Machine Learning (Supplement)

Kon Yi

September 17, 2025

Abstract

This is the note for my personal understanding to the lecture slides of machine learning of 李濬屹. The chapters and sections are following the slides. If I don't have some extra understanding of the sections, then I will remain it empty.

Contents

Chapter 1

Classical Machine Learning Fundamentals

Lecture 1

1.1 What is machine learning?

5 Sep. 16:30

1.2 ML problem class

1.3 Assumptions

1.4 Evaluation / Verification

1.5 ML types and procedures

1.6 Basic algorithms

Chapter 2

Loss Function, Optimization, and Advanced Classical Methods

Lecture 2

12 Sep. 16:30

2.1 Loss function

2.1.1 About the x-axis of the graph

Note that in the graph of loss function, the x-axis is the *margin*, which is $\hat{y}y$, where y is the real label and $\hat{y}$ is the label predicted by the model. Margin is used to estimate our cofidence in the prediction. Because our model generates a score to represent its confidence in the prediction, and in the classic binary classification problem, the label is $\{-1, 1\}$, so the sign of $\hat{y}y$ means the predicted label, and $|\hat{y}|$ is the confidence of the prediction. If it predicts something wrong with high confidence, then the loss function will return high value.

2.1.2 Meaning for cross entropy

The original definition of entropy of a probability distribution p is

$$H(p) = \mathbb{E}_{x\sim p}[-\log(x)] = -\int p(x)\log p(x)\,\mathrm{dx}.$$

The entropy represents how surprised you are to the whole distribution, if some x with low incidence rate happens, then you will be more surprsied than the one with high incidence rate happen, and note that the grpah of $y = -\log x$ is as follow:

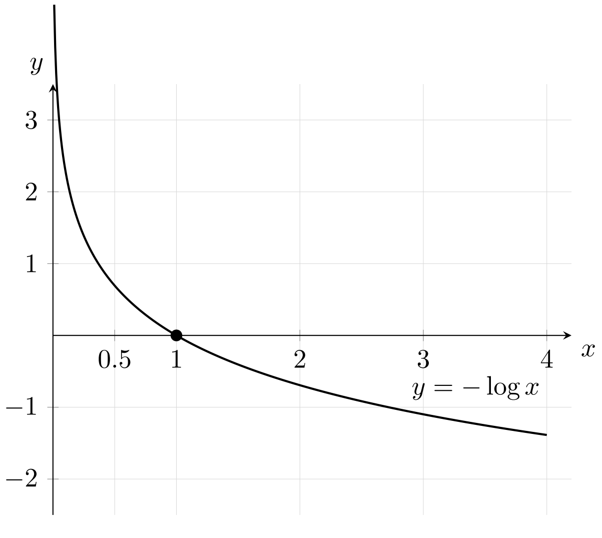

Figure 2.1: $y = -\log x$

Hence, if $p(x)$ is low, then $-\log p(x)$ is high, and thus the entropy is well-defined. Hence, we can use entropy to represent the amount of information or the average surprised extent in a probability distribution.

Now we talk about the cross entropy. Cross entropy can be used to estimate the offset of a predicted distribution q contrasted to the real distribution p. We will talk about the reason in the following text. The definition of cross entropy of p(real) and q(predicted) is

$$H(p,q) = \mathbb{E}_{x\in p}[-\log q(x)].$$

Also, notice that

$$\mathbb{E}_{x\sim p}[-\log q(x)] + \mathbb{E}_{x\sim p}\left[-\log \frac{p(x)}{q(x)}\right] = \mathbb{E}_{x\sim p}[-\log p(x)],$$

and that

$$\mathbb{E}_{x\sim p}\left[-\log \frac{p(x)}{q(x)}\right] = \mathbb{E}_{x\sim p}\left[\log \frac{q(x)}{p(x)}\right] \leq \log \mathbb{E}_{x\sim p}\left[\frac{q(x)}{p(x)}\right] = \log 1 = 0$$

by Jenson's inequality, and thus we know

$$H(p,q) \geq H(p)$$

and the equality holds only when p is identical to q. Hence, we can use cross entropy to represent if our prediction model is good.

2.1.3 Logistic loss function

If $\{y_i\}_{i=1}^n$ is from a distribution, and our model predicts discrete or continuous labels, then we can use sigmoid function to convert the generated labels to a probability distribution (note that the range of sigmoid function is $[0,1]$). Hence, by linking the sigmoid function to the cross entropy, we can estimate if our model is good.

