

Linear Models & Logistic Regression

Qing-Yuan Jiang

The school of Computer Science and Engineering, NJUST

2026.10.08

Review of Previous Class

- Why do we need model selection?
- What is the generalization?
- How do we separate training, validation, and test data?
- What is overfitting and underfitting?
- How do we detect overfitting and underfitting?
- How do we evaluate a model on unseen data?
- Metrics for classification task.
- Metrics for regression task.

Linear Model

From Data to Prediction

- Main goal of machine learning:

$$x \xrightarrow{f_\theta} \hat{y}$$

- $\mathcal{X}$ input data/features.
- f_θ models with parameter θ .
- $\hat{y}$ predictive label.
- What form should f_θ take?
 - **Modeling**

Why Linear Model?

- Simple:
 - Few parameters
 - Easy to train
- Interpretable:
 - Parameters have clear meanings
 - Feature effects can be examined directly
- Fundamental:
 - Provides a foundation for many machine learning methods
 - Connects naturally to optimization and probabilistic modeling

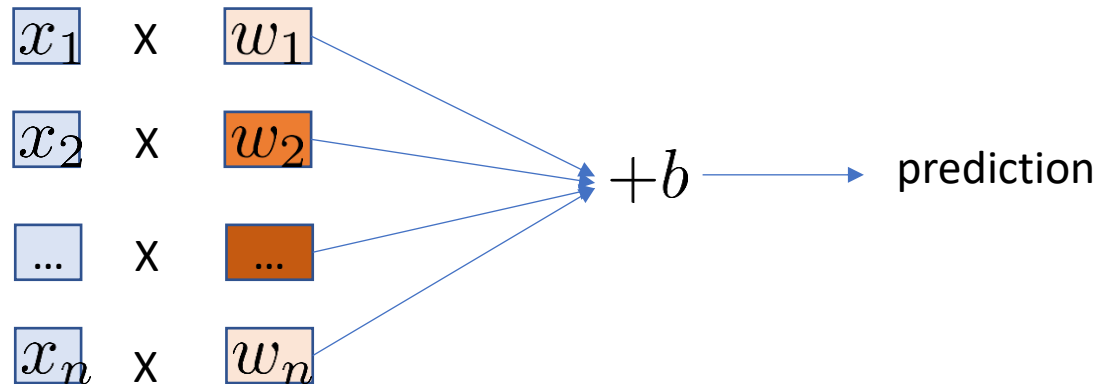
Linear models provide one of the simplest ways to learn from data.

The Basic Idea of Linear Models

- The Basic Idea:

Assign different weights to different features and combine them linearly.

$$f(x) = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$$



Linear Model

- Definition of Linear Model:

$$f(x) = wx + b$$

- $w \in \mathcal{R}^d$: weight
- $x \in \mathcal{R}^d$: input
- $b \in \mathcal{R}$: bias
- Key question:
 - How should we determine w and b ?

Example of House Price Prediction

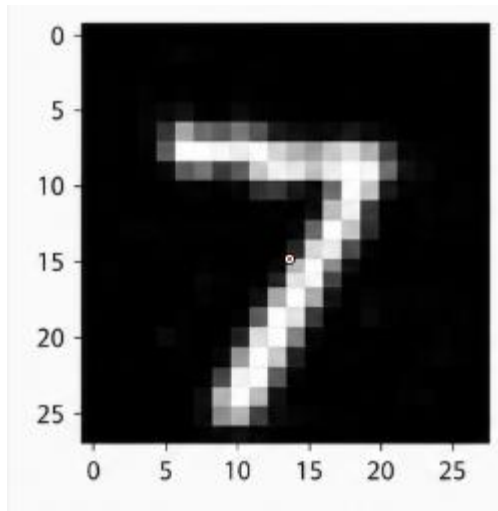
- Task: predicting the price of the house based on the given features:

$$x = [\text{area}, \text{num_of_rooms}, \text{distance}, \text{age}]$$

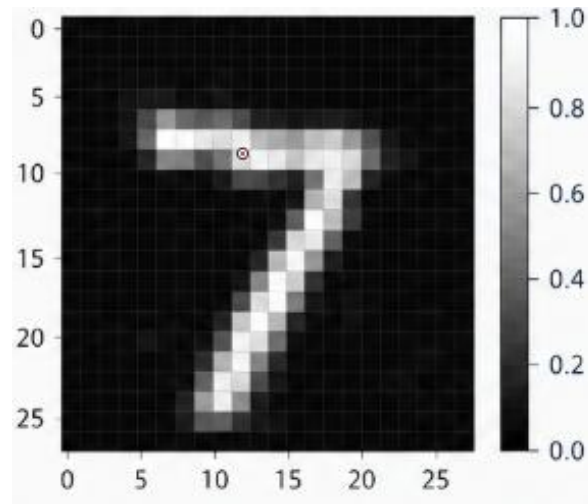
- w_1 : contribution of area
- w_2 : contribution of the number of rooms.
- w_3 : contribution of distance.
- w_4 : contribution of age.
- **High price** is associated with:
 - Large area
 - Large number of rooms
 - Small distance to the CBD
 - Short building age

Examples on MNIST

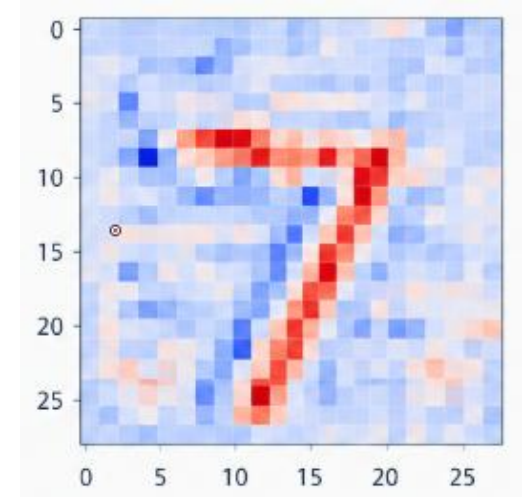
- Recall the detail of MNIST dataset
 - Pixel with size 28 X 28
- Learn a linear model to fit digits
 - We employ the one-hot label
 - $\hat{y} = W \times x + b$
 - $W \in \mathcal{R}^{10 \times 784}$



original image



normalized image



learned parameter

Geometric Interpretation

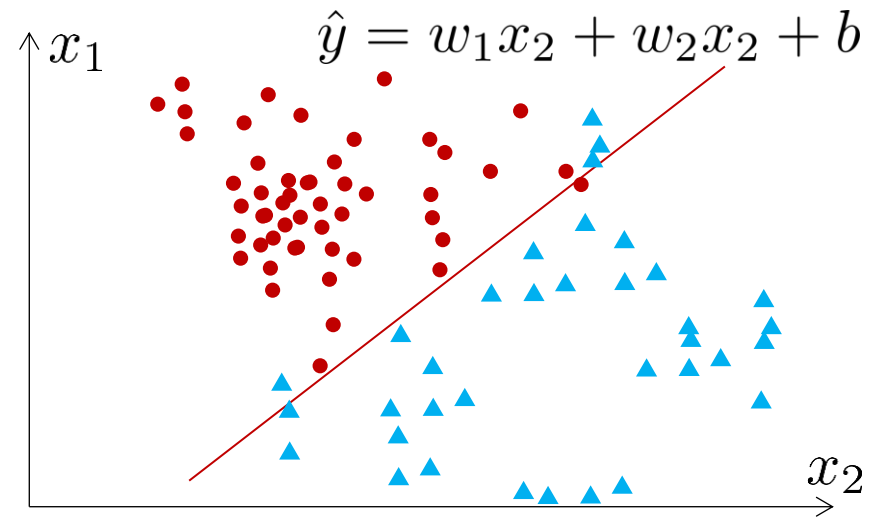
- What Does a Linear Model Represent?
- Two dimensions case: a line

$$\hat{y} = w_1x_1 + w_2x_2 + b$$

- Three dimensions case: a plane

$$\hat{y} = w_1x_1 + w_2x_2 + w_3x_3 + b$$

- Dims > 3: a hyperplane



Linear Regression

Regression Problem

- Regression:
 - Predict a continuous numerical value.
 - That is:

$$y \in \mathcal{R}$$

- Examples:
 - House price
 - Temperature
 - Demand

Linear Regression

- Definition:
 - Given dataset:

