{1}{K} \sum_{j=1}^{K} \exp \left(\frac{L_j(\mathbf{w})}{\tau} \right) \right) + \tau \rho. \tag{2.25}$$

With $\phi(t) = \mathbb{I}_{0-\infty}(t \leq 1/\alpha)$, CVaR-$\alpha$ group DRO (2.11) is equivalent to

$$\min_{\mathbf{w}, \nu} \frac{1}{K\alpha} \sum_{j=1}^{K} [L_j(\mathbf{w}) - \nu]_+ + \nu. \tag{2.26}$$

The Optimization Challenge

Again, these new optimization problems (2.25), (2.26) cannot be solved by simply using existing stochastic algorithms for ERM since $L_j(\mathbf{w})$ depends on many data and they are inside non-linear functions. In particular, the problem (2.26) is an instance of finite-sum coupled compositional optimization (FCCO), which will be explored in Chapter 5 in depth.

2.3 Empirical X-risk Minimization

So far, we have revisited classical ideas of machine learning based on empirical risk minimization and its distributionally robust variant. In these risk functions, we assume each data defines a loss based on itself. These losses are typically surrogate functions of a prediction error measuring the inconsistency between the prediction and the label.

However, such loss functions are insufficient to capture many objectives, which involve comparison between different data points. Examples include areas under ROC curves (AUROC) and areas under precision-recall curves (AUPRC) for imbalanced data classification, ranking measures such as normalized discounted cumulative gain (NDCG), mean average precision (MAP) and listwise losses for learning to rank, and contrastive losses for representation learning.

The standard ERM framework is inadequate for optimizing such metrics and losses, as they involve interactions across multiple data points. We need a new mathematical framework to understand the challenge and to design provable and practical algorithms. To this end, we introduce a new risk minimization framework, named empirical X-risk minimization (EXM), as defined below:

Empirical X-risk Minimization (EXM)

X-risk refers to a family of risks such that the loss of each data is defined in a way that contrasts the data with many others. Mathematically, empirical X-risk minimization is formulated as:

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} f_i(g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i)), \tag{2.27}$$

where $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is a set of data points, each $\mathcal{S}_i$ contains a number of items, f_i is a simple but non-linear function, and $g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i)$ involves the coupling between $\mathbf{x}_i$ and all data in $\mathcal{S}_i$. A simple instance of $g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i)$ is the following averaged form:

$$g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i) = \frac{1}{|\mathcal{S}_i|} \sum_{\mathbf{z} \in \mathcal{S}_i} \ell(\mathbf{w}; \mathbf{x}_i, \mathbf{z}). \tag{2.28}$$

With g given in (2.28), EXM is an instance of finite-sum coupled compositional optimization (**FCCO**), a framework explored in detail in Chapter 5.

Below, we present several important instances of X-risks.

2.3.1 AUC Losses

Areas under ROC curves (AUC) are commonly used to measure performance for the imbalanced data classification.

What is Imbalanced Data Classification?

Imbalanced data classification refers to classification problems, where the number of examples from some classes is significantly larger than that of other classes.

Definition and an Empirical Estimator of AUC

The ROC curve is the plot of the true positive rate (TPR) against the false positive rate (FPR) at each threshold setting. Let $\mathbb{P}_+, \mathbb{P}_-$ denote the distribution of random positive and negative data, respectively. Let $h(\cdot)$ denote a predictive function. For a given threshold t, the TPR of h can be written as $\mathrm{TPR}(t) = \Pr(h(\mathbf{x}) > t | y = 1) = \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_+}[\mathbb{I}(h(\mathbf{x}) > t)]$, and the FPR can be written as $\mathrm{FPR}(t) = \Pr(h(\mathbf{x}) > t | y = -1) = \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_-}[\mathbb{I}(h(\mathbf{x}) > t)]$. Let $F_-(t) = 1 - \mathrm{FPR}(t)$ denote the cumulative density function

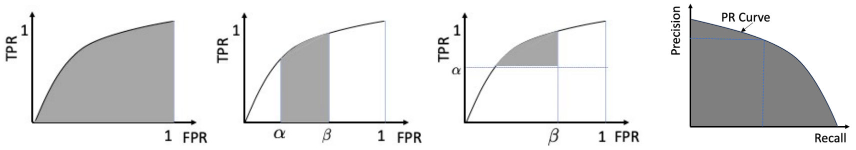

Fig. 2.3: Areas under ROC Curves

of the random variable $h(\mathbf{x}_-)$ for $\mathbf{x}_- \sim \mathbb{P}_-$. Let $p_-(t)$ denotes its corresponding probability density function. Similarly, let $F_+(t) = 1 - \text{TPR}(t)$ and $p_+(t)$ denote the cumulative density function and the probability density function of $h(\mathbf{x}_+)$ for $\mathbf{x}_+ \sim \mathbb{P}_+$, respectively.

For a given $u \in [0,1]$, let $\text{FPR}^{-1}(u) = \inf\{t \in \mathbb{R} : \text{FPR}(t) \leq u\}$. The ROC curve is defined as $\{u, \text{ROC}(u)\}$, where $u \in [0,1]$ and $\text{ROC}(u) = \text{TPR}(\text{FPR}^{-1}(u))$.

Hence, we have the following theorem.

Theorem 2.1 *The AUC for a predictive function h is equal to*

$$AUC(h) = \Pr(h(\mathbf{x}_+) > h(\mathbf{x}_-)) = \mathbb{E}_{\mathbf{x}_+\sim\mathbb{P}_+,\mathbf{x}_-\sim\mathbb{P}_-}[\mathbb{I}(h(\mathbf{x}_+) > h(\mathbf{x}_-))]. \tag{2.29}$$

Proof. The AUC score of h is given by

$$\begin{aligned}\text{AUC}(h) &= \int_0^1 \text{ROC}(u)du = \int_{-\infty}^{\infty} \text{TPR}(t)dF_-(t) = \int_{-\infty}^{\infty} \text{TPR}(t)p_-(t)dt \\ &= \int_{-\infty}^{\infty}\int_t^{\infty} p_+(s)dsp_-(t)dt = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_+(s)p_-(t)\mathbb{I}(s > t)dsdt.\end{aligned}$$

Since $h(\mathbf{x}_+)$ follows $p_+(s)$ and $h(\mathbf{x}_-)$ follows $p_-(t)$, we can conclude the proof. □

This indicates that AUC is a pairwise ranking metric. An ideal predictive function that ranks all positive examples above negative examples has a perfect AUC score 1. It also implies the following empirical non-parametric estimator of AUC based on a set of data $\mathcal{S}$ with n_+ positive samples in $\mathcal{S}_+$ and n_- negative samples in $\mathcal{S}_-$:

$$\text{AUC}(h; \mathcal{S}) = \frac{1}{n_+n_-}\sum_{\mathbf{x}_+\in\mathcal{S}_+,\mathbf{x}_-\in\mathcal{S}_-} \mathbb{I}(h(\mathbf{x}_+) > h(\mathbf{x}_-)), \tag{2.30}$$

which is also known as the Mann-Whitney U-statistic (Hanley and McNeil, 1982).

Necessity of Maximizing AUC

AUC is more appropriate than accuracy for assessing the performance of imbalanced data classification. Let us consider an example with 2 positive data and 100 negative data. If one positive data has a prediction score 0.5 and another one has a prediction score -0.2, and all negative data has prediction scores less than 0 but larger than -0.2. In this case, if we choose a classification threshold as 0, then the accuracy is $101/102 = 0.99$. However, the (emprical) AUC score according to (2.30) is given by $100/200 = 0.5$. "*Can a model that optimizes the accuracy also optimize the AUC score?*" Unfortunately, this is not the case as different classifiers that have the same accuracy could have dramatic different AUC (Cortes and Mohri, 2003). An example is illustrated in Table 2.2. Hence, it makes sense to directly optimize AUC.

Critical: A model that optimizes accuracy does not necessarily optimize AUC.

Example 1		2c**Example 2**		2c**Example 3**		
Prediction	Ground Truth	Prediction	Ground Truth		Prediction	Ground Truth
0.9	1	0.9	1		0.9	1
0.8	1	**0.41**(↓)	1		**0.41**(↓)	1
0.7	1	0.7	1		**0.40**(↓)	1
0.6	0	0.6	0		**0.49**(↓)	0
0.6	0	**0.49**(↓)	0		**0.48**(↓)	0
0.47	0	0.47	0		0.47	0
0.47	0	0.47	0		0.47	0
0.45	0	0.45	0		0.45	0
0.43	0	0.43	0		0.43	0
0.42	0	0.42	0		0.42	0
⋮	⋮	⋮	⋮		⋮	⋮
0.1	0	0.1	0		0.1	0
Acc=0.92		Acc=0.92 (—)			Acc=0.92 (—)	
AUC=1.00		AUC=**0.89** (↓)			AUC=**0.78** (↓)	

Table 2.2: Illustrations of variance of AUC for different classifiers with the same Accuracy on an imbalanced dataset of 25 samples with a positive ratio of 3/25. The accuracy threshold is 0.5. **Example 1** shows that all positive instances rank higher than negative instances and two negative instances are misclassified to positive class. **Example 2** shows that 1 positive instance ranks lower than 7 negative instances and 1 positive and 1 negative instances are missclassifed. **Example 3** shows that 2 positive instances rank lower than 7 negative instances, and 2 positive instances are also missclassifed as negative class. Overall, we can observe that AUC drops dramatically as the ranks of positive instances drop but meanwhile Accuracy remains unchanged.

Pairwise Surrogate Losses

Using a pairwise surrogate loss $\ell(\cdot)$ of the indicator function $\mathbb{I}(t \geq 0)$ (see examples in Table 2.3), we have the following empirical AUC optimization problem for learning a parameterized function $h(\mathbf{w}; \cdot)$:

Pairwise Loss	$\ell(t)$	Monotone
Squared Hinge	$(c+t)_+^2$	Yes
Hinge	$(c+t)_+$	Yes
Logistic	$\log(1+\exp(st))$	Yes
Sigmoid	$(1+\exp(-st))^{-1}$	Yes
Square	$(c+t)^2$	No
Barrier Hinge	$\max(-s(c-t)+c, \max(s(-t-c), c+t))$	No

Table 2.3: Surrogate loss functions for pairwise modeling with the input argument $t = h(\mathbf{w}; \mathbf{x}_-) - h(\mathbf{w}; \mathbf{x}_+)$. For the sake of simplicity, denote $\max(0, t)$ by t_+, denote the scaling hyper-parameter by $s > 0$ and margin hyper-parameter by $c > 0$.