2.1.4 Quantile loss

Quantile loss can help us predict the labels at specific quantile since the optimal point of expected risk function based on quantile loss is at q^* with $P(Y < q^*) = \tau$ if we define $\alpha = \tau$. Hence, if we use this loss funciton to train models, we will set q to be $\hat{y}$, and thus we can let the model to predict the label at τ-th quantile. In the real training session, we will use some samples to reflect the whole sample space, and then write a approximated $R(q)$ formula and then use some optimization method like gradient descent to find the optimal "pseudo" q. Now since we know $\arg\min R(q) = q^*$, so we know the τ-th quantile's label.

Let Y be a continuous random variable with cumulative distribution function (CDF) F_Y. For a fixed $0 < \tau < 1$, the quantile loss (pinball loss) is defined as

$$L_\tau(y,q) = \begin{cases} \tau(y-q), & y \geq q, \\ (1-\tau)(q-y), & y < q. \end{cases}$$

Define the expected risk:

$$R(q) = \mathbb{E}[L_\tau(Y,q)] = \int_{-\infty}^{\infty} L_\tau(y,q)\, dF_Y(y).$$

We want to compute the derivative of $R(q)$ with respect to q.

Step 1: Split the integral

$$R(q) = \underbrace{\int_{-\infty}^{q} (1-\tau)(q-y)\, dF_Y(y)}_{I_1(q)} + \underbrace{\int_{q}^{\infty} \tau(y-q)\, dF_Y(y)}_{I_2(q)}.$$

Step 2: Apply Leibniz's rule

For an integral of the form $\int_{a(q)}^{b(q)} f(y,q)\,dF_Y(y)$, if $f(y,q)$ is differentiable in q and integrable, the derivative is

$$\frac{d}{dq}\int_{a(q)}^{b(q)} f(y,q)\,dF_Y(y) = f(b(q),q)b'(q) - f(a(q),q)a'(q) + \int_{a(q)}^{b(q)} \frac{\partial f}{\partial q}(y,q)\,dF_Y(y).$$

Integral $I_1(q)$

$$I_1(q) = \int_{-\infty}^{q} (1-\tau)(q-y)\,dF_Y(y)$$

Here $a(q) = -\infty$, $b(q) = q$, $f(y,q) = (1-\tau)(q-y)$, $\frac{\partial f}{\partial q} = 1-\tau$. By Leibniz's rule:

$$\frac{dI_1}{dq} = f(q,q)\cdot 1 - f(-\infty,q)\cdot 0 + \int_{-\infty}^{q} (1-\tau)\,dF_Y(y) = (1-\tau)F_Y(q).$$

Integral $I_2(q)$

$$I_2(q) = \int_{q}^{\infty} \tau(y-q)\,dF_Y(y)$$

Here $a(q) = q$, $b(q) = \infty$, $f(y,q) = \tau(y-q)$, $\frac{\partial f}{\partial q} = -\tau$. By Leibniz's rule:

$$\frac{dI_2}{dq} = f(\infty,q)\cdot 0 - f(q,q)\cdot 1 + \int_{q}^{\infty} (-\tau)\,dF_Y(y) = -\tau(1-F_Y(q)).$$

Step 3: Combine derivatives

$$\frac{dR}{dq} = \frac{dI_1}{dq} + \frac{dI_2}{dq} = (1-\tau)F_Y(q) - \tau(1-F_Y(q)) = F_Y(q) - \tau.$$

Step 4: Find the minimizer

Setting $\frac{dR}{dq} = 0$ gives

$$F_Y(q^*) = \tau,$$

which is exactly the definition of the τ-quantile of Y.

□

2.2 Optimization

2.2.1 Why Use the Hessian in Newton's Method?

Newton's method is a second-order iterative optimization method. Its update rule is:

$$\theta_{t+1} = \theta_t - H_L(\theta_t)^{-1}\nabla L(\theta_t),$$

where $\nabla L(\theta_t)$ is the gradient (first derivative) and $H_L(\theta_t)$ is the Hessian (second derivative) of the loss function $L(\theta)$ at θ_t.

1. Intuition

- The gradient ∇L tells us **which direction** to move.
- The Hessian H_L tells us **how big a step to take** in that direction, based on the local curvature.

In other words, the Hessian allows Newton's method to automatically adjust the step size along each direction, unlike gradient descent which requires manually tuned learning rates.

2. Advantages of Using the Hessian

1. **Quadratic convergence:** Near the optimum, Newton's method converges much faster than gradient descent (which converges linearly). Using the Hessian significantly reduces the number of iterations needed.