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$$

- Assume:

$$y_i \approx w^\top x_i + b$$

- Hence:

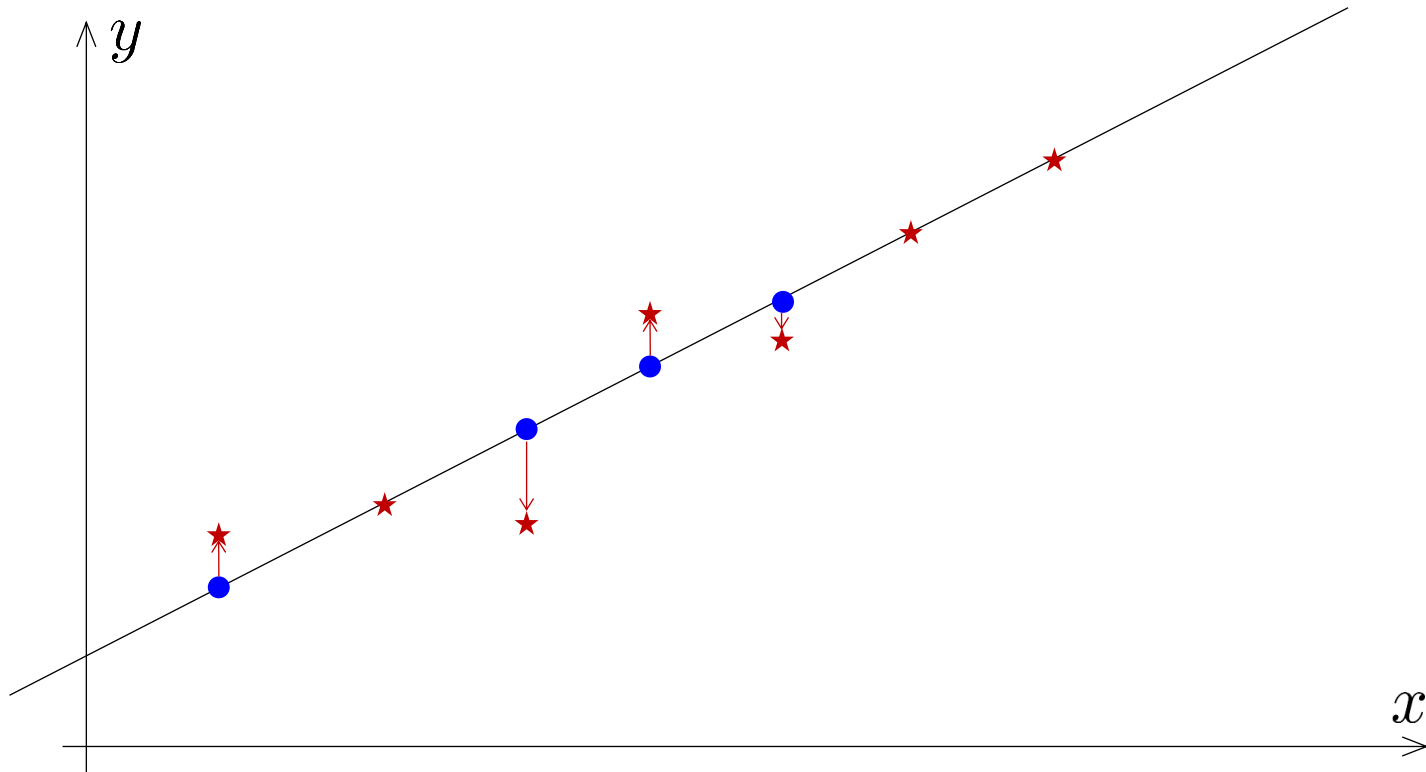
$$\hat{y}_i = w^\top x_i + b$$

- How can we find **the best line**?

Predictive Error

- Error between predictive value and ground-truth label:

$$\begin{aligned}e_i &= y_i - \hat{y}_i \\ &= y_i - (w^\top x_i + b)\end{aligned}$$



What Is a Good Regression Model?

- Core problem:
 - A good model should make the prediction errors **small**.
- Hence:

$$\min_{w,b} \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - w^\top x_i - b)^2$$

- Why do we use the squared error as the optimization objective?
 - Remove the sign
 - Penalize the large value
 - Smooth and differentiable

Least Squares

- Least Squared Problem:

$$\min_{w,b} \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - w^\top x_i - b)^2$$

- Convex optimization problem (w.r.t w, b)?
- How to solve it?
 - Closed-form solution
 - Gradient-relevant methods
- Large errors receive larger penalties.

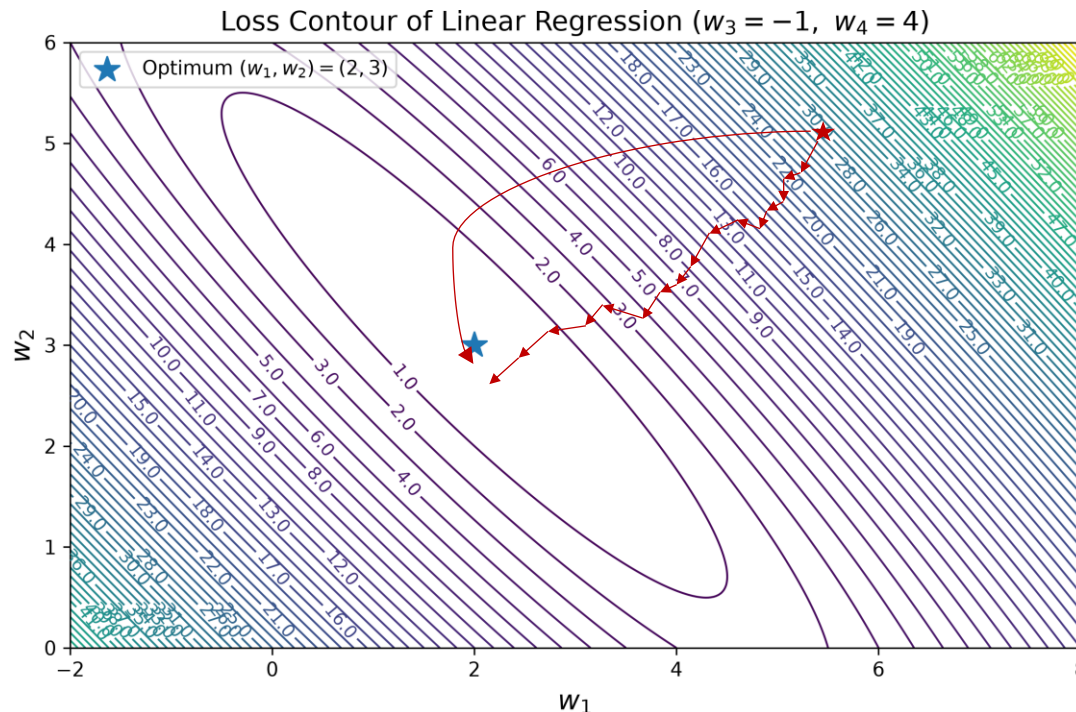
Optimization for Least Squared Problem

- Two Ways to Learn a Linear Regression Model
 - Closed-form solution:

$$w^* = (X^\top X)^{-1} X^\top Y$$

- Iterative optimization:

$$w_{t+1} = w_t - \eta \nabla \ell_w$$



Logistic Regression

From Regression to Classification

- Regression task:

$$y \in \mathcal{R}$$

- Classification task:
 - Label belongs to a discrete set
 - E.g.,

$$y \in \{0, 1\}$$

- Examples:
 - Spam email detection
 - Digit recognition
 - Natural image recognition

Linear Classification

- Regression task:

$$y \in \mathcal{R}$$

- Classification task:
 - Label belongs to a discrete set
 - E.g.,

$$y \in \{0, 1\}$$

- Examples:
 - Spam email detection
 - Digit recognition
 - Natural image recognition

Linear Classification

- Classifier:

$$f(x) = w^\top x + b$$

- Rules for classification:

$$\hat{y} = \begin{cases} 1, & f(x) > 0, \\ 0, & f(x) \leq 0. \end{cases}$$

- Decision boundary:

$$w^\top x + b = 0$$

The Problem of Linear Score

- Problem:

$$f(x) = w^{\top} x + b$$

- The range of output:

$$(-\infty, +\infty)$$

- Probability:

$$[0, 1]$$

- We need a function that converts a **real-valued score** into a **probability**.

