

Chapter 1 Linear Regression

Introduction

- ERM
- Gradient Descent

- Ridge Regression (L_2 regularization)
- Lasso Regression (L_1 regularization)

1.1 Basic Knowledge

Example 1.1 Linear Regression

Settings.

- Dataset: $D = \{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. Here, y_i denotes the regression target, while x_i represents the input features used to predict y_i .
- Linear Model: $f(x) = W^\top x + b$, with weight $W \in \mathbb{R}^d$ and bias $b \in \mathbb{R}$.



Note This definition is equivalent to an inner product: $\hat{y} = W^\top x + b$.

Definition 1.1 (Learnable / Trainable Parameters)

Learnable parameters are those that can be updated during the training process.



Quiz. How to determine whether a parameter is learnable? Quiz: How to determine W and b ?

Ans: **ERM** (Empirical Risk Minimization)

- Loss function. Squared Loss (SE) is commonly used during optimization. The training objective can be written as:

$$\operatorname{argmax}_{W, b} \frac{1}{n} \sum_{i \in [n]} (y_i - (W^\top x_i + b))^2 \quad (1.1)$$

The blue factor $1/n$ can be omitted in theoretical analysis, but is often kept in practice to stabilize the loss function during implementation.

Quiz: How to optimize the parameters?

Ans: **Gradient Descent** (as a traditional ML method). In the case of linear regression:

$$\frac{\partial \mathcal{L}}{\partial b} = -2 \sum_{i \in [n]} (y_i - W^\top x_i - b) \quad (1.2)$$

$$\frac{\partial \mathcal{L}}{\partial W} = -2 \sum_{i \in [n]} (y_i - W^\top x_i - b)x_i \quad (1.3)$$



Note In the field of machine learning, the gradient of a scalar with respect to a vector is itself a vector (**not a covector**). This means:

$$\frac{\partial \mathcal{L}}{\partial W} = \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial W_1} \\ \frac{\partial \mathcal{L}}{\partial W_2} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial W_d} \end{pmatrix} = \left(\frac{\partial \mathcal{L}}{\partial W_1}, \frac{\partial \mathcal{L}}{\partial W_2}, \dots, \frac{\partial \mathcal{L}}{\partial W_d} \right)^\top \quad (1.4)$$



Note Here are some commonly used derivative formulas:

$$\frac{\partial x^\top x}{\partial x} = 2x \quad (1.5)$$

$$\frac{\partial a^\top x}{\partial x} = a, \quad \frac{\partial Ax}{\partial x} = A^\top \quad (1.6)$$

$$\frac{\partial x^\top Ax}{\partial x} = (A + A^\top)x \quad (1.7)$$

Remark Both sides of an equation must have the same dimension. This principle can be used as a consistency check.

We optimize the parameters by subtracting a scalar multiple of the gradient from the parameters, considering the physical meaning of the gradient: the direction of the steepest **increase**.

Definition 1.2 (Hyperparameter)

A parameter that is fixed during optimization and specified before the training process.

That is:

$$W' = W - \alpha \frac{\partial \mathcal{L}}{\partial W}, \quad b' = b - \alpha \frac{\partial \mathcal{L}}{\partial b} \quad (1.8)$$

Optimization will stop when the norm of the parameter update becomes smaller than a given hyperparameter.

1.2 Closed-Form of Linear Regression

Proposition 1.1

Linear Regression has **Closed-Form** solution.

Settings.

- Matrix $\tilde{X} := (X, \mathbb{1}) \in \mathbb{R}^{n \times (d+1)}$;
- $y = (y_1, \dots, y_n)^\top \in \mathbb{R}^n$;
- $\hat{w} = (w, b)^\top \in \mathbb{R}^{d+1}$.

Then the loss function of \hat{w} can be written as:

$$\mathcal{L}(\hat{w}) = (y - X\hat{w})^\top (y - X\hat{w}) = \|y - X\hat{w}\|_2^2 \quad (1.9)$$

Here, $\|\cdot\|_p$ denotes the p -norm of a vector.



Note Vectors can sometimes be treated as scalars, since linearity ensures that the validity of a proposition can be extended to any finite dimension.

Notice that the optimization stops when $\partial \mathcal{L}(\hat{w}) / \partial \hat{w} = 0$. Under this condition, the parameters can be solved from the above constraint by following steps:

$$\frac{\partial \mathcal{L}(\hat{w})}{\partial \hat{w}} = -2X^\top (y - X\hat{w}) \quad (1.10)$$



Note Both dimensional analysis and calculation using Leibniz's rule lead to the same result as the formula above:

$$\begin{aligned} \mathcal{L}(\hat{w}) &= y^\top y - 2y^\top X\hat{w} + \hat{w}^\top X^\top X\hat{w} \\ \partial_{\hat{w}} \mathcal{L}(\hat{w}) &= -2X^\top y + 2X^\top X\hat{w} \\ &= -2X^\top (y - X\hat{w}) \end{aligned}$$

Remark More matrix formulas are available in **Matrix Cookbook**.

Thus, the target of the optimization satisfied:

$$X^\top y = X^\top X\hat{w} \quad (1.11)$$

That is:

$$\hat{w} = (X^\top X)^{-1} X^\top y \quad (1.12)$$

when $X^\top X$ invertible (non-singular / full-rank).

Example 1.2 When does $X^\top X$ not invertible?

Solution $X \in \mathbb{R}^{n \times (d+1)}$:

- $d + 1 > n$. *Brief Proof:* $\text{rank}(X^\top X) = \text{rank}(X) \leq \min(n, d + 1) = n < d + 1$.
- X has repeated columns. *Proof is trivial.*

When $X^\top X$ isn't invertible:

1. If $\text{rank}(X^\top X, X^\top y) > \text{rank}(X^\top X)$, \hat{w} has no solution;

2. \hat{w} has infinity solution o.w.

Situation 1 is **Impossible** because both $X^\top X$ and $X^\top y$ can be represented in the column space of X^\top . Therefore, the optimization problem must have a solution, which may be either unique or infinite.

As an infinite set of solutions makes it difficult to determine which estimate of \hat{w} to choose, we apply L_2 **regularization** to linear regression, which is commonly referred to as **Ridge Regression**. That is:

$$\mathcal{L}_{L_2} := \mathcal{L}(\hat{w}) + \boxed{\lambda \|\hat{w}\|_2^2} \quad (1.13)$$

Noticed that $\|\hat{w}\|_2^2 = \sum_{i=1}^{d+1} \hat{w}_i^2$, L_2 regularization prevents any single dimension from being assigned an excessively large weight, and encourages the model to make use of more dimensions during training.

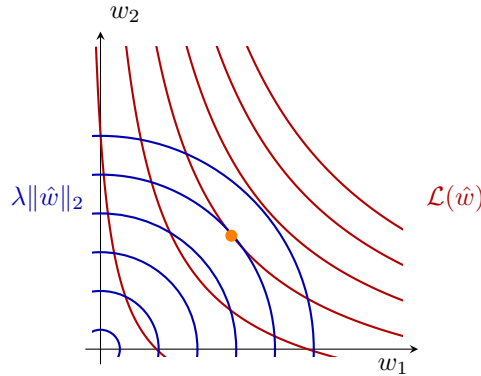


Figure 1.1: Illustration of L_2 regularization. The contours represent level sets of the regularized loss $\mathcal{L}(\hat{w}) + \lambda \|\hat{w}\|_2^2$, which take the form of concentric ellipses (circle in the plot).

During ridge regression, we minimize the \mathcal{L}_{L_2} :

$$\underset{\hat{w}}{\operatorname{argmin}} (y - X\hat{w})^\top (y - X\hat{w}) + \lambda \hat{w}^\top \hat{w} \quad (1.14)$$

The optimization stops when:

$$\frac{\partial \mathcal{L}_{L_2}}{\partial \hat{w}} = -2X^\top y + 2X^\top X \hat{w} + 2\lambda \hat{w} = 0 \quad (1.15)$$

$$\Rightarrow (X^\top X + \lambda I) \hat{w} = X^\top y \quad (1.16)$$

Proposition 1.2

$X^\top X + \lambda I$ always invertible.

Proof Since $X^\top X$ is a real symmetric matrix, we have the eigen-decomposition $X^\top X = U\Lambda U^\top$, where $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_{d+1})$. Moreover, as $X^\top X \succeq 0$ is positive semi-definite, it follows that $\forall i \in [d+1]$, $\lambda_i \geq 0$. Note that:

$$\lambda I = \lambda U U^\top \quad (1.17)$$

since U is an orthogonal matrix. Hence:

$$X^\top X + \lambda I = U(\Lambda + \lambda I)U^\top \quad (1.18)$$

For all $i \in [d+1]$, we have:

$$\lambda_i + \lambda > \lambda_i \geq 0 \quad (1.19)$$

Thus, $X^\top X + \lambda I$ is a full-rank matrix.

Remark Numerical issues may still occur even if $X^\top X$ is full rank (e.g., when eigenvalues λ_k are close to zero). The L_2 regularization factor λ mitigates this issue by shifting the eigenvalues upward, thereby improving numerical stability during training.

Another regularization method often used is L_1 regularization, where the loss function is defined as:

$$\mathcal{L}_{L_1} := \mathcal{L}(\hat{w}) + \boxed{\lambda \|\hat{w}\|_1} \quad (1.20)$$

L_1 regularization can induce sparsity in \hat{w} , which works in contrast to L_2 regularization. Specifically, L_1 regularization encourages the model to rely on only a small subset of input features, effectively performing **feature selection**.

Linear regression with L_1 regularization is called **Lasso Regression** (Least Absolute Shrinkage and Selection Operator).

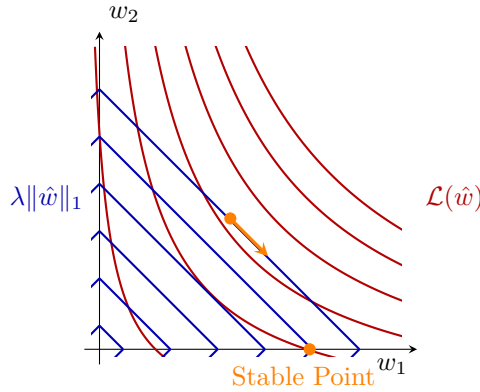


Figure 1.2: Illustration of L_1 regularization. The contours represent level sets of the regularized loss $\mathcal{L}(\hat{w}) + \lambda \|\hat{w}\|_1$, which take the form of nested diamonds (squares rotated by 45° in the plot).

1.3 Geomeric View of LR

Ideally, we would like to solve $X\hat{w} = y$. If y lies on the hypersurface

$$\mathcal{M}(X) := \text{Span}(X) = \{Xw : w \in \mathbb{R}^d\} \subset \mathbb{R}^n \quad (1.21)$$

then the equation admits an exact solution. In most cases, however, $y \notin \mathcal{M}(X)$, so no exact solution exists. Nevertheless, we can always find an estimator \hat{w} such that $\mathcal{P}_{\mathcal{M}(X)}y = X\hat{w}$, where $\mathcal{P}_{\mathcal{M}(X)}$ denotes the orthogonal projection onto the hypersurface $\mathcal{M}(X)$.

Proposition 1.3

$$\hat{y} = X\hat{w} \quad \Rightarrow \quad \hat{w} \text{ is solution to LR.} \quad (1.22)$$

Proof

$$\begin{aligned} y - \hat{y} \perp \mathcal{M}(X) &\Rightarrow y - X\hat{w} \perp \mathcal{M}(X) \\ &\Rightarrow X^\top(y - X\hat{w}) = 0 \quad \Rightarrow \quad \hat{w} = (X^\top X)^{-1}X^\top y \end{aligned}$$

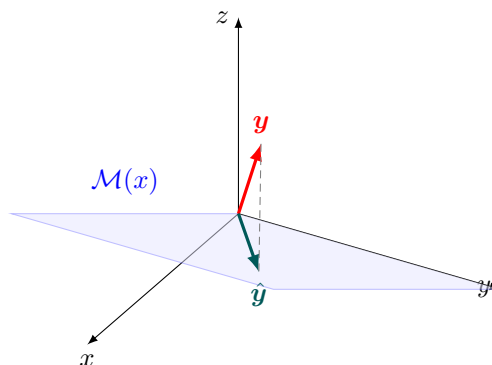


Figure 1.3: Orthogonal projection interpretation of linear regression. The predicted vector $X\hat{w}$ is obtained as the projection of y onto the hypersurface $\mathcal{M}(X) = \{Xw : w \in \mathbb{R}^d\}$, which is a linear subspace in the classical case.