2. **Adaptive steps along different directions:** For functions with steep curvature in some directions and flat curvature in others, gradient descent requires careful tuning of the learning rate. Newton's method scales the update according to the curvature automatically:

$$\text{Update step along eigenvector } v_i \sim \frac{1}{\lambda_i} \text{ (eigenvalue of Hessian).}$$

3. Why Not Always Use the Hessian?

- **Computational cost:** Hessian is a $d \times d$ matrix; storing and inverting it is expensive in high dimensions.
- **Ill-conditioning:** If the Hessian is nearly singular, inversion is unstable.
- **Memory usage:** In deep learning, d can be tens of thousands or more.

4. Practical Workarounds

- **Quasi-Newton methods (BFGS, L-BFGS):** approximate Hessian from past gradients.
- **Gauss-Newton / Levenberg-Marquardt:** specialized for squared-error problems, avoids full Hessian.
- **Hessian-free methods:** use Hessian-vector products to approximate the effect of the Hessian without storing the full matrix.

Summary

Hessian is valuable because it provides curvature information, which leads to faster and more stable convergence.

Even though computing the Hessian can be slow, approximate methods can capture its advantages with lower cost.

2.3 Recap: Basic classification

2.4 Advanced classification

2.4.1 Primal and Dual Problems in Optimization and SVM

1. General Optimization Problem

In optimization, we often want to find the "best" value of a function subject to some constraints. A general constrained optimization problem is called the **primal problem**:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \le 0, \quad i = 1, \dots, m, \\ & h_j(x) = 0, \quad j = 1, \dots, p, \end{aligned}$$

where

- $f(x)$ is the **objective function** we want to minimize,
- $g_i(x) \le 0$ are **inequality constraints**,
- $h_j(x) = 0$ are **equality constraints**.

2. Lagrangian and Dual Function

To solve the primal problem, we introduce **Lagrange multipliers** $\alpha_i \geq 0$ for the inequalities and β_j for the equalities. The **Lagrangian** is defined as:

$$\mathcal{L}(x, \alpha, \beta) = f(x) + \sum_{i=1}^{m} \alpha_i g_i(x) + \sum_{j=1}^{p} \beta_j h_j(x)$$

The **dual function** is obtained by minimizing the Lagrangian over x:

$$g(\alpha, \beta) = \inf_x \mathcal{L}(x, \alpha, \beta)$$

Then, the **dual problem** is:

$$\begin{aligned} \text{maximize} \quad & g(\alpha, \beta) \\ \text{subject to} \quad & \alpha_i \geq 0, \quad i = 1, \ldots, m \end{aligned}$$

Key points:

- The dual problem provides a lower bound to the primal problem (for minimization).
- Sometimes solving the dual is easier than the primal.
- Under certain conditions (called *strong duality*), the optimal value of the dual equals that of the primal.

3. Application to SVM

For a hard-margin SVM, the primal problem is:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ \text{subject to} \quad & y_i(w \cdot x_i + b) \geq 1, \quad i = 1, \ldots, n \end{aligned}$$

Here:

- w is the weight vector, b is the bias,
- $y_i \in \{-1, +1\}$ are the labels,
- x_i are the feature vectors,
- The constraints ensure that each point is correctly classified with a margin of at least 1.

The Lagrangian for this problem is:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i \big[y_i(w \cdot x_i + b) - 1\big], \quad \alpha_i \geq 0$$

Minimizing over w and b leads to the **dual problem**:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\ \text{subject to} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \alpha_i \geq 0 \end{aligned}$$

Benefits of the dual form:

- Depends only on **dot products** of data points, allowing the use of kernels.
- Often easier to solve when the number of features is large.
- The optimal w can be recovered from α_i:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

4. Summary

- The **primal problem** is the original optimization problem we want to solve.
- The **dual problem** is derived via Lagrange multipliers and often easier to solve.
- In SVMs, the dual problem is particularly useful for kernel methods and large feature spaces.

2.4.2 Why the Dual Solution Can Match the Primal Optimal Value and Recovering w in SVM

1. Weak and Strong Duality

Consider a general constrained optimization (primal) problem:

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \le 0, \quad i = 1, \dots, m, \\ & h_j(x) = 0, \quad j = 1, \dots, p. \end{aligned}$$

We define the **Lagrangian**:

$$\mathcal{L}(x, \alpha, \beta) = f(x) + \sum_{i=1}^{m} \alpha_i g_i(x) + \sum_{j=1}^{p} \beta_j h_j(x), \quad \alpha_i \ge 0$$

and the **dual function**:

$$g(\alpha, \beta) = \inf_{x} \mathcal{L}(x, \alpha, \beta)$$

Then the dual problem is:

$$\max_{\alpha, \beta} \quad g(\alpha, \beta) \quad \text{subject to } \alpha_i \ge 0$$

Weak duality: For any feasible primal x and dual (α, β),

$$g(\alpha, \beta) \le f(x)$$

This is always true, so the dual optimum is a *lower bound* of the primal optimum.