From Linear Score to Probability

- Definition:

$$z = w^{\top} x + b$$

- We hope:

$$z \mapsto p$$

- Here, $0 \leq p \leq 1$
- Logistic function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Logistic Function

- Formulation:

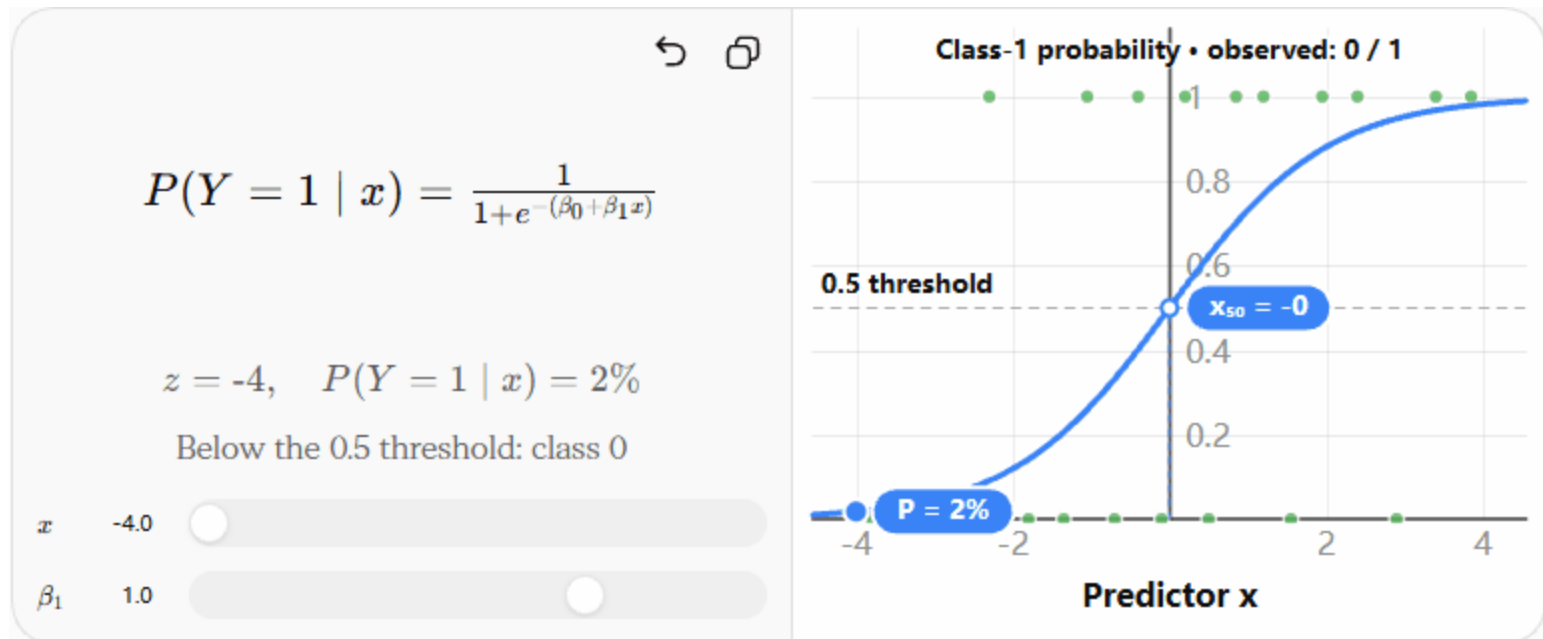
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- Property:

$$\sigma(-\infty) = 0$$

$$\sigma(0) = 0.5$$

$$\sigma(+\infty) = 1$$



Logistic Regression

- Logistic Regression:

$$\begin{aligned} p(y = 1|x) &= \sigma(w^\top x + b) \\ &= \frac{1}{1 + e^{-(w^\top x + b)}} \end{aligned}$$

- Linear model + logistic function $\rightarrow$ logistic regression

- Since:

$$\sigma(0) = 0.5$$

- We have:

$$p(y = 1|x) = 0.5 \iff w^\top x + b = 0$$

- Logistic regression has a **linear decision boundary**.

How About Squared Error?

- Can We Use the Same Loss as Linear Regression?
- Linear Regression:

$$\ell = (y - \hat{y})^2$$

- Logistic Regression:

$$\begin{aligned}\ell &= (y - \hat{y})^2 \\ &= (y - \sigma(w^\top x + b))^2\end{aligned}$$

- Squared loss is a possible choice, but not the **natural** choice for logistic regression.

Maximum Likelihood

- Review the definition of probability:

$$p(y = 1|x) = \frac{1}{1 + e^{-(w^\top x + b)}}$$

$$p(y = 0|x) = 1 - p(y = 1|x)$$

- For a given sample:

$$P(y_i|x_i; \{w, b\}) = p_i^{y_i} (1 - p_i)^{1-y_i}$$

- Which value of $\{w, b\}$ provides **the best fit** to the observed data?
- Likelihood over dataset:

$$L(\{w, b\}|D) = P(D|\{w, b\})$$

$$= \prod_{i=1}^N p_i^{y_i} (1 - p_i)^{1-y_i}$$

Maximum Likelihood

- Maximum Likelihood Estimation:

$$\theta^* = \arg \max_{\theta} L(\theta|D)$$

- Choose parameters that make the observed labels as **probable** as possible.
- Taking log for simplicity:

$$\begin{aligned}\log L(\theta|D) &= \log \left(\prod_{i=1}^N p_i^{y_i} (1 - p_i)^{1-y_i} \right) \\ &= \sum_{i=1}^N \log (p_i^{y_i} (1 - p_i)^{1-y_i}) \\ &= \sum_{i=1}^N (y_i \log p_i + (1 - y_i) \log(1 - p_i))\end{aligned}$$

Maximum Likelihood

- Log-Likelihood

$$\begin{aligned}\theta^* &= \arg \max_{\theta} L(\theta|D) \\ &= \arg \max_{\theta} \log L(\theta|D) \\ &= \arg \min_{\theta} -\log L(\theta|D) \\ &= \arg \min_{\theta} -\sum_i (y_i \log p_i + (1 - y_i) \log(1 - p_i))\end{aligned}$$

- We get binary cross entropy loss

Why Cross-Entropy Loss

- Penalty comparison:

$p(y = 1)$	MSE	Cross-Entropy
0.1	0.81	2.3
0.01	0.9801	4.61
0.001	0.9980	6.91
0.000001	0.999998	13.82

- Confident and wrong predictions are **strongly penalized**.

Learning Logistic Regression

- We get:

$$\min_{w,b} \ell(w, b) = - \sum_i (y_i \log p_i + (1 - y_i) \log(1 - p_i))$$

- Learning algorithm:
 - Stochastic gradient descent/gradient descent/mini-batch GD:

$$\theta_{t+1} = \theta_t - \eta \nabla \ell(\theta_t)$$

- The learning process follows the same optimization framework introduced in Lecture 2.

Linear Regression vs. Logistic Regression

	Linear Regression	Logistic Regression
Task	Regression	Classification
Target	$\mathcal{R}$	Discrete set
Score	$w^\top x + b$	$w^\top x + b$
Output	Target score	Probability
Mapping	Identity	Logistic function
Loss	Squared loss	Cross-entropy loss
Boundary	$w^\top x + b$	$w^\top x + b$

Advantages of Linear Models

- **Simple**
 - Few parameters and simple structure.
- **Efficient**
 - Fast training and prediction.
- **Interpretable**
 - Parameters provide direct information about feature effects.
- **Strong Baseline**
 - Useful as a baseline for more complex models.

Limitations of Linear Models

- **Limited Representation**
 - Linear models cannot directly capture complex nonlinear relationships.
- **Feature Dependence**
 - Performance can strongly depend on feature quality.
- **Limited Interaction Modeling**
 - Complex feature interactions may need to be explicitly constructed.

Thanks