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n_+}\frac{1}{n_-}\sum_{\mathbf{x}_i\in\mathcal{S}_+}\sum_{\mathbf{x}_j\in\mathcal{S}_-} \ell(h(\mathbf{w};\mathbf{x}_j) - h(\mathbf{w};\mathbf{x}_i)). \tag{2.31}$$

This can be regarded as a special case of (2.27) by setting

$$\begin{aligned} g(\mathbf{w};\mathbf{x}_i,\mathcal{S}_-) &= \frac{1}{n_-}\sum_{\mathbf{x}_j\in\mathcal{S}_-} \ell(h(\mathbf{w};\mathbf{x}_j) - h(\mathbf{w};\mathbf{x}_i)), \\ f_i(g) &= g. \end{aligned}$$

This is the simplest form of EXM as f is just a linear function. An unbiased stochastic gradient can be easily computed based on a pair of data points consisting of a random positive and a random negative data point.

Compositional Objectives

An alternative approach to formulate AUC maximization is to decouple the pairwise comparison between positive and negative examples. A generic formulation is given by:

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^d,(a,b)\in\mathbb{R}^2} \quad & \frac{1}{|\mathcal{S}_+|}\sum_{\mathbf{x}_i\in\mathcal{S}_+}(h(\mathbf{w};\mathbf{x}_i) - a)^2 + \frac{1}{|\mathcal{S}_-|}\sum_{\mathbf{x}_j\in\mathcal{S}_-}(h(\mathbf{w};\mathbf{x}_j) - b)^2 \\ & + f\left(\frac{1}{|\mathcal{S}_-|}\sum_{\mathbf{x}_j\in\mathcal{S}_-} h(\mathbf{w};\mathbf{x}_j) - \frac{1}{|\mathcal{S}_+|}\sum_{\mathbf{x}_i\in\mathcal{S}_+} h(\mathbf{w};\mathbf{x}_i)\right), \end{aligned} \tag{2.32}$$

where f is a non-linear function. The last component is a compositional function.

The above formulation also has a clear physical meaning. In particular, the first two terms aim to push the prediction scores of positive and negative examples to center around their means, respectively, and the third term aims to push the mean score of positive examples to be larger than the mean score of negative examples.

The above formulation is motivated by the pairwise formulation with a square surrogate function $\ell(h(\mathbf{w};\mathbf{x}_j) - h(\mathbf{w};\mathbf{x}_i)) = (c + h(\mathbf{w};\mathbf{x}_j) - h(\mathbf{w};\mathbf{x}_i))^2$. Indeed, in this case, (2.31) is equivalent to (2.32) with $f(s) = (s+c)^2$. We leave this as an exercise for interested readers. Nevertheless, using $f(s) = [s+c]_+^2$ in (2.32) is more robust than $f(s) = (s+c)^2$ with $c > 0$.

Solving the above problem requires compositional optimization techniques, which will be discussed in Section 6.4.1.

2.3.2 Average Precision Loss

Area under precision-recall curve (AUPRC) is another commonly used measure for highly imbalanced data. The precision and recall of a scoring function h at threshold t are defined as

$$\begin{aligned}\mathrm{Rec}(t) &:= \Pr(h(\mathbf{x}) > t \mid y = 1) = \mathrm{TPR}(t),\\ \mathrm{Prec}(t) &:= \Pr(y = 1 \mid h(\mathbf{x}) > t).\end{aligned}$$

For a given $u \in [0,1]$, let

$$\mathrm{TPR}^{-1}(u) = \inf\{t \in \mathbb{R} : \mathrm{TPR}(t) \le u\}.$$

The precision–recall (PR) curve is defined as $\{(u, \mathrm{PR}(u))\}$, where $u \in [0,1]$ and

$$\mathrm{PR}(u) = \mathrm{Prec}\big(\mathrm{TPR}^{-1}(u)\big).$$

Hence, the area under the precision–recall curve (AUPRC) for h is defined as

$$\mathrm{AUPRC}(h) = \int_0^1 \mathrm{PR}(u)\, du.$$

Theorem 2.2 *The AUPRC for a predictive function h is equal to*

$$AUPRC(h) = \int_{-\infty}^{\infty} Prec(t)\, p_+(t)\, dt = \mathbb{E}_{\mathbf{x}_+\sim\mathbb{P}_+}\big[Prec(h(\mathbf{x}_+))\big]. \tag{2.33}$$

Proof. By definition,

$$\mathrm{AUPRC}(h) = \int_0^1 \mathrm{PR}(u)\, du = \int_0^1 \mathrm{Prec}\big(\mathrm{TPR}^{-1}(u)\big)\, du.$$

Let $u = \mathrm{TPR}(t) = 1 - F_+(t)$. Then $du = -p_+(t)\, dt$. Therefore,

$$\mathrm{AUPRC}(h) = \int_{\infty}^{-\infty} \mathrm{Prec}(t)\, (-p_+(t)\, dt) = \int_{-\infty}^{\infty} \mathrm{Prec}(t)\, p_+(t)\, dt,$$

which proves (2.33). □

The above theorem yields the following empirical estimator of AUPRC. For a set of training examples $\mathcal{S} = \mathcal{S}_+ \cup \mathcal{S}_-$, a non-parametric estimator of AUPRC is average precision (AP) (Boyd et al., 2013):

$$\mathrm{AP}(h) = \frac{1}{n_+} \sum_{\mathbf{x}_i \in \mathcal{S}_+} \frac{\sum_{\mathbf{x}_j \in \mathcal{S}_+} \mathbb{I}(h(\mathbf{x}_j) \geq h(\mathbf{x}_i))}{\sum_{\mathbf{x}_j \in \mathcal{S}} \mathbb{I}(h(\mathbf{x}_j) \geq h(\mathbf{x}_i))}. \tag{2.34}$$

AP is an unbiased estimator of AUPRC in the limit $n \longrightarrow \infty$.

Necessity of Maximizing AUPRC

While AUC is generally more suitable than accuracy for imbalanced classification tasks, it may fail to adequately capture misorderings among top-ranked examples. Consider a scenario with 2 positive and 100 negative samples. If the two positive samples are ranked below just two of the negative ones, followed by the remaining 98 negatives, the resulting AUC is $196/200 = 0.98$, which appears high. However, this model would be inadequate if our focus is on the top two predicted positive instances. In drug discovery, for example, models are expected to identify the most promising candidate molecules for experimental validation. If these top-ranked predictions turn out to lack the desired properties, the resulting experimental efforts may lead to significant wasted resources and costly failures.

To avoid this issue, area under the precision-recall curve or its empirical estimator average precision is typically used as a performance metric. According to its definition (2.34), the AP score for the above example is $\frac{1}{2}(\frac{1}{3} + \frac{2}{4}) = 0.42$. In contrast, a perfect ranking that ranks the two positive examples at the top gives an AP score of 1. Unfortunately, optimizing AUC does not necessarily lead to optimal AP, as two models with identical AUC scores can exhibit significantly different AP values. This highlights the need for efficient optimization algorithms that directly maximize AP.

Critical: AUPRC/AP penalizes more on the error at the top of the ranked list.

Surrogate Loss of AP

To construct a differentiable objective for minimization, a differentiable surrogate loss $\ell(h(\mathbf{x}_j) - h(\mathbf{x}_i))$ is used in place of $\mathbb{I}(h(\mathbf{x}_j) \geq h(\mathbf{x}_i))$. Then AP can be approximated by :

$$\mathrm{AP} \approx \frac{1}{n_+} \sum_{\mathbf{x}_i \in \mathcal{S}_+} \frac{\sum_{\mathbf{x}_j \in \mathcal{S}} \mathbb{I}(y_j = 1)\ell(h(\mathbf{x}_j) - h(\mathbf{x}_i))}{\sum_{\mathbf{x}_j \in \mathcal{S}} \ell(h(\mathbf{x}_j) - h(\mathbf{x}_i))}. \tag{2.35}$$

Let us define

$$
\begin{aligned}
&f(\mathbf{g}) = -\frac{[\mathbf{g}]_1}{[\mathbf{g}]_2},\\
&\mathbf{g}(\mathbf{w};\mathbf{x}_i,\mathcal{S}) = [g_1(\mathbf{w};\mathbf{x}_i,\mathcal{S}), g_2(\mathbf{w};\mathbf{x}_i,\mathcal{S})]\\
&g_1(\mathbf{w};\mathbf{x}_i,\mathcal{S}) = \frac{1}{|\mathcal{S}|}\sum_{\mathbf{x}_j\in\mathcal{S}} \mathbb{I}(y_j = 1)\ell(h(\mathbf{w};\mathbf{x}_j) - h(\mathbf{w};\mathbf{x}_i)),\\
&g_2(\mathbf{w};\mathbf{x}_i,\mathcal{S}) = \frac{1}{|\mathcal{S}|}\sum_{\mathbf{x}_j\in\mathcal{S}} \ell(h(\mathbf{w};\mathbf{x}_j) - h(\mathbf{w};\mathbf{x}_i)).
\end{aligned}
$$

Then, we formulate AP maximization as the following problem:

$$
\min_{\mathbf{w}} \frac{1}{n_+}\sum_{\mathbf{x}_i\in\mathcal{S}_+} f(\mathbf{g}(\mathbf{w};\mathbf{x}_i,\mathcal{S})), \tag{2.36}
$$

which is a special case of EXM. We will explore efficient algorithms for optimizing AP in Section 6.4.2 using FCCO techniques.

2.3.3 Partial AUC Losses

There are two commonly used versions of partial AUC (pAUC), namely one-way pAUC (OPAUC) and two-way pAUC (TPAUC). OPAUC puts a restriction on the range of FPR, i.e., FPR $\in [\alpha,\beta]$ (Figure 2.3 middle) and TPAUC puts a restriction on the lower bound of TPR and the upper bound of FPR, i.e., TPR $\geq \alpha$, FPR $\leq \beta$ (Figure 2.3 right).

By the definition, we have the following probabilistic interpretation.

Theorem 2.3 *OPAUC with FPR restricted in the range $[\alpha,\beta]$ for a predictive function h is equal to*

$$
\mathit{OPAUC}(h|\mathit{FPR} \in (\alpha,\beta)) = \Pr(h(\mathbf{x}_+) > h(\mathbf{x}_-), h(\mathbf{x}_-) \in [\mathit{FPR}^{-1}(\beta), \mathit{FPR}^{-1}(\alpha)]). \tag{2.37}
$$

Similarly, TPAUC with FPR restricted in a range of $[0,\beta]$ and TPR restricted in a range of $[\alpha,1]$ is equal to

$$
\begin{aligned}
&\mathit{TPAUC}(h|\mathit{TPR} \geq \alpha, \mathit{FPR} \leq \beta) \qquad (2.38)\\
&\quad = \Pr(h(\mathbf{x}_+) > h(\mathbf{x}_-), h(\mathbf{x}_-) \geq \mathit{FPR}^{-1}(\beta), h(\mathbf{x}_+) \leq \mathit{TPR}^{-1}(\alpha)\}).
\end{aligned}
$$

Proof. The first part about OPAUC is similar to AUC except for the range of integral:

$$\begin{aligned}\mathrm{OPAUC}(h|\mathrm{FPR}\in(\alpha,\beta)) &= \int_{\mathrm{FPR}^{-1}(\beta)}^{\mathrm{FPR}^{-1}(\alpha)} \mathrm{TPR}(t)dF_-(t)\\ &= \int_{\mathrm{FPR}^{-1}(\beta)}^{\mathrm{FPR}^{-1}(\alpha)} \int_{-\infty}^{\infty} p_+(s)p_-(t)\mathbb{I}(s>t)dsdt.\end{aligned}$$

This concludes the proof of the first part.

For TPAUC with FPR restricted in $[0,\beta]$ and TPR restricted in $[\alpha,1]$, it is equal to OPAUC with FPR restricted in $[\gamma,\beta]$ minus the square area with FPR $\in[\gamma,\beta]$ and TPR $<\alpha$, where γ is the FPR that corresponds to TPR equals to α, i.e., $\mathrm{FPR}^{-1}(\gamma)=\mathrm{TPR}^{-1}(\alpha)$. Since $\mathrm{TPR}(t)=\int_t^\infty p_+(s)ds$ and $\mathrm{FPR}(t)=\int_t^\infty p_-(s)ds$, we have

$$\alpha=\int_{\mathrm{TPR}^{-1}(\alpha)}^{\infty} p_+(s)ds,\quad \beta=\int_{\mathrm{FPR}^{-1}(\beta)}^{\infty} p_-(t)dt.$$

Then, we have

$$\begin{aligned}&(\beta-\gamma)\alpha\\ &=\int_{\mathrm{FPR}^{-1}(\beta)}^{\infty}\int_{\mathrm{TPR}^{-1}(\alpha)}^{\infty} p_+(s)p_-(t)dsdt-\int_{\mathrm{FPR}^{-1}(\gamma)}^{\infty}\int_{\mathrm{TPR}^{-1}(\alpha)}^{\infty} p_+(s)p_-(t)dsdt\\ &=\int_{\mathrm{FPR}^{-1}(\beta)}^{\mathrm{FPR}^{-1}(\gamma)}\int_{\mathrm{TPR}^{-1}(\alpha)}^{\infty} p_+(s)p_-(t)dsdt.\end{aligned}$$

As a result,

$$\begin{aligned}&\mathrm{TPAUC}(h|\mathrm{TPR}\geq\alpha,\mathrm{FPR}\leq\beta)=\mathrm{OPAUC}(h|\mathrm{FPR}\in(\gamma,\beta))-(\beta-\gamma)\alpha\\ &=\int_{\mathrm{FPR}^{-1}(\beta)}^{\mathrm{FPR}^{-1}(\gamma)}\int_{t}^{\infty} p_+(s)p_-(t)dsdt-\int_{\mathrm{FPR}^{-1}(\beta)}^{\mathrm{FPR}^{-1}(\gamma)}\int_{\mathrm{TPR}^{-1}(\alpha)}^{\infty} p_+(s)p_-(t)dsdt\\ &=\int_{\mathrm{FPR}^{-1}(\beta)}^{\mathrm{FPR}^{-1}(\gamma)}\int_{t}^{\mathrm{TPR}^{-1}(\alpha)} p_+(s)p_-(t)dsdt=\int_{\mathrm{FPR}^{-1}(\beta)}^{\infty}\int_{t}^{\mathrm{TPR}^{-1}(\alpha)} p_+(s)p_-(t)dsdt,\end{aligned}$$

where the last equality is follows from $\mathrm{FPR}^{-1}(\gamma)=\mathrm{TPR}^{-1}(\alpha)$. Thus,

$$\begin{aligned}\mathrm{TPAUC}(h|\mathrm{TPR}\geq\alpha,\mathrm{FPR}\leq\beta)&=\int_{\mathrm{FPR}^{-1}(\beta)}^{\infty}\int_{t}^{\mathrm{TPR}^{-1}(\alpha)} p_+(s)p_-(t)dsdt\\ &=\int_{\mathrm{FPR}^{-1}(\beta)}^{\infty}\int_{-\infty}^{\mathrm{TPR}^{-1}(\alpha)} p_+(s)p_-(t)\mathbb{I}(s>t)dsdt.\end{aligned}$$

This concludes the proof of the second part. □

Hence, an empirical estimator of OPAUC with FPR restricted in the range $[\alpha,\beta]$ can be computed by

$$\frac{1}{n_+ k_1} \sum_{\mathbf{x}_i \in \mathcal{S}_+} \sum_{\mathbf{x}_j \in \mathcal{S}_-^{\downarrow}[k_1+1, k_2]} \mathbb{I}(h(\mathbf{x}_+) > h(\mathbf{x}_-)), \tag{2.39}$$

where $k_1 = \lceil n_- \alpha \rceil$, $k_2 = \lfloor n_- \beta \rfloor$, and $\mathcal{S}^{\downarrow}[k_1, k_2] \subseteq \mathcal{S}$ denotes the subset of examples whose rank in terms of their prediction scores in the descending order are in the range of $[k_1, k_2]$.

An empirical estimator of TPUC with with FPR restricted in a range of $[0, \beta]$ and TPR restricted in a range of $[\alpha, 1]$ is computed by:

$$\frac{1}{k_1} \frac{1}{k_2} \sum_{\mathbf{x}_i \in \mathcal{S}_+^{\uparrow}[1, k_1]} \sum_{\mathbf{x}_j \in \mathcal{S}_-^{\downarrow}[1, k_2]} \mathbb{I}(h(\mathbf{w}; \mathbf{x}_i) > h(\mathbf{w}; \mathbf{x}_i)), \tag{2.40}$$

where $k_1 = \lceil n_+(1-\alpha) \rceil$, $k_2 = \lfloor n_- \beta \rfloor$, and $\mathcal{S}^{\uparrow}[k_1, k_2] \subseteq \mathcal{S}$ denotes the subset of examples whose rank in terms of their prediction scores in the ascending order are in the range of $[k_1, k_2]$.

Necessity of Maximizing partial AUC

In many applications, there are large monetary costs due to high false positive rates (FPR) and low true positive rates (TPR), e.g., in medical diagnosis. Hence, a measure of interest would be the pAUC- the region of the ROC curve corresponding to low FPR and/or high TPR. With a similar argument as last section, a model that maximizes AUC does not necessarily optimizes pAUC. Let us compare two models on a dataset with 2 positive and 100 negative molecules (Figure 2.4). The model 1 ranks two negatives above the two positives followed by the remaining 98 negatives. The model 2 ranks one positive at the top, and then four negatives above the other positive followed by the remaining 96 negatives. The two models have the same AUC score of $196/200 = 0.98$ but have different pAUC scores. When restricting FPR $\in [0, 0.02]$, model 1 has an empirical pAUC score of $\frac{0}{4} = 0$ and model 2 has an empirical pAUC score of $\frac{2}{4} = 0.5$ according to (2.39).

Critical: Partial AUC emphasize the correct order between the top ranked negative data and/or the bottom ranked positive data.

A Direct Formulation

Using a surrogate loss of zero-one loss, OPAUC maximization for learning a parameterized model $h(\mathbf{w}; \cdot)$ can be formulated as:

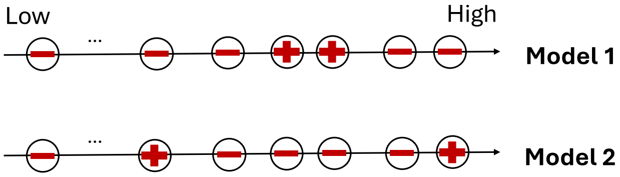

Fig. 2.4: Two models that have the same AUC score but differ dramatically in pAUC. The arrows indicate the prediction scores from low to high.

$$\min_{\mathbf{w}} \frac{1}{n_+}\frac{1}{k_2}\sum_{\mathbf{x}_i\in\mathcal{S}_+}\sum_{\mathbf{x}_j\in\mathcal{S}_-^{\downarrow}[1,k_2]} \ell(h(\mathbf{w};\mathbf{x}_j)-h(\mathbf{w};\mathbf{x}_i)). \tag{2.41}$$

Similarly, TPAUC maximization can be formulated as:

$$\min_{\mathbf{w}} \frac{1}{k_1}\frac{1}{k_2}\sum_{\mathbf{x}_i\in\mathcal{S}_+^{\uparrow}[1,k_1]}\sum_{\mathbf{x}_j\in\mathcal{S}_-^{\downarrow}[1,k_2]} \ell(h(\mathbf{w};\mathbf{x}_j)-h(\mathbf{w};\mathbf{x}_i)), \tag{2.42}$$

where $k_1 = \lfloor n_+(1-\alpha)\rfloor, k_2 = \lfloor n_-\beta\rfloor$.

Both problems are not standard ERM. The challenge for solving the above problems is that the selection of examples in a range, e.g., $\mathcal{S}_-^{\downarrow}[1,k_2]$ and $\mathcal{S}_+^{\uparrow}[1,k_1]$, is not only expensive but also non-differentiable. We will explore different approaches for optimizing OPAUC and TPUC in Section 6.4.3 using advanced compositional optimization techniques.

An Indirect Formulation

When the surrogate loss $\ell(t)$ is non-decreasing, the top-k selector of negative examples $\mathcal{S}_-^{\downarrow}[1,k_2]$ can be transferred into the top-k average of pairwise losses, which becomes an CVaR. By drawing the connection between CVaR and KL-regularized DRO, an indirect objective for OPAUC maximization is formulated by:

$$\min_{\mathbf{w}} \frac{1}{n_+}\sum_{\mathbf{x}_i\in\mathcal{S}_+}\tau\log\left(\sum_{\mathbf{x}_j\in\mathcal{S}_-}\exp\left(\frac{\ell(h(\mathbf{w};\mathbf{x}_j)-h(\mathbf{w};\mathbf{x}_i))}{\tau}\right)\right). \tag{2.43}$$

This problem is an instance of EXM, which will be solved by FCCO techniques. TPAUC maximization can be handled similarly. We will present detailed exposition in Chapter 6.4.3.

2.3.4 Ranking Losses

Ranking losses are commonly employed in learning to rank.

What is Learning to Rank?

Learning to rank (LTR) is a machine learning problem that aims to learn a ranking model, which can be used to predict the relevance order of a set of items given a query.

Let $\mathcal{Q}$ denote the query set of size N, and let $q \in \mathcal{Q}$ represent an individual query. For each query q, let $\mathcal{S}_q$ be a set of N_q items (e.g., documents, movies) to be ranked. For each item $\mathbf{x}_{q,i} \in \mathcal{S}_q$, let $y_{q,i} \in \mathbb{R}^+$ denote its relevance score, which quantifies the relevance between the query q and the item $\mathbf{x}_{q,i}$. Define $\mathcal{S}_q^+ \subseteq \mathcal{S}_q$ as the subset of N_q^+ items relevant to q, i.e., those with non-zero relevance scores. Let $\mathcal{S} = \{(q, \mathbf{x}_{q,i}) \mid q \in \mathcal{Q}, \mathbf{x}_{q,i} \in \mathcal{S}_q^+\}$ represent the collection of all relevant query-item (Q-I) pairs.

Let $s(\mathbf{w}; \mathbf{x}, q)$ denote the predicted relevance score for item $\mathbf{x}$ with respect to query q, parameterized by $\mathbf{w} \in \mathbb{R}^d$ (e.g., a deep neural network). Define the rank of item $\mathbf{x}$ within $\mathcal{S}_q$ as:

$$r(\mathbf{w}; \mathbf{x}, \mathcal{S}_q) = \sum_{\mathbf{x}' \in \mathcal{S}_q} \mathbb{I}(s(\mathbf{w}; \mathbf{x}', q) - s(\mathbf{w}; \mathbf{x}, q) \geq 0),$$

where ties are ignored.

NDCG and NDCG Loss

Normalized Discounted Cumulative Gain (NDCG) is a metric commonly used to evaluate the quality of ranking algorithms, especially in information retrieval and recommender systems.

NDCG evaluates how well a model ranks relevant items near the top of a list for a query q. The DCG of a ranked list according to $\{s(\mathbf{w}; \mathbf{x}, q), \mathbf{x} \in \mathcal{S}_q\}$ is given by:

$$\mathrm{DCG}_q = \sum_{\mathbf{x} \in \mathcal{S}_q} \frac{2^{y_i} - 1}{\log_2(1 + r(\mathbf{w}; \mathbf{x}, \mathcal{S}_q))} = \sum_{\mathbf{x} \in \mathcal{S}_q^+} \frac{2^{y_i} - 1}{\log_2(1 + r(\mathbf{w}; \mathbf{x}, \mathcal{S}_q))}.$$

Note that the summation is over $\mathcal{S}_q^+$ rather than $\mathcal{S}_q$, as only relevant items contribute to the DCG score due to their non-zero relevance.

NDCG normalizes DCG by the ideal DCG denoted by Z_q of the best possible ranking:

$$\mathrm{NDCG}_q = \frac{\mathrm{DCG}_q}{Z_q}.$$

The average NDCG over all queries is given by:

$$\text{NDCG:} \quad \frac{1}{N}\sum_{q=1}^{N}\frac{1}{Z_q}\sum_{\mathbf{x}_{q,i}\in\mathcal{S}_q^+}\frac{2^{y_{q,i}}-1}{\log_2(r(\mathbf{w};\mathbf{x}_{q,i},\mathcal{S}_q)+1)}, \tag{2.44}$$

where Z_q can be precomputed.

By replacing the indicator function with a surrogate function in Table 2.3, we approximate $r(\mathbf{w};\mathbf{x},\mathcal{S}_q)/N_q$ by

$$g(\mathbf{w};\mathbf{x},\mathcal{S}_q) = \frac{1}{N_q}\sum_{\mathbf{x}'\in\mathcal{S}_q}\ell(s(\mathbf{w};\mathbf{x}',q)-s(\mathbf{w};\mathbf{x},q)).$$

Then the NDCG loss minimization is defined by

$$\min_{\mathbf{w}}\frac{1}{N}\sum_{q=1}^{N}\frac{1}{Z_q}\sum_{\mathbf{x}_{q,i}\in\mathcal{S}_q^+}\frac{1-2^{y_{q,i}}}{\log_2(N_q g(\mathbf{w};\mathbf{x}_{q,i},\mathcal{S}_q)+1)}, \tag{2.45}$$

which is an instance of EXM. We will explore FCCO technique for solving this problem in Section 6.4.4.

Listwise Cross-Entropy Loss

Analogous to multi-class classification, we can define a listwise cross-entropy loss for ranking. This is based on modeling the probability that a specific item is ranked at the top:

$$P_{\text{top}}(\mathbf{x}\mid q) = \frac{\exp(s(\mathbf{w};\mathbf{x},q))}{\sum_{\mathbf{x}_j\in\mathcal{S}_q}\exp(s(\mathbf{w};\mathbf{x}_j,q))}. \tag{2.46}$$

Accordingly, the listwise cross-entropy loss for query q is defined as:

$$L(\mathbf{w};q) = \sum_{\mathbf{x}_{q,i}\in\mathcal{S}_q^+} -p_{q,i}\log\left(\frac{\exp(s(\mathbf{w};\mathbf{x}_{q,i},q))}{\sum_{\mathbf{x}_j\in\mathcal{S}_q}\exp(s(\mathbf{w};\mathbf{x}_j,q))}\right),$$

where $p_{q,i}$ denotes the top-one prior probability for item $\mathbf{x}_{q,i}$, such as

$$p_{q,i} = \frac{\exp(y_{q,i})}{\sum_{\mathbf{x}_{q,i}\in\mathcal{S}_q}\exp(y_{q,i})} \quad \text{or} \quad p_{q,i} = \frac{1}{N_q}.$$

An optimization objective based on the average of listwise cross-entropy losses over all queries leads to the following formulation known as ListNet:

$$\min_{\mathbf{w}} \quad \frac{1}{N}\sum_{q=1}^{N}\sum_{\mathbf{x}_{q,i}\in\mathcal{S}_q^+} p_{q,i}\log\left(\sum_{\mathbf{x}_j\in\mathcal{S}_q}\exp(s(\mathbf{w};\mathbf{x}_j,q)-s(\mathbf{w};\mathbf{x}_{q,i},q))\right). \tag{2.47}$$

This formulation closely resembles equation (2.43) and constitutes a special case of the EXM framework.

2.3.5 Contrastive Losses

Contrastive losses are commonly used in representation learning, which is a fundamental problem in the era of deep learning and modern AI.

What is Representation Learning?

Representation Learning is a process in machine learning where algorithms extract meaningful patterns from raw data (e.g., images) to create representations that are useful for many downstream tasks, e.g., learning a classifier or a retrieval model.

A deep neural network is usually used to extract representation from unstructured raw data. Let $h(\mathbf{w};\cdot) : \mathcal{X} \to \mathbb{R}^{d_1}$ denote the representation network that outputs an embedding vector, which is sometimes called the encoder. A meaningful encoder should capture the semantics such that 'similar' data points (positive pairs) are closer to each other and dissimilar data points (negative pairs) are far away from each other in the embedding space.

To conduct the representation learning, the following data is usually constructed. Let $\mathbf{x}_i$ be an anchor data, and let $\mathbf{x}_i^+$ denote a positive data of $\mathbf{x}_i$. Denote by $\mathcal{S}_i^-$ the set of negative data of $\mathbf{x}_i$. Let $s(\mathbf{w};\mathbf{x},\mathbf{y})$ denote a similarity score between the two encoded representations. For example, if $h(\mathbf{w};\mathbf{x})$ is a normalized vector such that $\|h(\mathbf{w};\mathbf{x})\|_2 = 1$, we can use $s(\mathbf{w};\mathbf{x},\mathbf{y}) = h(\mathbf{w};\mathbf{x})^\top h(\mathbf{w};\mathbf{y})$.

A contrastive loss for each positive pair $(\mathbf{x}_i, \mathbf{x}_i^+)$ is defined by:

$$L(\mathbf{w};\mathbf{x}_i,\mathbf{x}_i^+) = \tau \log\left(\frac{1}{|\mathcal{S}_i^-|}\sum_{\mathbf{y}\in\mathcal{S}_i^-} \exp((s(\mathbf{w};\mathbf{x}_i,\mathbf{y}) - s(\mathbf{w};\mathbf{x}_i,\mathbf{x}_i^+))/\tau)\right). \qquad (2.48)$$

where $\tau > 0$ is called the temperature parameter. Given a set of data $\{(\mathbf{x}_i, \mathbf{x}_i^+, \mathcal{S}_i^-)\}_{i=1}^n$, minimizing a contrastive objective for representation learning is formulated as:

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^n \tau \log\left(\frac{1}{|\mathcal{S}_i^-|}\sum_{\mathbf{y}\in\mathcal{S}_i^-} \exp((s(\mathbf{w};\mathbf{x}_i,\mathbf{y}) - s(\mathbf{w};\mathbf{x}_i,\mathbf{x}_i^+))/\tau)\right). \qquad (2.49)$$

A variant of the contrastive objective is to add a small constant $\varepsilon > 0$ inside the log:

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^n \tau \log\left(\varepsilon + \frac{1}{|\mathcal{S}_i^-|}\sum_{\mathbf{y}\in\mathcal{S}_i^-} \exp((s(\mathbf{w};\mathbf{x}_i,\mathbf{y}) - s(\mathbf{w};\mathbf{x}_i,\mathbf{x}_i^+))/\tau)\right). \qquad (2.50)$$

Traditional supervised representation learning methods construct the positive and negative data using the annotated class labels, such that data in the same class are deemed as positive and data from different classes are considered as negative. However, this requires a large amount of labeled data to learn the encoder, which requires significant human effort in labeling. To address this issue, self-supervised representation learning (SSRL) techniques are employed to fully exploit the vast data readily available on the internet via self-supervision (where $\mathbf{x}_i, \mathbf{x}_i^+$ are constructed from the data itself) to learn representations that are useful for many downstream tasks, which will be explored more in Section 6.5.

Optimization Challenge

Optimizing the above contrastive objectives is challenging due to the presence of summations both inside and outside the logarithmic function. These losses can be reformulated as special cases of X-risk, where the outer function is $f(g_i) = \tau \log(g_i)$, and g_i represents the inner average computed over negative samples associated with each $\mathbf{x}_i$.

2.4 Discriminative Data Prediction

The aforementioned X-risks can be unified under a principled discriminative learning framework for data prediction, providing a statistical foundation for developing advanced methods to train foundation models in modern AI.

What is a Foundation Model?

A foundation model (FM) is a type of machine learning model trained on large, diverse datasets (generally using self-supervision at scale) that can be adapted to a wide range of downstream tasks.

The widely used foundation models include Contrastive Language-image Pretrained (CLIP) model (see Section 6.5), Dense Passage Retrieval (DPR) model, large language models (LLMs) such as the Generative Pretrained Transformer (GPT) series (see Section 6.6), and vision-language models (VLMs). These models fall into two main categories: **representation models**, such as CLIP and DPR, and **generative models**, including LLMs and VLMs.

We present a discriminative data prediction framework to facilitate the learning of these foundation models. Suppose there exists a set of observed paired data, $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathcal{X}$ and $\mathbf{y}_i \in \mathcal{Y}$. These pairs typically represent real-world positive correspondences. While this setup resembles traditional supervised learning where $\mathbf{x}_i$ represents input data and $\mathbf{y}_i$ denotes a class label, there is a crucial difference: here, $\mathbf{y}_i$ refers to data from a **continuous space** (e.g., images) or an **uncountable space** (e.g., text). For instance:

- In training the CLIP model, $\mathbf{x}_i$ represents an image and $\mathbf{y}_i$ is the corresponding text caption (or vice versa).
- In training the DPR model, $\mathbf{x}_i$ is an input question, and $\mathbf{y}_i$ is the corresponding textual answer.
- In fine-tuning LLMs or VLMs, $\mathbf{x}_i$ represents input data (e.g., prompts or images), and $\mathbf{y}_i$ represents the text to be generated.

Discriminative Data Prediction

The problem of learning a representation model or fine-tuning a generative model can be framed as discriminative learning, which we term as data prediction, such that given any anchor data $\mathbf{x}$, the parameterized scoring function $s(\mathbf{w};\cdot,\cdot)$ is able to discriminate a positive data $\mathbf{y}$ from any other negative data $\mathbf{y}'$, i.e., $s(\mathbf{w};\mathbf{x},\mathbf{y}) \geq s(\mathbf{w};\mathbf{x},\mathbf{y}')$.

Since the risk function usually involves coupling each positive data with many other possibly negative data points in a compositional structure, the resulting risk is called discriminative X-risk. The following subsections detail two specific approaches to formulating discriminative X-risk.

2.4.1 A Discriminative Probabilistic Modeling Approach

Without loss of generality, we assume that $\mathcal{X}$ and $\mathcal{Y}$ are continuous spaces. Let $\mathbb{P}_{\mathcal{J}}$ denote the joint distribution of a pair $(\mathbf{x},\mathbf{y})$, and let $\mathbb{P}_1$ and $\mathbb{P}_2$ denote the marginal distributions of $\mathbf{x}$ and $\mathbf{y}$, respectively. We write their corresponding density functions as $p(\cdot,\cdot)$, $p_1(\cdot)$, and $p_2(\cdot)$. We denote the conditional density functions by $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{x}|\mathbf{y})$, corresponding to the conditional distributions $\mathbb{P}(\mathbf{y}|\mathbf{x})$ and $\mathbb{P}(\mathbf{x} \mid \mathbf{y})$. Below, we present two approaches based on discriminative probabilistic modeling (DPM)

Symmetric DPM

For symmetric DPM, we use $s(\mathbf{w};\mathbf{x},\mathbf{y})$ to model both conditional distributions $\mathbb{P}(\mathbf{y}|\mathbf{x})$ and $\mathbb{P}(\mathbf{x}|\mathbf{y})$. A discriminative probabilistic approach models the conditional probability $p(\mathbf{y}|\mathbf{x})$ using a scoring function $s(\mathbf{w};\mathbf{x},\mathbf{y})$ by:

$$p_{\mathbf{w}}(\mathbf{y}|\mathbf{x}) = \frac{p_2(\mathbf{y})\exp(s(\mathbf{w};\mathbf{x},\mathbf{y})/\tau)}{\int_{\mathbf{y}\in\mathcal{Y}} p_2(\mathbf{y})\exp(s(\mathbf{w};\mathbf{x},\mathbf{y}')/\tau)d\mathbf{y}'}, \tag{2.51}$$

where $\tau > 0$ is a temperature hyperparameter. The above parameterized distribution is the solution to the following problem for a fixed $\mathbf{x}$:

$$p_{\mathbf{w}}(\cdot|\mathbf{x}) = \arg\max_{\mathbb{Q}\in\mathcal{Q}} \mathbb{E}_{\mathbf{y}'\sim\mathbb{Q}} s(\mathbf{w};\mathbf{x},\mathbf{y}') - \tau\mathrm{KL}(\mathbb{Q},\mathbb{P}_2),$$

where $\mathcal{Q} = \{\mathbb{Q}|\mathbb{Q} \ll \mathbb{P}_2\}$ is a set of probability distributions over $\mathbf{y} \in \mathcal{Y}$.

Similarly, we model $p(\mathbf{x}|\mathbf{y})$ as

$$p_{\mathbf{w}}(\mathbf{x}|\mathbf{y}) = \frac{p_1(\mathbf{x})\exp(s(\mathbf{w};\mathbf{x},\mathbf{y})/\tau)}{\int_{\mathbf{x}\in\mathcal{X}} p_1(\mathbf{x})\exp(s(\mathbf{w};\mathbf{x}',\mathbf{y})/\tau)d\mathbf{x}'}. \tag{2.52}$$

Given a set of observed positive pairs $\{(\mathbf{x}_i,\mathbf{y}_i)\}_{i=1}^n$, the model parameters $\mathbf{w}$ are learned by minimizing the empirical risk of the negative log-likelihood:

$$\min_{\mathbf{w}} -\frac{1}{n}\sum_{i=1}^{n}\left\{\tau\log\frac{\exp(s(\mathbf{w};\mathbf{x}_i,\mathbf{y}_i)/\tau)}{\mathbb{E}_{\mathbf{y}'\sim\mathbb{P}_2}\exp(s(\mathbf{w};\mathbf{x}_i,\mathbf{y}')/\tau)} + \tau\log\frac{\exp(s(\mathbf{w};\mathbf{x}_i,\mathbf{y}_i)/\tau)}{\mathbb{E}_{\mathbf{x}'\sim\mathbb{P}_1}\exp(s(\mathbf{w};\mathbf{x}',\mathbf{y}_i)/\tau)}\right\}.$$

A significant challenge in solving this problem lies in handling the partition functions,

$$Z(\mathbf{x}_i) = \mathbb{E}_{\mathbf{y}'\sim\mathbb{P}_2}\exp(s(\mathbf{w};\mathbf{x}_i,\mathbf{y}')/\tau)d\mathbf{y}', \quad Z(\mathbf{y}_i) = \mathbb{E}_{\mathbf{x}'\sim\mathbb{P}_1}\exp(s(\mathbf{w};\mathbf{x}',\mathbf{y}_i)/\tau),$$

which are often computationally intractable. To overcome this, an approximation can be constructed using a set of samples $\hat{\mathcal{Y}}_i \subseteq \mathcal{Y}, \hat{\mathcal{X}}_i \subseteq \mathcal{X}$. The partition function is then estimated as:

$$\hat{Z}(\mathbf{x}_i) = \frac{1}{|\hat{\mathcal{Y}}_i|}\sum_{\hat{\mathbf{y}}_j\in\hat{\mathcal{Y}}_i}\exp(s(\mathbf{w};\mathbf{x}_i,\hat{\mathbf{y}}_j)/\tau), \quad \hat{Z}(\mathbf{y}_i) = \frac{1}{|\hat{\mathcal{X}}_i|}\sum_{\hat{\mathbf{x}}_j\in\hat{\mathcal{X}}_i}\exp(s(\mathbf{w};\hat{\mathbf{x}}_j,\mathbf{y}_i)/\tau).$$

Consequently, the resulting optimization problem is an empirical X-risk minimization problem:

$$\begin{aligned}\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n}\tau\log&\left(\sum_{\hat{y}_j\in\hat{\mathcal{Y}}_i}\exp\left(\frac{s(\mathbf{w};\mathbf{x}_i,\hat{\mathbf{y}}_j)-s(\mathbf{w};\mathbf{x}_i,\mathbf{y}_i)}{\tau}\right)\right)\\ &+\tau\log\left(\sum_{\hat{x}_j\in\hat{\mathcal{X}}_i}\exp\left(\frac{s(\mathbf{w};\hat{\mathbf{x}}_j,\mathbf{y}_i)-s(\mathbf{w};\mathbf{x}_i,\mathbf{y}_i)}{\tau}\right)\right).\end{aligned} \tag{2.53}$$

The above approach can be justified that if $s(\mathbf{w},\cdot,\cdot)$ is optimized over all possible scoring functions, then the learned $p_{\mathbf{w}}(\mathbf{y}|\mathbf{x})$ approaches the true density function of $\Pr(\mathbf{y}|\mathbf{x})$ when n approaches ∞.

Theorem 2.4 *Let us consider the following problem over all possible scoring functions $s(\cdot,\cdot)$:*

$$\min_{s} -\mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\tau\log\frac{p_2(\mathbf{y})\exp(s(\mathbf{x},\mathbf{y})/\tau)}{\mathbb{E}_{\mathbf{y}'\sim\mathbb{P}_2}\exp(s(\mathbf{x},\mathbf{y}')/\tau)} + \tau\log\frac{p_1(\mathbf{x})\exp(s(\mathbf{x},\mathbf{y})/\tau)}{\mathbb{E}_{\mathbf{x}'\sim\mathbb{P}_1}\exp(s(\mathbf{x}',\mathbf{y})/\tau)}\right]. \tag{2.54}$$

Then the set of global minimizers is given by

$$\mathcal{S}_* = \left\{ s : \frac{s(\mathbf{x},\mathbf{y})}{\tau} = \log \frac{p(\mathbf{x},\mathbf{y})}{p_1(\mathbf{x})p_2(\mathbf{y})} + const \right\},$$

and we have

$$\begin{aligned} p_s(\mathbf{y}|\mathbf{x}) &= \frac{p_2(\mathbf{y})\exp(s(\mathbf{x},\mathbf{y})/\tau)}{\int_{\mathbf{y}'\in\mathcal{Y}} p_2(\mathbf{y}')\exp(s(\mathbf{x},\mathbf{y}')/\tau)d\mathbf{y}'} = p(\mathbf{y}|\mathbf{x}), \\ p_s(\mathbf{x}|\mathbf{y}) &= \frac{\mathbb{P}_1(\mathbf{y})\exp(s(\mathbf{x},\mathbf{y})/\tau)}{\int_{\mathbf{x}'\in\mathcal{X}} \mathbb{P}_1(\mathbf{x})\exp(s(\mathbf{x}',\mathbf{y})/\tau)d\mathbf{y}'} = p(\mathbf{x}|\mathbf{y}). \end{aligned}$$

Proof. Let $\mathcal{F}_1$ be a class of functions $f_1(\mathbf{x},\mathbf{y}) : \mathcal{X}\times\mathcal{Y} \rightarrow \mathbb{R}$ such that $f_1(\mathbf{x},\mathbf{y}) \geq 0$ and $\int_{\mathbf{y}\in\mathcal{Y}} f_1(\mathbf{x},\mathbf{y}) = 1$, which induces a probability distribution $\mathbb{Q}_{1,x}(\cdot)$ over $\mathcal{Y}$ for any $\mathbf{x}$. Similarly, we define $f_2(\mathbf{x},\mathbf{y}) \in \mathcal{F}_2$ that induces a probability distribution $\mathbb{Q}_{2,y}(\cdot)$ over $\mathcal{X}$ for any $\mathbf{y}$

Let us define a problem:

$$\min_{f_1\in\mathcal{F}_1, f_2\in\mathcal{F}_2} \mathbb{E}_{\mathbf{x},\mathbf{y}}[-\log f_1(\mathbf{x},\mathbf{y}) - \log f_2(\mathbf{x},\mathbf{y})]$$

Since

$$\begin{aligned} &\mathbb{E}_{\mathbf{x},\mathbf{y}}[-\log f_1(\mathbf{x},\mathbf{y}) - \log f_2(\mathbf{x},\mathbf{y})] \\ &= \mathbb{E}_{\mathbf{x}}\mathbb{E}_{\mathbf{y}\sim\mathbb{P}(\cdot|\mathbf{x})}\left[-\log\frac{f_1(\mathbf{x},\mathbf{y})}{p(\mathbf{y}|\mathbf{x})} - \log p(\mathbf{y}|\mathbf{x})\right] \\ &+ \mathbb{E}_{\mathbf{y}}\mathbb{E}_{\mathbf{x}\sim\mathbb{P}(\cdot|\mathbf{y})}\left[-\log\frac{f_2(\mathbf{x},\mathbf{y})}{p(\mathbf{x}|\mathbf{y})} - \log p(\mathbf{y}|\mathbf{x})\right] \\ &= \mathbb{E}_{\mathbf{x}}[\mathrm{KL}(\mathbb{P}(\cdot|\mathbf{x}), \mathbb{Q}_{1,x}(\cdot))] + \mathbb{E}_{\mathbf{y}}[\mathrm{KL}(\mathbb{P}(\cdot|\mathbf{y}), \mathbb{Q}_{2,y}(\cdot))] + \mathrm{const}, \end{aligned}$$

where const is independent of f. Hence the minimizer $f_1^*(\mathbf{x},\mathbf{y})$ is equal to $p(\mathbf{y}|\mathbf{x})$ and the minimizer $f_2^*(\mathbf{x},\mathbf{y})$ is equal to $p(\mathbf{x}|\mathbf{y})$. As a result, for optimal $s_*(\cdot,\cdot)$ we require

$$\frac{p_2(\mathbf{y})\exp(s_*(\mathbf{x},\mathbf{y})/\tau)}{\int_{\mathcal{Y}} p_2(\mathbf{y})\exp(s_*(\mathbf{x},\mathbf{y}')/\tau)d\mathbf{y}'} = f_1^*(\mathbf{x},\mathbf{y}) = p(\mathbf{y}|\mathbf{x}) \tag{2.55}$$

$$\frac{p_1(\mathbf{x})\exp(s_*(\mathbf{x},\mathbf{y})/\tau)}{\int_{\mathcal{Y}} p_1(\mathbf{x})\exp(s_*(\mathbf{x}',\mathbf{y})/\tau)d\mathbf{x}'} = f_2^*(\mathbf{x},\mathbf{y}) = p(\mathbf{x}|\mathbf{y}). \tag{2.56}$$

From the first equation, we can derive that $s_*(\mathbf{x},\mathbf{y}) = \log\frac{p(\mathbf{y}|\mathbf{x})}{p_2(\mathbf{y})} + h_1(\mathbf{x})$, where $h_1(\mathbf{x})$ is any arbitrary function of $\mathbf{x}$. From the second equation, we can derive that $s_*(\mathbf{x},\mathbf{y}) = \log\frac{p(\mathbf{x}|\mathbf{y})}{p_1(\mathbf{x})} + h_2(\mathbf{x})$, where $h_2(\mathbf{y})$ is any arbitrary function of $\mathbf{y}$. As a result, the global minimizer $s_*(\mathbf{x},\mathbf{y})$ will be in the form of $\log\frac{p(\mathbf{x},\mathbf{y})}{p_1(\mathbf{x})p_2(\mathbf{y})} + \mathrm{const}$. □

One-sided DPM

If we are only interested in modeling $\mathbb{P}(\mathbf{y}|\mathbf{x})$, then we can consider one-sided DPM. We define the following parametric probability function to model $\mathbb{P}(\mathbf{y}|\mathbf{x})$:

$$p_{\mathbf{w}}(\mathbf{y}|\mathbf{x}) = \frac{\exp(s(\mathbf{w};\mathbf{x},\mathbf{y})/\tau)}{\int_{\mathcal{Y}} \exp(s(\mathbf{w};\mathbf{x},\mathbf{y}')/\tau) d\mu(\mathbf{y}')}, \tag{2.57}$$

where $\tau > 0$ is a temperature hyperparameter, and μ is the Lebesgue measure associated with the space $\mathcal{Y}$.

Given a set of observed positive pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ where $\mathbf{x}_i$ denotes an input and $\mathbf{y}_i$ denotes the output of the generative model, the model parameters $\mathbf{w}$ are learned by minimizing the empirical risk of the negative log-likelihood:

$$\min_{\mathbf{w}} -\frac{1}{n}\sum_{i=1}^{n} \tau \log \frac{\exp(s(\mathbf{w};\mathbf{x}_i,\mathbf{y}_i)/\tau)}{\int_{\mathcal{Y}} \exp(s(\mathbf{w};\mathbf{x}_i,\mathbf{y}')/\tau) d\mu(\mathbf{y}')}.$$

A significant challenge in solving this problem lies in handling the partition function,

$$Z_i = \int_{\mathcal{Y}} \exp(s(\mathbf{w};\mathbf{x}_i,\mathbf{y}')/\tau) d\mu(\mathbf{y}'),$$

which is often computationally intractable. To overcome this, an approximation can be constructed using a set of samples $\hat{\mathcal{Y}}_i \subseteq \mathcal{Y}$. The partition function is then estimated as:

$$\hat{Z}_i = \sum_{\hat{\mathbf{y}}_j \in \hat{\mathcal{Y}}_i} \frac{1}{q_j} \exp(s(\mathbf{w};\mathbf{x}_i,\hat{\mathbf{y}}_j)/\tau),$$

where q_j is an importance weight that accounts for the sample probability of $\hat{\mathbf{y}}_j$. Consequently, the empirical X-risk minimization problem is reformulated as:

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \tau \log\left(\sum_{\hat{\mathbf{y}}_j \in \hat{\mathcal{Y}}_i} \exp((s(\mathbf{w};\mathbf{x}_i,\hat{\mathbf{y}}_j) + \zeta_j - s(\mathbf{w};\mathbf{x}_i,\mathbf{y}_i))/\tau)\right),$$

where $\zeta_j = \tau \ln \frac{1}{q_j}$.

We can similarly justify the above approach by the following theorem.

Theorem 2.5 *Let us consider the following problem over all possible similarity score functions $s(\cdot,\cdot)$:*

$$\min_{s} -\mathbb{E}_{\mathbf{x},\mathbf{y}} \tau \log \frac{\exp(s(\mathbf{x},\mathbf{y})/\tau)}{\int_{\mathbf{y}'\in\mathcal{Y}} \exp(s(\mathbf{x},\mathbf{y}')/\tau) d\mu(\mathbf{y}')}. \tag{2.58}$$

Then the set of global minimizers is given by

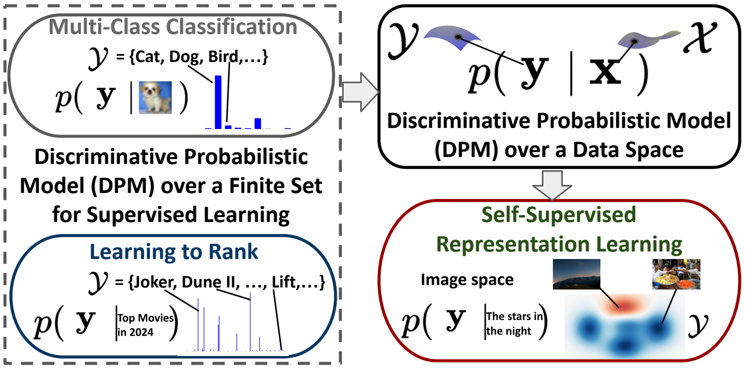
Multi-Class Classification
$\mathcal{Y}$ = {Cat, Dog, Bird,...}
$p($ y $|$ $)$
Discriminative Probabilistic Model (DPM) over a Finite Set for Supervised Learning
Learning to Rank
$\mathcal{Y}$ = {Joker, Dune II, ..., Lift,...}
$p($ y $|$ Top Movies in 2024 $)$
$\mathcal{Y}$
$p($ y $|$ x $)$
$\mathcal{X}$
Discriminative Probabilistic Model (DPM) over a Data Space
Self-Supervised Representation Learning
Image space
$p($ y $|$ The stars in the night $)$
$\mathcal{Y}$

Fig. 2.5: DPM for supervised learning and self-supervised representation learning.

$$\mathcal{S}_* = \left\{ s : \frac{s(\mathbf{x},\mathbf{y})}{\tau} = \log p(\mathbf{y}|\mathbf{x}) + h(\mathbf{x}) \right\},$$

where $h(\cdot)$ *is an arbitrary function of* $\mathbf{x}$*, and we have* $p_s(\mathbf{y}|\mathbf{x}) = \frac{\exp(s(\mathbf{x},\mathbf{y})/\tau)}{\int_{\mathcal{Y}} \exp(s(\mathbf{x},\mathbf{y}')/\tau) d\mathbf{y}'} = p(\mathbf{y}|\mathbf{x})$.

The proof is similar to the previous one and thus is omitted.

Instantiation

The fundamental difference between symmetric DPM and one-sided DPM lies in what their scoring functions $s(\mathbf{w};\mathbf{x},\mathbf{y})$ are designed to capture. We can use symmetric DPM for learning *representation models* and one-sided DPM for learning generative models and supervised prediction models.

The standard cross-entropy loss for classification and the listwise cross-entropy loss for learning to rank can both be viewed as special cases of the one-sided DPM framework, where $\mathcal{Y}$ represents either a finite set of class labels or a list of items to be ranked for each query. In these cases, the integral naturally simplifies to a finite summation, eliminating the need to approximate the normalization term Z_i. However, when $\mathcal{Y}$ is large, computing Z_i remains computationally demanding. This challenge, in turn, motivates the development of more advanced compositional optimization techniques.

For representation learning, the goal is to learn a symmetric score function $s(\mathbf{w};\mathbf{x},\mathbf{y}) = h_1(\mathbf{w};\mathbf{x})^\top h_2(\mathbf{w};\mathbf{y})$ that approximates the global optimum

$$s^*(\mathbf{x},\mathbf{y}) = \tau \log \frac{p(\mathbf{x},\mathbf{y})}{p_1(\mathbf{x})p_2(\mathbf{y})} + \text{const},$$

which measures how much the joint distribution $P(\mathbf{x},\mathbf{y})$ deviates from independence between $\mathbf{x}$ and $\mathbf{y}$. We will consider contrastive losses of CLIP in Section 6.5 for multi-modal representation learning, which can be interpreted by the symmetric DPM with $\mathbf{x}, \mathbf{y}$ denoting an image-text pair.

For generative modeling, we can use underlying models to induce a scoring function $s(\mathbf{w}; \mathbf{x}, \mathbf{y})$ for approximating the global optimum $s^*(\mathbf{x}, \mathbf{y}) = \tau \log p(\mathbf{y}|\mathbf{x}) + h(\mathbf{x})$. We will also consider discriminative fine-tuning of LLMs Section 6.6, which can be interpreted by the one-sided DPM with $\mathbf{x}, \mathbf{y}$ denoting an input-output pair.

An illustration of the connection between the probabilistic model for multi-modal representation learning and traditional supervised learning tasks including multi-class classification and learning to rank is shown in Figure 2.5.

Critical: Discriminative probabilistic model over a data space is a framework that unifies traditional label prediction and data ranking of supervised learning and modern self-supervised representation learning, and induces new approaches for fine-tuning LLMs.

2.4.2 A Robust Optimization Approach

The goal of discriminative learning is to increase the score $s(\mathbf{w}; \mathbf{x}, \mathbf{y}_+)$ for a "positive" pair $(\mathbf{x}, \mathbf{y}_+)$ while decreasing the score $s(\mathbf{w}; \mathbf{x}, \mathbf{y}_-)$ for any "negative" pair $(\mathbf{x}, \mathbf{y}_-)$.

Full Supervised setting

Let us first consider the supervised learning setting, where positive and negative samples are labeled, i.e., there is a function $r(\mathbf{x}, \mathbf{y}) \in (0, 1)$ that indicates whether they from a positive pair or a negative pair. We let $(\mathbf{x}, \mathbf{y}_+) \sim \mathbb{P}_+(\mathbf{x}, \mathbf{y}_+)$ denote a positive pair and $(\mathbf{x}, \mathbf{y}_-) \sim \mathbb{P}_-(\mathbf{x}, \mathbf{y}_-)$ denote a negative pair, where $\mathbb{P}_+(\mathbf{x}, \mathbf{y}_+) = \mathbb{P}(\mathbf{x})\mathbb{P}_+(\mathbf{y}_+|\mathbf{x}), \mathbb{P}_-(\mathbf{x}, \mathbf{y}_-) = \mathbb{P}(\mathbf{x})\mathbb{P}_-(\mathbf{y}_-|\mathbf{x})$, and $\mathbb{P}(\mathbf{x}, \mathbf{y}_+, \mathbf{y}_-) = \mathbb{P}_+(\mathbf{y}_+|\mathbf{x})\mathbb{P}_-(\mathbf{y}_-|\mathbf{x})\mathbb{P}(\mathbf{x})$. Let us denote a pairwise loss by $\ell(s(\mathbf{w}; \mathbf{x}, \mathbf{y}_-) - s(\mathbf{w}; \mathbf{x}, \mathbf{y}_+))$.

A naive goal is to minimize the expected risk:

$$\min_{\mathbf{w}} \mathbb{E}_{\mathbf{x}, \mathbf{y}_+, \mathbf{y}_- \sim \mathbb{P}(\mathbf{x}, \mathbf{y}_+, \mathbf{y}_-)} \left[\ell(s(\mathbf{w}; \mathbf{x}, \mathbf{y}_-) - s(\mathbf{w}; \mathbf{x}, \mathbf{y}_+)) \right].$$

However, a fundamental challenge for data prediction is that the number of negative data is usually much larger than the number of positive data. Hence, the expected risk is not a strong measure. To address this challenge, we can leverage DRO or OCE. In particular, we replace the expected risk $\mathbb{E}_{\mathbf{y}_- \sim \mathbb{P}(\mathbf{y}_-|\mathbf{x})} \left[\ell(s(\mathbf{w}; \mathbf{x}, \mathbf{y}_-) - s(\mathbf{w}; \mathbf{x}, \mathbf{y}_+)) \right]$ by its OCE counterpart, resulting the following population risk:

$$\min_{\mathbf{w}} \mathbb{E}_{\mathbf{x}, \mathbf{y}_+} \left[\min_{\nu} \tau \mathbb{E}_{\mathbf{y}_-|\mathbf{x}} \phi^* \left(\frac{\ell(s(\mathbf{w}; \mathbf{x}, \mathbf{y}_-) - s(\mathbf{w}; \mathbf{x}, \mathbf{y}_+)) - \nu}{\tau} \right) + \nu \right]. \tag{2.59}$$

If the training dataset is $\mathcal{S} = \{\mathbf{x}_i, \mathbf{y}_i^+, \mathbf{y}_{ij}^-, i \in [n], j \in [m]\}$, where $\mathbf{y}_i^+ \sim \mathbb{P}_+(\cdot|\mathbf{x}_i)$ and $\mathbf{y}_{ij}^- \sim \mathbb{P}_-(\cdot|\mathbf{x}_i)$, then the empirical version becomes:

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \min_{\nu_i} \tau \frac{1}{m} \sum_{j=1}^{m} \phi^* \left(\frac{\ell(s(\mathbf{w}; \mathbf{x}_i, \mathbf{y}_{ij}^-) - s(\mathbf{w}; \mathbf{x}_i, \mathbf{y}_i^+)) - \nu_i}{\tau} \right) + \nu_i. \tag{2.60}$$

Semi-supervised setting

We can extend the above framework to the semi-supervised learning setting, where we only have samples from the positive distribution $\mathbb{P}_+(\cdot|\mathbf{x})$ and samples from the distribution $P(\cdot|\mathbf{x})$.

Let us assume that $\mathbb{P}(\cdot|\mathbf{x}) = \pi_+(\mathbf{x})\mathbb{P}_+(\cdot|\mathbf{x}) + \pi_-(\mathbf{x})\mathbb{P}_-(\cdot|\mathbf{x})$ and $\pi_+(\mathbf{x}) \ll \pi_-(\mathbf{x})$. This means that for a fixed data $\mathbf{x}$, the sampled data $\mathbf{y} \sim P(\cdot|\mathbf{x})$ is mostly likely from the negative distribution $\mathbb{P}_-(\cdot|\mathbf{x})$. Hence, we can approximate $\mathbb{E}_{\mathbf{y}_- \sim \mathbb{P}_-(\cdot|\mathbf{x})}$ by $\mathbb{E}_{\mathbf{y} \sim \mathbb{P}(\cdot|\mathbf{x})}$. Hence, a population risk in the semi-supervised learning setting becomes

$$\min_{\mathbf{w}} \mathbb{E}_{\mathbf{x}, \mathbf{y}_+} \left[\min_{\nu} \tau \mathbb{E}_{\mathbf{y}|\mathbf{x}} \phi^* \left(\frac{\ell(s(\mathbf{w}; \mathbf{x}, \mathbf{y}) - s(\mathbf{w}; \mathbf{x}, \mathbf{y}_+)) - \nu}{\tau} \right) + \nu \right], \tag{2.61}$$

and its empirical version becomes

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \min_{\nu_i} \tau \frac{1}{m} \sum_{j=1}^{m} \phi^* \left(\frac{\ell(s(\mathbf{w}; \mathbf{x}_i, \mathbf{y}_{ij}) - s(\mathbf{w}; \mathbf{x}_i, \mathbf{y}_i^+)) - \nu_i}{\tau} \right) + \nu_i, \tag{2.62}$$

where $\{\mathbf{y}_{ij}, j = 1, \ldots, m\}$ are samples from $\mathbb{P}(\cdot|\mathbf{x})$.

Self-supervised setting

For self-supervised learning, we let $(\mathbf{x}, \mathbf{y}^+) \sim \mathbb{P}(\mathbf{x}, \mathbf{y}^+)$ denote a "positive" pair, and $(\mathbf{x}, \mathbf{y}^-) \sim \mathbb{P}(\mathbf{x})\mathbb{P}(\mathbf{y}^-)$ denote a "negative" pair. For empirical learning, we only have a training set of $\mathcal{S} = \{\mathbf{x}_i, \mathbf{y}_i^+, i \in [n]\}$. We use $\mathcal{S}_i^- = \{\mathbf{y}_j^+\}_{j \neq i}$ to define the empirical risk:

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \min_{\nu_i} \tau \frac{1}{m} \sum_{\mathbf{y}' \in \mathcal{S}_i^-} \phi^* \left(\frac{\ell(s(\mathbf{w}; \mathbf{x}_i, \mathbf{y}') - s(\mathbf{w}; \mathbf{x}_i, \mathbf{y}_i^+)) - \nu_i}{\tau} \right) + \nu_i, \tag{2.63}$$

We refer to the problems in (2.60), (2.62) and (2.63) as the Compositional OCE (COCE) optimization. We will present and analyze stochastic algorithms for solving COCE optimization in Chapter 5[Section 5.5].

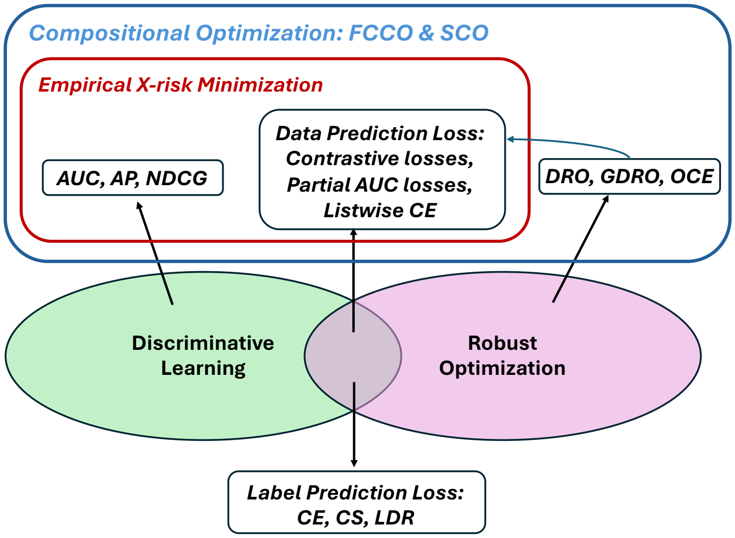
Compositional Optimization: FCCO & SCO
Empirical X-risk Minimization
AUC, AP, NDCG
Data Prediction Loss: Contrastive losses, Partial AUC losses, Listwise CE
DRO, GDRO, OCE
Discriminative Learning
Robust Optimization
Label Prediction Loss: CE, CS, LDR

Fig. 2.6: Overview of different losses and two fundamental learning principles

Instantiation

When $\phi(t) = t\log t - t + 1$, the inner optimization over v_i in (2.62) admits a closed-form solution, which can be substituted back into the objective, yielding:

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \tau \log\left(\frac{1}{m}\sum_{j=1}^{m} \exp\left(\frac{\ell(s(\mathbf{w};\mathbf{x}_i,\mathbf{y}_{ij}) - s(\mathbf{w};\mathbf{x}_i,\mathbf{y}_i^+))}{\tau}\right)\right). \tag{2.64}$$

This formulation unifies several well-known losses as special cases:

- **Cross-Entropy Loss for Classification**: Let $\mathbf{x}_i$ denote an input data point, let y_i^+ represent its true class label and $\{y_{ij}, j = 1, \ldots\} = \{1, \ldots, K\}$ forms the full label space. Define the prediction score for the y-th class of $\mathbf{x}$ as $s(\mathbf{w};\mathbf{x},y) = h_0(\mathbf{w}_0;\mathbf{x})^\top \mathbf{w}_y$. When the loss function is $\ell(s) = s$ and $\tau = 1$, the objective reduces to the empirical risk with the standard cross-entropy loss.
- **Listwise Cross-Entropy Loss for Ranking**: Let $\mathbf{x}_i$ denote a query, $\{\mathbf{y}_i^+\}$ denote a relevant (positive) document, and $\{\mathbf{y}_{ij}\}_{j=1}^m$ the complete candidate list to be ranked. Let $s(\mathbf{w};\mathbf{x},\mathbf{y})$ be the predicted relevance score between a query $\mathbf{x}$ and a document $\mathbf{y}$. When the loss function is $\ell(s) = s$ and $\tau = 1$, the objective simplifies to the listwise cross-entropy loss.
- **Self-supervised Contrastive Loss for Representation Learning**: If $\mathbf{x}_i$ is an anchor (e.g., an image), $\mathbf{y}_i^+$ denotes its positive pair (e.g., the corresponding text) and $\{\mathbf{y}_{i,j}, j = 1, \ldots, m\} = \mathcal{S}_i^-$, the the objective in (2.64) recovers the contrastive loss (2.48) used in self-supervised contrastive representation learning.

- **Partial AUC Loss for Imbalanced Binary Classification**: Let $\mathbf{x}_i$ be a fixed class label ($i = 1$), with $\{\mathbf{y}_i^+\}$ denoting its positive data set and $\{\mathbf{y}_{ij}\}_{j=1}^m$ its negative data set. Define the score function as $s(\mathbf{w}; \mathbf{x}, \mathbf{y}) = h(\mathbf{w}; \mathbf{y}) \in \mathbb{R}$. Under this setting, the objective in (2.64) reduces to the partial AUC loss in (2.43).

This framework offers a flexible foundation for designing alternative contrastive objectives by varying the loss function $\ell(\cdot)$, the temperature τ, the divergence function $\phi(\cdot)$, and the distributionally robust optimization (DRO) formulation, including its constrained variants.

Finally, Figure 2.6 illustrates the losses, objectives, and learning frameworks discussed in this chapter, along with their connections to the principles of discriminative learning and robust optimization. This perspective highlights the necessity of stochastic compositional optimization and finite-sum coupled compositional optimization, which will be presented in subsequent chapters.

2.5 History and Notes

Loss functions

A pioneering work analyzing the infinite-sample consistency of various multi-class surrogate loss functions is provided by Zhang (2004b). This work proves the consistency of several losses, including the cross-entropy loss. It also shows that the consistency of the Crammer-Singer and hinge losses can fail unless the maximum conditional probability of a class label given the input exceeds 0.5.

The Label-Distribution-Aware Margin (LDAM) Loss was proposed and studied by Cao et al. (2019), inspired by margin-based generalization error bounds tailored for each class. The label distributionally robust (LDR) losses and their consistency was proposed and studied by Zhu et al. (2023b).

Variants of standard loss functions have been developed to minimize the top-k error for $k > 1$, such as the top-k SVM loss and the top-k cross-entropy loss (Lapin et al., 2018; Yang and Koyejo, 2020). The top-k SVM loss can be recovered as a special case of the general LDR loss by setting $R(\mathbf{p}) = 0$ and $\Omega = \{\mathbf{p} \in \Delta_K : p_k \leq 1/k\}$. Although this formulation is generally inconsistent, adding a small strongly convex regularizer $R(\mathbf{p})$ to the LDR loss can restore consistency.

A sufficient condition for a loss function to be noise-tolerant is the symmetry property, as introduced by Ghosh et al. (2017). A loss function is considered noise-tolerant if the minimizer of the expected risk under the true label distribution remains the same under the noisy label distribution, provided the noise level is not excessively high.

Robust optimization

Robust optimization dates back to Scarf (1958), who studied an inventory problem in which the goal is to determine the purchase quantity that maximizes profit when future demand is a random variable whose underlying probability distribution is assumed to belong to a set of plausible distributions. The problem is reformulated as a worst-case analysis over all distributions in this set with known mean and variance. Later, Dupačová (1966) investigated the min−max robust formulation of stochastic linear programming. Since then, robust optimization has been extensively studied in management science, operations research, and mathematical programming (Kouvelis and Yu, 1997; Shapiro and Kleywegt, 2002; Rustem and Howe, 2002; Ben-Tal et al., 2009b). The term *distributionally robust optimization* was introduced by Delage and Ye (2010).

The ϕ-divergence (sometimes called f-divergence, where both f and ϕ denote a function) was introduced by Csiszár (1967). The use of ϕ-divergence to define the uncertainty set in robust optimization was first studied by Ben-Tal et al. (2013), while earlier works had considered using the KL divergence to define an uncertainty set of probabilities (Calafiore, 2007). A special case of DRO, namely the maximal loss, was shown to be beneficial for imbalanced classification by Shalev-Shwartz and Wexler (2016). The popularity of DRO in machine learning is largely attributed to Namkoong and Duchi (2017), who established a variance-based generalization error bound for DRO with the χ^2 divergence, building on their preceding work (Duchi et al., 2022). The optimized certainty equivalent (OCE) was proposed by Ben-Tal and Teboulle (1986b), and its connection to DRO was later established in (Ben-Tal and Teboulle, 2007). Group DRO was first proposed by Hu et al. (2018) and became widely recognized due to Sagawa et al. (2019).

AUC and NDCG

The receiver operating characteristic (ROC) curve was originally developed in the 1940s by electrical and radar engineers during World War II to detect enemy objects on the battlefield, which gave rise to its name (”receiver operating characteristic”) (Marcum, 1947). It was subsequently formalized within the framework of signal detection theory (Green and Swets, 1966). The probabilistic interpretation of AUC and its equivalence to the Mann–Whitney U-statistic (or Wilcoxon statistic) were later established by Hanley and McNeil (1982). The concept was subsequently introduced into machine learning as a standard metric for evaluating learning algorithms (Spackman, 1989). The first study of the one-way partial AUC (pAUC) was presented by Dodd and Pepe (2003), and the notion of two-way partial AUC was later introduced by Yang et al. (2019).

The study of AUC maximization dates back to Verrelst et al. (1998) and has since been extensively explored in machine learning. Yan et al. (2003) were the first to apply the gradient descent method to optimize a hinge-based pairwise surrogate loss for AUC, while Cortes and Mohri (2003) employed the Rankboost algorithm (Freund et al., 2003) to optimize AUC. The compositional objective for AUC maximization was first proposed by Ying et al. (2016a) in a min–max form and was later generalized in (Yuan et al., 2021; Zhu et al., 2022c). For a comprehensive overview of related work, see the survey by Yang and Ying (2022). The first work on maximizing average precision was conducted by Morgan et al. (2004). The use of DRO for formulating partial AUC losses was proposed by Zhu et al. (2022a).

NDCG was introduced by Järvelin and Kekäläinen (2000), and the listwise cross-entropy loss for learning to rank was proposed by Cao et al. (2007). The concept of empirical X-risk minimization for unifying a family of non-decomposable losses was developed by the author of this book in (Yang, 2022), which also presents additional examples of X-risks.

Foundation Models

Representation learning in traditional machine learning is related to principal component analysis and distance metric learning (Yang and Jin, 2006). Conventional contrastive losses are defined on pairs $(\mathbf{x}, \mathbf{y})$ using it binary label indicating positive or negative pair (Hadsell et al., 2006) or triplets $(\mathbf{x}, \mathbf{y}_+, \mathbf{y}_-)$ (Weinberger and Saul, 2009). The Contrastive loss defined on a list of negative data for a positive pair was first introduced by Sohn (2016).

The term foundation model was introduced by Bommasani et al. (2021). The use of DRO to formulate the contrastive loss was first proposed by Qiu et al. (2023), providing a principled approach for optimizing individualized temperature parameters. The discriminative probabilistic modeling approach for self-supervised representation learning was first explored by Wang et al. (2025).

Generalization Error

Generalization error analysis is a central topic in several classical machine learning texts (Shalev-Shwartz and Ben-David, 2014; Mohri et al., 2018) and in the statistical learning theory literature (Koltchinskii, 2011). Typically, uniform convergence bounds of the form $\sup_{\mathbf{w}\in\mathcal{W}} |\mathcal{R}(\mathbf{w}) - \mathcal{R}_S(\mathbf{w})|$ are derived using concentration inequalities, with dependencies on both the number of training samples n and the complexity of the hypothesis class. More recently, there has been growing interest in directly analyzing the generalization performance of models returned by stochastic optimization algorithms using stability-based techniques (Hardt et al., 2016; Lei and Ying, 2019).

Generalization error analyses for DRO and OCE objectives have been extensively developed in the literature: Brown (2007) established theoretical bounds for CVaR, Namkoong and Duchi (2017) developed bounds for χ^2-constrained DRO, and Lee et al. (2020) explored generalization for general OCE risk. However, the generalization error for compositional OCE is under-development.

Machine Learning texts

There are excellent textbooks on machine learning (Shalev-Shwartz and Ben-David, 2014; Mohri et al., 2018; Bishop, 2006) and on robust optimization (Ben-Tal et al., 2009a). However, to the best of our knowledge, this book is the first to provide a comprehensive and unified treatment of diverse loss functions and objectives, ranging from the traditional cross-entropy loss to the contrastive loss used in self-supervised representation learning, through the lens of robust optimization and discriminative learning.