Strong duality: If the problem is convex (i.e., $f(x)$ convex, $g_i(x)$ convex, $h_j(x)$ linear) and satisfies some regularity conditions (e.g., Slater's condition), then

$$\max_{\alpha, \beta} g(\alpha, \beta) = \min_{x} f(x)$$

Hence, the *optimal values* of the primal and dual problems are equal. However, the *variables* are generally different: the primal variables x are in the original space, while the dual variables (α, β) correspond to the constraints.

2. Application to SVM

For a hard-margin SVM, the primal problem is:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ \text{subject to} \quad & y_i(w \cdot x_i + b) - 1 \ge 0, \quad i = 1, \dots, n \end{aligned}$$

The Lagrangian is:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i \big[y_i(w \cdot x_i + b) - 1\big], \quad \alpha_i \ge 0$$

To find the dual, we minimize $\mathcal{L}$ with respect to w and b:

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Substituting back, the dual problem becomes:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\ \text{subject to} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \alpha_i \geq 0 \end{aligned}$$

3. Why We Can Recover w from the Dual Solution

The key is the **stationarity condition** of the Lagrangian:

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

- This equation comes from setting the derivative of the Lagrangian with respect to the primal variable w to zero. - Therefore, once we solve the dual problem and obtain the optimal α_i^*, we can recover the optimal w^* as:

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$$

Only the **support vectors** have $\alpha_i^* > 0$, so the sum involves only those points that lie on the margin. The bias b can then be recovered using any support vector (x_i, y_i):

$$b = y_i - w^* \cdot x_i$$

4. Summary

- Duality allows us to transform the primal problem into another problem that may be easier to solve.
- Weak duality always holds; strong duality holds for convex problems, including SVM.
- In SVM, the dual variables α_i correspond to constraints, not w directly.
- w can be recovered from the dual solution via the stationarity condition of the Lagrangian.

2.4.3 When to use primal problem and dual problem?

In the p.28 of the lecture slide, it says that if small features and many samples, then use primal problem, while if many features and few samples, then use dual problem. This is because in the primal problem, we have to directly find the best w, which takes $O(d)$ time to optimize it, where d is the number of features. As for the dual problem, it takes $O(n)$ time, where n is the size of sample set. Note that the time complexity is highly relavant to the number of parameters need to be chose. In the primal problem, we have to select $d+1$ parameters $b, w_1, w_2, \ldots, w_d$, while in the dual problem we need to select n parameters $\alpha_1, \alpha_2, \ldots, \alpha_n$.

2.4.4 Connection between Weak Duality and the Min-Max Inequality

1. Definition of the Dual Function

For a primal problem:

$$\min_x f(x) \quad \text{s.t. } g_i(x) \leq 0, \, h_j(x) = 0$$

we define the Lagrangian:

$$\mathcal{L}(x, \alpha, \beta) = f(x) + \sum_i \alpha_i g_i(x) + \sum_j \beta_j h_j(x), \quad \alpha_i \geq 0$$

The dual function is:

$$g(\alpha, \beta) = \inf_x \mathcal{L}(x, \alpha, \beta)$$

2. Weak Duality

By definition of the infimum:

$$g(\alpha, \beta) = \inf_x \mathcal{L}(x, \alpha, \beta) \leq \mathcal{L}(x, \alpha, \beta), \quad \forall x$$

Now, for any feasible x (satisfying $g_i(x) \leq 0$ and $h_j(x) = 0$), the Lagrangian reduces to:

$$\mathcal{L}(x, \alpha, \beta) = f(x) + \sum_i \alpha_i g_i(x) + \sum_j \beta_j h_j(x) \leq f(x)$$

because $\alpha_i \geq 0$ and $g_i(x) \leq 0$, and $h_j(x) = 0$.
Thus, for any feasible x and any $\alpha, \beta \geq 0$:

$$g(\alpha, \beta) \leq f(x)$$

This is exactly the **weak duality inequality**.
—

3. Min-Max Inequality

Consider the general min-max problem:

$$\min_x \max_{\alpha,\beta \geq 0} \mathcal{L}(x, \alpha, \beta) \geq \max_{\alpha,\beta \geq 0} \min_x \mathcal{L}(x, \alpha, \beta)$$

- The left-hand side (LHS) is the *primal perspective*: for each x, pick the worst-case (largest) Lagrangian over α, β, then pick the best x to minimize it.
- The right-hand side (RHS) is the *dual perspective*: for each α, β, pick the x that minimizes $\mathcal{L}$, then pick α, β that maximize this minimum.

