ML&MEA (2024)
Lecture 2 - Linear regression

Quanying Liu

BME, SUSTech

2024.2.22

Content

Recap Lecture 1

- Three topics: AI $>$ ML $>$ DL

Recap Lecture 1

- Three topics: AI $>$ ML $>$ DL
- Three categories of machine learning

Recap Lecture 1

- Three topics: AI $>$ ML $>$ DL
- Three categories of machine learning
- Three key components of machine learning

Recap Lecture 1

- Three topics: AI $>$ ML $>$ DL
- Three categories of machine learning
- Three key components of machine learning
  1. Data: $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, $x^{(i)} \in \mathbb{R}^p$ or (X, Y)

Recap Lecture 1

- Three topics: AI $>$ ML $>$ DL
- Three categories of machine learning
- Three key components of machine learning
  1. Data: $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, $x^{(i)} \in \mathbb{R}^p$ or (X, Y)
  2. Model: a function from input to output $y = f(x, \theta)$, a set of parameters θ, a loss function $\mathcal{L}(\theta)$

Recap Lecture 1

- Three topics: AI $>$ ML $>$ DL
- Three categories of machine learning
- Three key components of machine learning
  1. Data: $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, $x^{(i)} \in \mathbb{R}^p$ or (X, Y)
  2. Model: a function from input to output $y = f(x, \theta)$, a set of parameters θ, a loss function $\mathcal{L}(\theta)$
  3. Optimization algorithm: analytical solution, grid search, gradient descent, and so on

What is machine learning? Answer from Wikipedia

"Machine learning (ML) is the study of computer algorithms that improve automatically through experience." (Wikipedia)

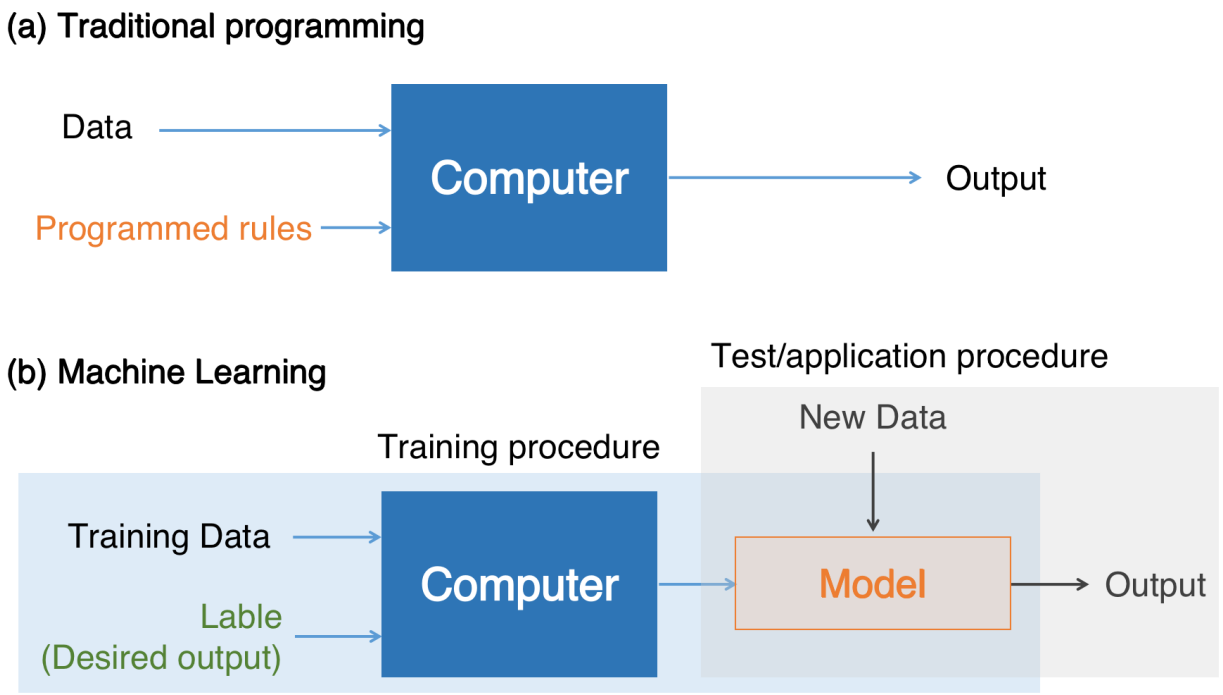


图 1: Traditional Programming vs Machine Learning

What is machine learning? Answer from GPT3.5

Machine learning is a subset of artificial intelligence (AI) that involves the development of algorithms and statistical models that enable computers to learn and improve their performance on a specific task without being explicitly programmed. In other words, machine learning algorithms use data to learn patterns, make predictions, or decisions, and adapt their behavior accordingly.
The process of machine learning typically involves the following steps:
1. **Data Collection:** Gathering relevant data from various sources, such as sensors, databases, or the internet.
2. **Data Preprocessing:** Cleaning, transforming, and preparing the data for analysis by handling missing values, scaling features, and encoding categorical variables.
3. **Feature Engineering:** Selecting or extracting the most relevant features from the data to represent the underlying patterns effectively.
4. **Model Selection:** Choosing an appropriate machine learning model or algorithm based on the nature of the problem and the available data. Common types of machine learning models include supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning.
5. **Model Training:** Using the prepared data to train the selected model by adjusting its parameters to minimize the error or maximize the performance metric.
6. **Model Evaluation:** Assessing the performance of the trained model using evaluation metrics and validation techniques to ensure its generalization to unseen data.
7. **Model Deployment:** Deploying the trained model into production environments to make predictions or decisions on new, unseen data.
Machine learning techniques are widely used in various applications, including but not limited to:
- Predictive analytics
- Image and speech recognition
- Natural language processing
- Recommendation systems
- Autonomous vehicles
- Medical diagnosis
- Financial forecasting
Overall, machine learning enables computers to learn from data, identify patterns, and make decisions or predictions with increasing accuracy over time, making it a powerful tool for solving complex problems across different domains.

Goals of ML

"The goal of machine learning is to develop methods that can automatically detect patterns in data, and then to use the uncovered patterns to predict future data to other outcomes of interests." (MLAPP)

- Detect patterns: e.g., detecting objects (dog/cat), diseases...
- Predict future data: e.g., predicting possibility of a disease in 3 years, next-step fMRI signals, next word in a sentence
- Reconstruct data: e.g., reconstructing MRI image, EEG time course
- Generate new data: e.g., synthesising images, sentence, ...
- ...

ML vs. Statistics

Both statistics and machine learning aim to "detect patterns in data" by building *predictive models*.

Statistics: use the model as a tool to learn something about the world (statistical inference).

- Focus on simple, interpretable models.
- Develop theoretical analysis, work out statistical guarantees under some assumptions.

Machine learning: use the model as a tool to actually make useful predictions.

- Focus on complicated, competitive models.
- Use large datasets. