Mathematical Introduction to Machine Learning

Lecture 3: Classification

October 14, 2025

Lecturer: Lei Wu *Scribe: Lei Wu*

1 Problem Setup

In classification problems, we are given n samples $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$ with $x_i \in \mathcal{X}$ and $y_i \in \{1, 2, \dots, C\}$, where C denotes the number of classes. Unlike regression, the labels are discrete (category variable). Our goal is to learn a classifier

$$f : \mathcal{X} \to \mathcal{Y} := \{1, \dots, C\} =: [C],$$

to minimize the population *classification error*

$$\mathcal{R}(f) = \mathbb{E}_{\mathbf{x},y}[\ell_{0\text{-}1}(f(\mathbf{x}), y)], \qquad \ell_{0\text{-}1}(y, y') = \begin{cases} 0, & \text{if } y = y', \\ 1, & \text{otherwise.} \end{cases}.$$

In practice, we only have access to finitely many training samples. Hence, we minimize the *empirical risk* with a regularization term to control model complexity:

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell_{0\text{-}1}\big(f(\mathbf{x}_i), y_i\big) + \lambda\, \Omega(f), \tag{1}$$

where $\mathcal{F}$ is the hypothesis class and $\Omega : \mathcal{F} \to \mathbb{R}_{\geq 0}$ is a regularizer that penalizes model complexity.

Unfortunately, the problem (1) is difficult to solve because the 0–1 loss is discontinuous and non-convex.

The central question in classification is to design an appropriate surrogate loss *to replace* $\ell_{0\text{-}1}$.

Here, "appropriate" means that (1) the resulting optimization problem is computationally tractable, and (2) minimizing the surrogate loss also leads to a small 0–1 loss. There is no single universal principle for designing such surrogate losses. In this lecture, we introduce two widely used approaches:

- **Probabilistic modeling:** the softmax classifier;
- **Geometric modeling:** the max-margin classifier.

In classification tasks, one is often concerned not only with whether the prediction is correct, but also with the **confidence** of the decision. The two approaches above model this confidence in fundamentally different ways: the probabilistic approach interprets it as a predictive probability, whereas the geometric approach quantifies it by the decision margin.

Question 1.1. Is the square loss $\ell(f(\mathbf{x}), y) = |f(\mathbf{x}) - y|^2$ a suitable choice for classification problems?

2 The Probabilistic Modeling: Capturing the Structure of Label Space via the Softmax Classifier

In many situations, it is natural to model the label as a *probability distribution* over the C classes, rather than as a single deterministic category. This is especially useful when the data from different classes are not well separated, as illustrated in Figure 1(left). For data points lying near class boundaries, assigning soft probabilistic labels better reflects the underlying uncertainty. Such ambiguity also arises frequently in real-world scenarios. For example, as shown in Figure 1(right), an image may simultaneously contain both a 'dog' and a 'car', making a probabilistic label representation clearly more appropriate.

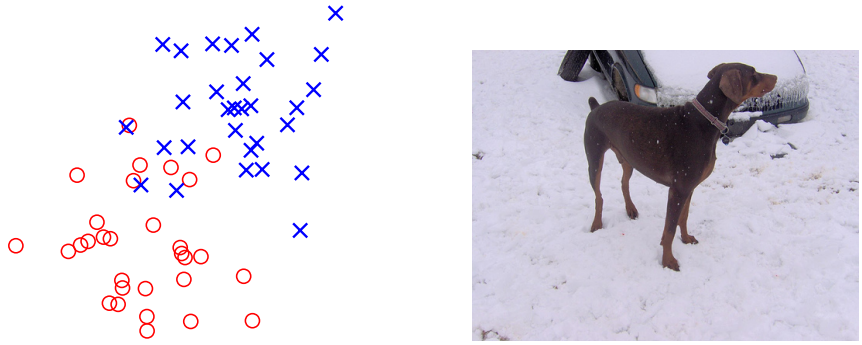

Figure 1: **Left:** Illustration of two classes that are not well separated. **Right:** An image containing both a dog and a car, whose label is better represented as a probability distribution over classes.

In probabilistic modeling of classification, the "true label" of a data point $\mathbf{x}$ is a discrete probability distribution over the C classes:

$$p(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_C(\mathbf{x})) \in \Delta^{C-1},$$

where

$$\Delta^{C-1} = \left\{ p \in \mathbb{R}^C \,:\, \sum_j p_j = 1, \min_j p_j \geq 0 \right\}$$

denotes the $(C-1)$-dimensional probability simplex—the set of all discrete probability distributions over C classes.