The inequality

$$\min_x \max_{\alpha,\beta} \mathcal{L}(x, \alpha, \beta) \geq \max_{\alpha,\beta} \min_x \mathcal{L}(x, \alpha, \beta)$$

holds *for any function* $\mathcal{L}(x, \alpha, \beta)$, not necessarily convex.
—

4. Connecting Weak Duality and Min-Max

Notice that

$$\max_{\alpha,\beta \geq 0} \min_x \mathcal{L}(x, \alpha, \beta) = \max_{\alpha,\beta \geq 0} g(\alpha, \beta)$$

and for feasible x:

$$\min_x \max_{\alpha,\beta \geq 0} \mathcal{L}(x, \alpha, \beta) = \min_x f(x) = f(x^*) \quad \text{(because feasible } x \text{ makes constraints satisfied)}$$

Here we briefly explain why $\min_x \max_{\alpha,\beta \geq 0} \mathcal{L}(x, \alpha, \beta) = \min_x f(x)$. Consider the Lagrangian for a constrained optimization problem:

$$\mathcal{L}(x, \alpha, \beta) = f(x) + \sum_{i=1}^m \alpha_i g_i(x) + \sum_{j=1}^p \beta_j h_j(x), \quad \alpha_i \geq 0$$

where $g_i(x) \leq 0$ are inequality constraints and $h_j(x) = 0$ are equality constraints.

1. Maximizing Lagrangian over α, β

For a **feasible point** x (satisfying $g_i(x) \leq 0$ and $h_j(x) = 0$):

$$\sum_{j=1}^{p} \beta_j h_j(x) = 0 \quad \text{for any } \beta_j$$

$$\sum_{i=1}^{m} \alpha_i g_i(x) \leq 0 \quad \text{because } \alpha_i \geq 0 \text{ and } g_i(x) \leq 0$$

- If $g_i(x) < 0$, choosing $\alpha_i > 0$ would decrease $\mathcal{L}(x, \alpha, \beta)$ - Therefore, the maximum over $\alpha_i \geq 0$ is achieved at $\alpha_i = 0$

Hence, for feasible x:

$$\max_{\alpha \geq 0, \beta} \mathcal{L}(x, \alpha, \beta) = f(x)$$

—

2. Minimizing over x

$$\min_x \max_{\alpha \geq 0, \beta} \mathcal{L}(x, \alpha, \beta)$$

- For infeasible x ($g_i(x) > 0$ or $h_j(x) \neq 0$), we can make $\sum_i \alpha_i g_i(x) + \sum_j \beta_j h_j(x)$ arbitrarily large by choosing large α_i or β_j. - Hence, infeasible x will never minimize the max Lagrangian. - Only feasible x contribute, giving $\max_{\alpha,\beta} \mathcal{L}(x, \alpha, \beta) = f(x)$

Therefore:

$$\min_x \max_{\alpha \geq 0, \beta} \mathcal{L}(x, \alpha, \beta) = \min_{x \text{ feasible}} f(x)$$

—

3. Intuition

- The Lagrange multipliers α_i, β_j act as *penalties* for violating constraints. - For feasible x, the penalty terms vanish or are non-positive, so the max over α, β returns $f(x)$. - For infeasible x, the max over α, β blows up, so these points cannot be the minimizer.

This explains why the min-max formulation over the Lagrangian recovers exactly the primal optimal value.

Then the min-max inequality gives:

$$\min_{x \text{ feasible}} f(x) = \min_x \max_{\alpha,\beta} \mathcal{L}(x, \alpha, \beta) \geq \max_{\alpha,\beta \geq 0} g(\alpha, \beta)$$

This is exactly the weak duality statement:

$$\text{dual optimum} \leq \text{primal optimum}$$

—

5. Summary

- The inequality $g(\alpha, \beta) \leq f(x)$ for feasible x is a pointwise statement.
- The min-max inequality $\min_x \max_{\alpha,\beta} \mathcal{L} \geq \max_{\alpha,\beta} \min_x \mathcal{L}$ is the global version and directly implies weak duality.
- For convex problems (like SVM), strong duality holds, meaning the inequality becomes equality:

$$\min_x \max_{\alpha,\beta} \mathcal{L}(x, \alpha, \beta) = \max_{\alpha,\beta} \min_x \mathcal{L}(x, \alpha, \beta)$$

Lecture 3

19 Sep. 16:30

Appendix