In practice, however, the *observed label* $y \in [C]$ is a single sample drawn from the multinomial distribution parameterized by $p(\mathbf{x})$, i.e., $y \sim p(\mathbf{x})$. A closely related notion is the *one-hot label*:

$$\mathbf{e}_y = (0, \ldots, 0, 1, 0, \ldots, 0) \in \Delta^{C-1}$$

where the nonzero entry appears in the y-th position. This can be viewed as a discrete delta distribution, providing a point-mass approximation of the true label distribution $p(\mathbf{x})$ based on a single observation

Any C-dimensional vector can be converted to a discrete probability distribution by using the *softmax operator*: $\mathbb{R}^C \to \Delta^{C-1}$,

$$\text{softmax}(\mathbf{z}) = \left(\frac{e^{z_1/T}}{\sum_j e^{z_j/T}}, \frac{e^{z_2/T}}{\sum_j e^{z_j/T}}, \ldots, \frac{e^{z_C/T}}{\sum_j e^{z_j/T}} \right),$$

where $T > 0$ is a hyperparameter, referred to as the *temperature*, that controls the softness of the resulting distribution. Obviously, when $T \to 0$, we have

$$\mathrm{softmax}(\mathbf{z}) \to \mathbf{e}_k \qquad k = \operatorname{argmax} z_j.$$

Given any parametrized model $f_\theta : \mathcal{X} \to \mathbb{R}^C$, we can define the corresponding classifier as

$$F(\mathbf{x}; \theta) = \mathrm{softmax}(f_\theta(\mathbf{x})) : \mathcal{X} \to \Delta^{C-1}.$$

Thus, learning a classifier can be formulated as the optimization problem

$$\min_\theta \ \frac{1}{n} \sum_{i=1}^{n} d\big(F(\mathbf{x}_i; \theta),\ \mathbf{e}_{y_i}\big) + \lambda\, \Omega(\theta),$$

where $\mathbf{e}_{y_i}$ denotes the one-hot label associated with $\mathbf{x}_i$, and

$$d : \Delta^{C-1} \times \Delta^{C-1} \to \mathbb{R}_{\geq 0}$$

is a distance (or divergence) function measuring the discrepancy between two probability distributions.

A variety of metrics exist to quantify the difference between two probability measures. A fundamental and widely used measure in statistics, machine learning, and information theory is the Kullback-Leibler (KL) divergence, also known as relative entropy. For discrete distributions $p, q \in \Delta^{C-1}$, it is defined as:

$$D_{\mathrm{KL}}(p \,\|\, q) = \sum_{j=1}^{C} p_j \log\left(\frac{p_j}{q_j}\right).$$

While the KL divergence captures information-theoretic discrepancies, one might wonder if simpler geometric metrics are suitable. This leads to a natural question:

Question 2.1. Is it a good idea to use metrics like $d(p, q) = \|p - q\|_2$ for $p, q \in \Delta^{C-1}$?

2.1 The KL Divergence and Cross Entropy

For two probability measures P and Q on the same measurable space, the KL divergence is defined as

$$D_{\mathrm{KL}}(P \,\|\, Q) = \int \log\left(\frac{\mathrm{d}P}{\mathrm{d}Q}\right) \mathrm{d}P,$$

where $\frac{\mathrm{d}P}{\mathrm{d}Q}$ denotes the Radon–Nikodým derivative of P with respect to Q.

- If P and Q are discrete distributions with probability mass functions (p_j) and (q_j), then

$$D_{\mathrm{KL}}(P \,\|\, Q) = \sum_{j} p_j \log\left(\frac{p_j}{q_j}\right).$$

- If P and Q have densities $p(\mathbf{x})$ and $q(\mathbf{x})$ with respect to the Lebesgue measure, then

$$D_{\mathrm{KL}}(P \,\|\, Q) = \int p(\mathbf{x})\, \log\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) \mathrm{d}\mathbf{x}.$$

Some basic properties of the KL divergence are as follows:

- **Asymmetry:** $D_{\mathrm{KL}}(P \,\|\, Q) \neq D_{\mathrm{KL}}(Q \,\|\, P)$ in general;
- **Non-negativity:** $D_{\mathrm{KL}}(P \,\|\, Q) \geq 0$ for all P, Q;
- **Identity of indiscernibles:** $D_{\mathrm{KL}}(P \,\|\, Q) = 0$ if and only if $P = Q$ almost surely;
- **Joint convexity:**

$$D_{\mathrm{KL}}\big(\lambda P_1 + (1-\lambda)P_2 \,\|\, \lambda Q_1 + (1-\lambda)Q_2\big) \leq \lambda D_{\mathrm{KL}}(P_1 \,\|\, Q_1) + (1-\lambda) D_{\mathrm{KL}}(P_2 \,\|\, Q_2).$$

- **Relation to (cross-)entropy:**

$$D_{\mathrm{KL}}(P \,\|\, Q) = -\sum_j p_j \log q_j + \sum_j p_j \log p_j = H(P, Q) - H(P),$$

  where $H(P, Q)$ denotes the *cross-entropy* between P and Q, and $H(P)$ the entropy of P.

2.2 Softmax Classifier

For simplicity, we first consider a single sample $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \Delta^{C-1}$, where $\mathbf{y}$ denotes a general (possibly soft) label distribution. The KL divergence between $\mathbf{y}$ and the model prediction $F(\mathbf{x}; \theta)$ is

$$D_{\mathrm{KL}}\big(\mathbf{y} \,\|\, F(\mathbf{x}; \theta)\big) = \sum_{j=1}^{C} y_j \log\left(\frac{y_j}{F_j(\mathbf{x}; \theta)}\right) = \sum_{j=1}^{C} y_j \log y_j \;-\; \sum_{j=1}^{C} y_j \log F_j(\mathbf{x}; \theta). \tag{2}$$

The first term represents the (negative) entropy of the label distribution and is independent of θ. Hence, minimizing the KL divergence is equivalent to minimizing the *cross-entropy loss*—the second term.

Given a dataset $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ with $\mathbf{y}_i = (y_{i,1}, \ldots, y_{i,C}) \in \Delta^{C-1}$ denoting the label distribution of the i-th sample, the empirical objective becomes

$$\min_\theta \; \frac{1}{n} \sum_{i=1}^{n} \Big(-\sum_{j=1}^{C} y_{i,j} \; \log F_j(\mathbf{x}_i; \theta) \Big). \tag{3}$$

When the label is one-hot, i.e., $\mathbf{y}_i = \mathbf{e}_{y_i}$, the objective simplifies to

$$\min_\theta \; -\frac{1}{n} \sum_{i=1}^{n} \log F_{y_i}(\mathbf{x}_i; \theta), \tag{4}$$

where, by a slight abuse of notation, y_i denotes the class index.

Question 2.2. Can we instead minimize $D_{\mathrm{KL}}\big(F(\mathbf{x}; \theta) \,\|\, \mathbf{y}\big)$?

We have seen that minimizing the KL divergence provides an information-theoretic justification for the cross-entropy loss. Next, we show that the same loss can also be derived from a statistical perspective, namely, the principle of *maximum likelihood estimation (MLE)*.

Maximum likelihood estimation (MLE). Let P_θ denote a parametric family of probability distributions, and let $\mathbf{z}_1, \ldots, \mathbf{z}_n$ be n i.i.d. samples drawn from P_θ. The goal is to estimate θ from the data. The *maximum likelihood estimator (MLE)* is defined as

$$\hat{\theta}_{\mathrm{MLE}} = \operatorname*{argmax}_\theta \prod_{i=1}^{n} P_\theta(\mathbf{z}_i), \tag{5}$$

that is, the parameter that maximizes the probability of generating the observed data. Taking logarithms gives the equivalent *log-likelihood* form:

$$\hat{\theta}_{\mathrm{MLE}} = \operatorname*{argmax}_\theta \sum_{i=1}^{n} \log P_\theta(\mathbf{z}_i), \tag{6}$$

Question 2.3. Why is the log-likelihood form (6) often preferred to (5)?

In classification problems, each sample $\mathbf{z} = (\mathbf{x}, y) \in \mathcal{X} \times [C]$ is generated as

$$P_\theta(\mathbf{z}) = P(\mathbf{x})\, F_y(\mathbf{x}; \theta),$$

where $P(\mathbf{x})$ is the marginal distribution of inputs, and $F_y(\mathbf{x}; \theta)$ is the predicted conditional probability of class y given $\mathbf{x}$. Hence,

$$\sum_{i=1}^{n} \log P_\theta(\mathbf{z}_i) = \sum_{i=1}^{n} \log\big(P(\mathbf{x}_i)\, F_{y_i}(\mathbf{x}_i; \theta)\big) = C + \sum_{i=1}^{n} \log F_{y_i}(\mathbf{x}_i; \theta),$$

where $C = \sum_{i=1}^{n} \log P(\mathbf{x}_i)$ is independent of θ. Therefore, MLE also gives (4), i.e., maximizing the likelihood is equivalent to minimize the cross-entropy.

It is worth noting, however, that the MLE derivation above relies on the i.i.d. assumption, whereas the divergence-based derivation does not. The divergence perspective is more flexible, as it explicitly captures the probabilistic structure of the label space, allowing straightforward extensions of the classifier.

- **Alternative mapping operators.** Can we replace the softmax with another operator that maps $\mathbb{R}^C$ into Δ^{C-1}? For instance, $s(\mathbf{z}) = \left(\frac{|z_1|^\alpha}{\sum_j |z_j|^\alpha}, \ldots, \frac{|z_C|^\alpha}{\sum_j |z_j|^\alpha}\right)$, for some $\alpha > 0$. The answer is affirmative; however, the softmax operator generally performs better in practice, while such alternatives may be useful in specific applications.
- **Alternative divergence/divergence measures.** The KL divergence can be replaced by other divergence functions between probability distributions (e.g., ℓ_1, ℓ_2, Jensen–Shannon divergence).
- **Encoding prior knowledge.** Prior information about the output probabilities can be incorporated naturally, as illustrated next.

Entropy regularization and label smoothing. When the classification task involves substantial uncertainty, it is undesirable for the model to produce overly confident (near-delta) output distributions. A natural idea is to encourage higher-entropy predictions:

$$\min_\theta \; \frac{1}{n} \sum_{i=1}^{n} D_{\mathrm{KL}}\big(\mathbf{y}_i \,\|\, F(\mathbf{x}_i; \theta)\big) - \lambda\, H(F(\mathbf{x}_i; \theta))\,,$$

where $\lambda > 0$ controls the strength of regularization. Computing and optimizing the entropy term can be inconvenient in practice. A simpler and widely used alternative is *label smoothing*, which replaces the one-hot labels with *smoothed labels*, given by

$$S_\delta : (0, \ldots, 0, 1, 0, \ldots, 0) \mapsto \left(\frac{\delta}{C-1}, \ldots, \frac{\delta}{C-1}, 1-\delta, \frac{\delta}{C-1}, \ldots, \frac{\delta}{C-1} \right),$$

where $\delta > 0$ is a small constant chosen so that the resulting vector remains a valid probability distribution. Training with smoothed labels implicitly encourages higher-entropy predictions, achieving a similar effect to entropy regularization while remaining simple and effective in implementation [Müller et al., 2019].

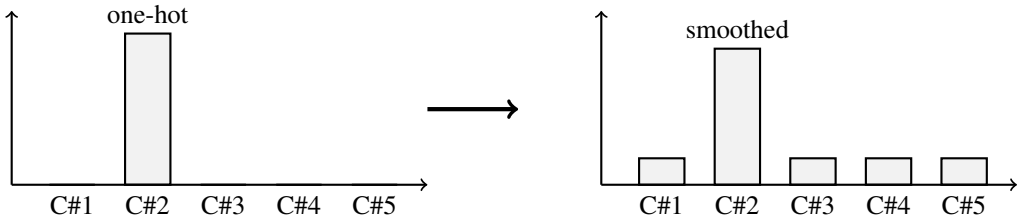

Figure 2: Illustration of label smoothing.

3 The Geometric Modeling: SVM and the Max-Margin Principle

When most data points are well separated, the classifier's predictions are highly confident, behaving almost like delta functions. In such cases, probabilistic modeling becomes unnecessary, as there is little uncertainty to capture. The *support vector machine* (SVM) instead quantifies classification confidence through the concept of a *margin*. For simplicity, we focus in this section on binary classification with labels $y \in \{-1, 1\}$.

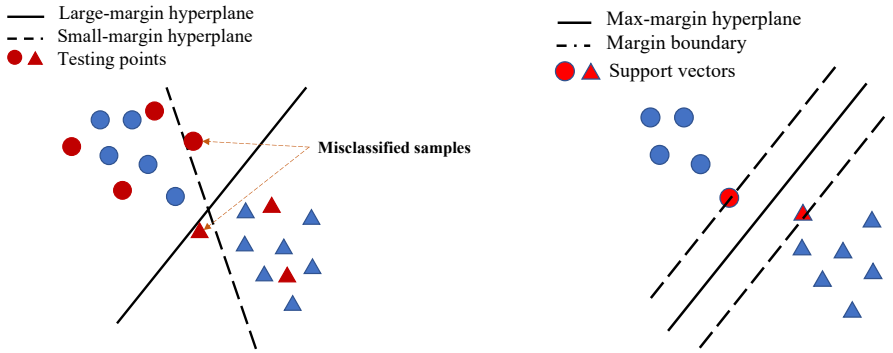

Figure 3: **Left:** Illustration of why the large-margin solution is preferred: a small-margin classifier often misclassifies test samples lying near the decision boundary. **Right:** Illustration of the **hard-margin SVM** solution——the max-margin separating hyperplane together with its margin boundaries and support vectors.

Suppose that the data are *linearly separable*; see Figure 3(Left). In this setting, many hyperplanes can perfectly separate the two classes. Among these candidates, it is intuitively desirable to choose the one that leaves the largest gap—or *margin*—between the two classes. A small-margin classifier may still fit the training data, but it is sensitive to perturbations and likely to misclassify test samples lying near the decision

boundary. In contrast, a large-margin classifier pushes the boundary away from all training points, reducing the chance that unseen samples fall into the ambiguous region and thereby achieving better generalization.

3.1 Hard-Margin SVM

The geometric intuition behind the support vector machine (SVM) begins with a linear **decision function**:

$$f_\theta(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b, \qquad \theta = (\mathbf{w}, b).$$

A sample $\mathbf{x}$ is assigned to class $+1$ if $f_\theta(\mathbf{x}) \geq 0$ and to class -1 otherwise. The decision boundary between the two classes is the hyperplane

$$H_\theta = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{w}^\top \mathbf{x} + b = 0\}.$$

Assumption 3.1 (Linear separability)**.** *There exists* $(\mathbf{w}, b) \in \mathbb{R}^{d+1}$ *such that* $y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 0$ *for all* $i \in [n]$.

Among all separating hyperplanes, we prefer the one that is most *robust* to perturbations of the data. Intuitively, a hyperplane with a larger "safety margin" can tolerate larger variation without changing its classification. The distance from $\mathbf{x}$ to the decision boundary,

$$d(\mathbf{x}; H_\theta) := \inf_{\mathbf{x}' \in H_\theta} \|\mathbf{x} - \mathbf{x}'\|_2 = \frac{|\mathbf{w}^\top \mathbf{x} + b|}{\|\mathbf{w}\|_2},$$

quantifies the *confidence* of prediction and is called the **geometric margin**. Hard-margin SVM aims to maximize the smallest margin over all samples:

$$\max_{\theta} \min_{i \in [n]} d(\mathbf{x}_i; H_\theta) \quad \text{s.t.} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 0. \qquad \text{(Hard-Margin SVM)} \tag{7}$$

Question 3.2. Why do we maximize the *minimum margin* rather than the *average margin* $\frac{1}{n}\sum_{i=1}^n d(\mathbf{x}_i; H_\theta)$?

The max–min problem (7) can be transformed into a convex optimization problem.

Theorem 3.3. *Under Assumption 3.1, problem* (7) *is equivalent to*

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{R}^d,\, b \in \mathbb{R}} \quad & \|\mathbf{w}\|_2^2, \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad \forall i \in [n]. \end{aligned} \tag{8}$$

Proof. For any feasible θ, we have $d(\mathbf{x}_i; H_\theta) = \frac{y_i(\mathbf{w}^\top \mathbf{x}_i + b)}{\|\mathbf{w}\|_2}$. Let $t = \min_i d(\mathbf{x}_i; H_\theta)$. Then problem (7) can be rewritten as

$$\max_{(\mathbf{w}, b) \in \mathbb{R}^{d+1},\, t > 0} t \quad \text{s.t.} \quad \frac{y_i(\mathbf{w}^\top \mathbf{x}_i + b)}{\|\mathbf{w}\|_2} \geq t, \quad \forall i.$$

Since scaling $(\mathbf{w}, b)$ by a positive constant does not change the separating hyperplane, we can rescale the parameters so that the minimum margin equals 1, i.e., $\min_i d(\mathbf{x}_i; H_\theta) = 1$. Equivalently, define

$$\tilde{\mathbf{w}} := \frac{\mathbf{w}}{t\|\mathbf{w}\|_2}, \qquad \tilde{b} := \frac{b}{t\|\mathbf{w}\|_2}.$$

Substituting into the constraint yields

$$\min_{\tilde{\mathbf{w}},\tilde{b}} \|\tilde{\mathbf{w}}\|_2 \quad \text{s.t.} \quad y_i(\tilde{\mathbf{w}}^\top \mathbf{x}_i + \tilde{b}) \geq 1, \quad \forall i,$$

whose squared form is exactly (8).

□

Problem (8) is convex and can be solved efficiently. It finds the separating hyperplane with the smallest ℓ_2-norm—equivalently, the largest margin.

Margin boundaries and support vectors. Let $\hat{\theta} = (\widehat{\mathbf{w}}, \hat{b})$ be the solution to (8) and then, $H_{\hat{\theta}}$ denote the optimal separating hyperplane. Define two parallel hyperplanes on each side:

$$H_+ = \{\mathbf{x} \in \mathbb{R}^d : \widehat{\mathbf{w}}^\top \mathbf{x} + \hat{b} = +1\}, \qquad H_- = \{\mathbf{x} \in \mathbb{R}^d : \widehat{\mathbf{w}}^\top \mathbf{x} + \hat{b} = -1\}.$$

These are called the **margin boundaries**. They pass through the nearest points of each class to $H_{\hat{\theta}}$ and are parallel to it. The perpendicular distance between $H_{\hat{\theta}}$ and either margin boundary is

$$d(H_\pm, H_{\hat{\theta}}) = \frac{1}{\|\widehat{\mathbf{w}}\|_2}.$$

The training samples can thus be categorized as follows:

- **Support vectors:** Points lying exactly on the margin boundaries satisfy $y_i(\widehat{\mathbf{w}}^\top \mathbf{x}_i + \hat{b}) = 1$. They are called *support vectors* because they "support" the optimal hyperplane—removing any of them would change the classifier.
- **Non-support vectors:** Points strictly outside the margin satisfy $y_i(\widehat{\mathbf{w}}^\top \mathbf{x}_i + \hat{b}) > 1$. Removing these points does not affect the optimal parameters $\hat{\theta}$.

Generalization insight. Let n_s denote the number of support vectors. Since the optimal separating hyperplane depends solely on these points, n_s can be regarded as a measure of the classifier's *effective complexity*. Intuitively, when most training samples lie far from the decision boundary (so that $n_s \ll n$), the classifier is determined by only a small subset of the data, and hence behaves like a low-complexity model with strong generalization ability:

$$\text{gen-err} \lesssim \frac{\text{effective complexity}}{n} \approx \frac{n_s}{n}.$$

In short, the principle of SVM can be summarized as:

$$\textbf{fewer support vectors} \quad \Rightarrow \quad \textbf{better generalization.}$$

3.2 Extensions to Nonseparable Data

A natural next step is to consider situations where the data are not linearly separable; see Figure 4. Essentially, there are two typical cases to address, each requiring a different approach:

- **Nonlinear decision boundary:** As illustrated in Figure 4(Left), a simple nonlinear decision boundary can perfectly separate the two classes.
- **Soft-margin classification:** As shown in Figure 4(Right), it may be preferable to allow a few points to be misclassified in exchange for a simpler decision boundary.

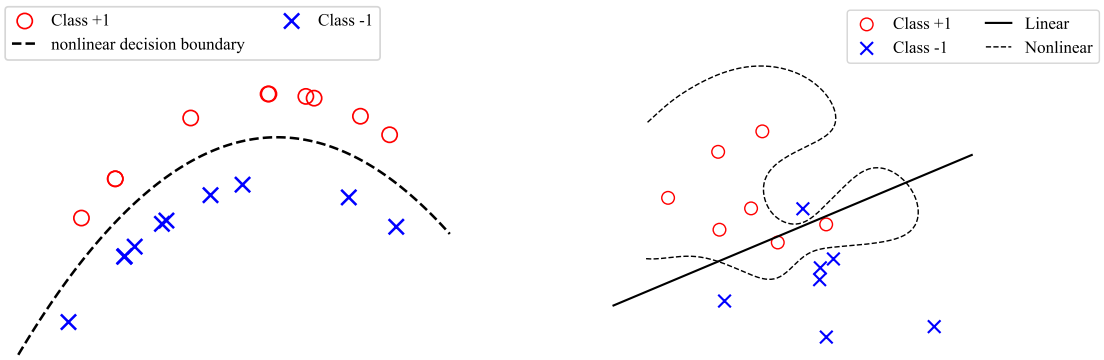

Figure 4: **Left:** A case where two classes can be separated by a simple nonlinear decision boundary. In this situation, the nonlinear hard-margin SVM is preferred. **Right:** An illustration where a soft-margin SVM is more appropriate, allowing a few points to be misclassified in exchange for a simpler decision boundary.

3.2.1 Nonlinear Hard-Margin SVM

To extend the max-margin principle beyond linear classifiers, we follow the insight behind the formulation (8). Let $\mathcal{F}$ be a (nonlinear) function class that is closed under positive scaling, i.e., $\alpha f \in \mathcal{F}$ for all $f \in \mathcal{F}$ and $\alpha > 0$. We say that the data are *$\mathcal{F}$-separable* if there exists $f \in \mathcal{F}$ such that

$$\min_{i \in [n]} y_i f(\mathbf{x}_i) \geq 1.$$

Definition 3.4 (Margin)**.** For a decision function $f \in \mathcal{F}$, the (sign) *margin* at a sample $\mathbf{x}$ is defined by

$$\gamma_f(\mathbf{x}) := y\, f(\mathbf{x}).$$

The margin $\gamma_f(\mathbf{x})$ quantifies the classifier's confidence in its prediction: a larger value means not only correct classification but also a stronger level of certainty. In the linear case, $\gamma_f(\mathbf{x})$ corresponds (up to normalization) to the signed distance between $\mathbf{x}$ and the decision boundary $S_f = \{\mathbf{x} \in \mathbb{R}^d : f(\mathbf{x}) = 0\}$. For general function classes $\mathcal{F}$, the margin may not admit a simple geometric interpretation, yet it remains a fundamental measure of classification confidence and the basis for defining nonlinear SVMs.

Following 8, the general *max-margin classifier* is then defined as the solution of

$$\begin{aligned} &\min_{f \in \mathcal{F}} \Omega(f), \\ &\quad \text{s.t. } \; y_i f(\mathbf{x}_i) \geq 1, \quad i \in [n], \end{aligned} \tag{9}$$

where $\Omega(\cdot)$ is a regularization functional that controls the complexity of the decision function. When $\mathcal{F}$ is the class of linear functions and $\Omega(f)$ is the ℓ_2 norm of the weight vector, this formulation recovers exactly to the hard-margin SVM (8).

Kernelized SVM. Given a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, the corresponding *kernelized SVM* considers decision functions and complexity measure given by

$$\mathcal{F} = \Big\{ f_{\boldsymbol{\alpha}}(\cdot) = \sum_{i=1}^{n} \alpha_i\, k(x_i, \cdot) \;:\; \boldsymbol{\alpha} \in \mathbb{R}^n \Big\}, \qquad \Omega(f_{\boldsymbol{\alpha}}) = \sum_{i,j=1}^{n} \alpha_i \alpha_j\, k(x_i, x_j).$$

A classical route to obtain this nonlinear extension is via the *kernel trick*, which arises naturally from the Lagrange dual formulation and the KKT conditions. Although this derivation differs in appearance from the one in this section, we follow (9) here because it generalizes naturally to a wider class of nonlinear function spaces, beyond those associated with a fixed kernel. For a detailed exposition of kernelized SVMs and Lagrange duality, see [Shalev-Shwartz and Ben-David, 2014, Section 15].

3.2.2 Soft-Margin SVM

When the data are not perfectly separable, enforcing all margin constraints may be impossible. The *soft-margin SVM* relaxes these constraints by allowing limited margin violations. To this end, we introduce nonnegative *slack variables* $\xi_i \geq 0$ that quantify the amount by which the i-th sample fails to satisfy the margin condition. The optimization problem becomes

$$\begin{aligned} \min_{f \in \mathcal{F},\, \boldsymbol{\xi} \in \mathbb{R}^n} \lambda\, \Omega(f) + \frac{1}{n} \sum_{i=1}^{n} \xi_i, \\ \text{s.t.} \quad y_i f(\mathbf{x}_i) \geq 1 - \xi_i, \quad i \in [n], \\ \xi_i \geq 0, \qquad i \in [n], \end{aligned} \tag{10}$$

where $\sum_{i=1}^{n} \xi_i$ measures the total margin violation.

Lemma 3.5. *Let $\ell(z) = \max(0, 1 - z)$ denote the hinge loss. Then the constrained problem (10) is equivalent to the unconstrained optimization problem*

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell\big(y_i f(\mathbf{x}_i)\big) + \lambda\, \Omega(f).$$

Proof. For any fixed classifier $f \in \mathcal{F}$, the optimization problem (10) is linear and decoupled in the variables $\xi_1, \ldots, \xi_n$ Thus, each ξ_i can be minimized independently given f. Specifically, for each $i \in [n]$, we have

$$\min_{\xi_i \geq 0} \quad \xi_i \quad \text{s.t.} \quad y_i f(\mathbf{x}_i) \geq 1 - \xi_i.$$

The smallest feasible ξ_i satisfying the constraint is therefore $\xi_i(f) = \max\big(0,\, 1 - y_i f(\mathbf{x}_i)\big)$. Substituting this optimal value back into the objective of (10) yields

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \max\big(0,\, 1 - y_i f(\mathbf{x}_i)\big) + \lambda\, \Omega(f),$$

which is precisely the desired formulation involving the hinge loss $\ell(z) = \max(0, 1 - z)$. □

We have the following observations for the hinge loss:

- If the margin is less than 1, a linear penalization is imposed.
- If the margin is greater than 1, no penalization is imposed.

The second point is very important, which allows soft-SVM to recover the hard-SVM when data are linearly separated (need also to take $\lambda \to 0$). In addition, it also makes much sense intuitively since one should not impose any or too much penalization if the predictions are correct and confident.

3.3 Designing Surrogate Losses via the Margin Perspective

revise: The key insight behind the soft-margin SVM is that we should penalize a sample only when its *margin* (i.e., prediction confidence) is small. The hinge loss is one such choice, but many other *surrogate losses* can serve a similar purpose—providing smooth, convex approximations to the discontinuous 0–1 loss while emphasizing the margin-based principle.

- **Squared hinge loss:** $\ell(z) = \max(0, 1 - z)^2$. It imposes a quadratic penalty for small or negative margins, and no penalty when the margin is sufficiently positive.
- **Exponential loss:** $\ell(z) = e^{-z}$. It penalizes misclassified points exponentially, while assigning exponentially smaller penalties as the margin increases. This loss is widely used in boosting algorithms (e.g., AdaBoost).
- **Logistic loss:** $\ell(z) = \log(1 + e^{-z})$. This corresponds to the loss used in the softmax (logistic regression) classifier for binary classification. Its asymptotic behavior is

$$\ell_{\text{logistic}}(z) \approx \begin{cases} e^{-z}, & z \gg 1, \\ -z, & z \ll -1. \end{cases}$$

  It behaves like a blend of the hinge and exponential losses—it decays exponentially when the margin is large and positive, and increases approximately linearly when the margin is negative.

Figure 5 visually compares several common loss functions, highlighting how each serves as a convex surrogate for the discontinuous 0–1 loss, while differing in smoothness and tail behavior. For reference, we also include the square loss $\ell(z) = (1 - z)^2$, which penalizes deviations of the margin from 1. Unlike the other losses, it imposes a large penalty even for confident predictions (i.e., large positive margins), which may lead to undesirable solutions. Consequently, this loss is rarely used in classification.

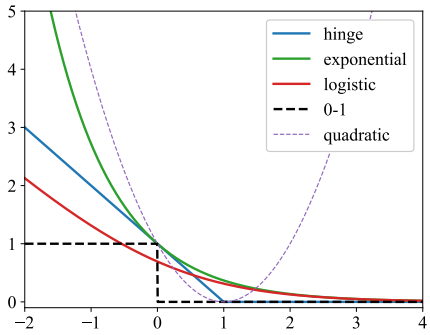

Figure 5: Comparison of common margin-based loss functions. All are convex surrogates of the 0–1 loss but differ in smoothness and penalization strength for small margins.

Question 3.6. Under what circumstances does the exponential loss outperform the hinge loss?

4 Bridging the Two Views: From Softmax to Max-Margin

We now show that, despite their different formulations, the probabilistic and geometric perspectives are closely connected. We first need some properties of the softmax operator.

Lemma 4.1. *Let $k = \operatorname{argmax}_j z_j$. Then, as $T \to 0$, we have*

$$\left(\frac{e^{z_1/T}}{\sum_j e^{z_j/T}}, \frac{e^{z_2/T}}{\sum_j e^{z_j/T}}, \ldots, \frac{e^{z_C/T}}{\sum_j e^{z_j/T}} \right) \to (0, \ldots, 0, 1, 0 \ldots, 0) = \mathbf{e}_k.$$

In other words, as the temperature $T \to 0$, the softmax becomes the max operator.

Proof. Omited! □

Remark 4.2. Note that the max operator is not continuous and thus replaced with a smoothed max operator is very useful in many applications. The softmax is the most popular one but there exist other choices; see https://en.wikipedia.org/wiki/Smooth_maximum.

The following lemma provides a non-asymptotic bound.

Lemma 4.3 (The LogSumExp trick)**.** *Let $z_1, \ldots, z_n \in \mathbb{R}$. For any $T > 0$, we have*

$$\max_{j \in [n]} z_j \le T \log \sum_{j=1}^{n} e^{z_j/T} \le \max_{j \in [n]} z_j + T \log(n).$$

The proof is left to homework.

Theorem 4.4. *Let $\ell(z) = e^{-z/T}$ with temperature $T > 0$, and define the (normalized) margin $\Gamma(f) := \min_{i \in [n]} y_i f(\mathbf{x}_i)$. Let $\Omega(\cdot)$ be a complexity measure used to constrain the hypothesis space, and define*

$$\hat{f}_T = \operatorname*{argmin}_{\Omega(f) \le 1} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i f(\mathbf{x}_i)), \qquad \hat{f}_{\mathrm{SVM}} = \operatorname*{argmax}_{\Omega(f) \le 1} \Gamma(f).$$

Then the margins of the two classifiers satisfy

$$\Gamma(\hat{f}_{\mathrm{SVM}}) - T \log n \;\le\; \Gamma(\hat{f}_T) \;\le\; \Gamma(\hat{f}_{\mathrm{SVM}}).$$

This result shows that the softmax classifier obtained from minimizing the exponential loss achieves a worst-case margin that is within $T \log n$ of the SVM margin. In particular, as $T \to 0$, the softmax classifier approaches the max-margin solution.

Proof. Let $Q(f) = -T \log \left(\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)/T} \right)$ be the softmax margin. Then,

$$\hat{f}_T = \operatorname*{argmin}_f T \log \left(\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)/T} \right) = \operatorname*{argmax}_f Q(f).$$

By Lemma 4.3, we have $\max_i(-y_i f(\mathbf{x}_i)) \le -Q(f) \le \max_i(-y_i f(\mathbf{x}_i)) + T \log n$, leading to

$$\Gamma(f) - T \log n \le Q(f) \le \Gamma(f).$$

By the deinition of f_{SVM} and $\hat{f}_T$, we have $\Gamma(\hat{f}_T) \le \Gamma(f_{\mathrm{SVM}})$ and

$$\Gamma(f_{\mathrm{SVM}}) - T \log n \le Q(f_{\mathrm{SVM}}) \le Q(\hat{f}_T) \le \Gamma(\hat{f}_T).$$

Thus, we complete the proof. □

In the above theorem, we only consider the exponential loss for simplicity. The same observation holds for any surrogate loss with an exponential tail if taking T to be sufficiently small. The cross-entropy loss (i.e., the logisitic loss for binary classification) is a particular example.

5 Summary and Final Remarks

In summary, the probabilistic and geometric perspectives offer two complementary ways of understanding classification. The probabilistic view, exemplified by the softmax classifier, models confidence through output probabilities and optimizes likelihood-based objectives. The geometric view, represented by the support vector machine, measures confidence via the margin and seeks the separating hyperplane that maximizes it. Although these approaches arise from different motivations—one statistical, the other geometric—they ultimately share the same goal: to achieve confident and generalizable classification. Bridging the two perspectives reveals that the softmax classifier can be regarded as a smooth, temperature-controlled approximation of the max-margin solution, thus unifying probabilistic modeling and geometric reasoning under a common principle.

Let $F(\cdot;\theta) : \mathcal{X} \to \Delta^{C-1}$ denote the classifier's output distribution. Consider training it with the simple square loss:

$$\min_{\theta} \sum_{i=1}^{n} \left\| F(\mathbf{x}_i;\theta) - \mathbf{e}_{y_i} \right\|_2^2.$$

Does the square loss really perform worse than the well-designed losses discussed above? The answer, in fact, depends on the data distribution.

- **Large-margin regime.** When most data points lie far from the decision boundary, as shown on the left of Figure 3, the square loss is indeed a poor choice. It penalizes large margins, even though such margins are desirable for generalization and should not be suppressed.

- **Boundary-dense regime.** In contrast, when data are distributed close to the decision boundary, as illustrated in Figure 6, penalizing all margins to make them more uniform is not necessarily a bad idea. In fact, in many modern machine learning applications, data are often nonlinearly separable yet concentrated near the boundary. In such cases, classifiers trained with the square loss perform comparably to those trained with softmax or max-margin losses. For further discussion, see [Hui and Belkin, , Muthukumar et al., 2021].

The term *classification* in this lecture should be understood primarily as a *modeling perspective* rather than merely a specific type of problem. While classification and regression are often presented as distinct tasks—predicting discrete versus continuous targets—their boundary is in fact fluid. Many problems that appear purely regression-like can be modeled more effectively from a classification viewpoint, especially when the goal is to capture decision boundaries or coarse-level distinctions rather than precise numeric values.

For instance, consider predicting a person's height. Although height is a continuous variable ranging, for example, from 140 cm to 190 cm, in many applications we care only about coarse distinctions—say, within 1 cm accuracy. In that case, the problem can equivalently be viewed as an 50-class classification task, where each class corresponds to an integer height value. This perspective highlights that the choice between regression and classification is often a matter of modeling resolution, not of fundamental problem type.

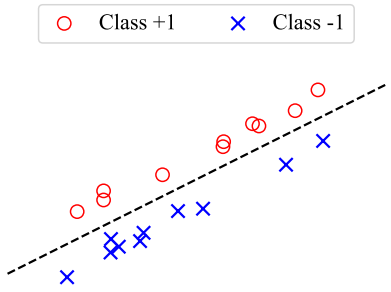

Figure 6: A classification problem in which most data points lie close to the decision boundary. In this case, maximizing the minimum margin provides little additional benefit.

References

[Hui and Belkin,] Hui, L. and Belkin, M. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. In *International Conference on Learning Representations*.

[Müller et al., 2019] Müller, R., Kornblith, S., and Hinton, G. E. (2019). When does label smoothing help? *Advances in neural information processing systems*, 32.

[Muthukumar et al., 2021] Muthukumar, V., Narang, A., Subramanian, V., Belkin, M., Hsu, D., and Sahai, A. (2021). Classification vs regression in overparameterized regimes: Does the loss function matter? *Journal of Machine Learning Research*, 22(222):1–69.

[Shalev-Shwartz and Ben-David, 2014] Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.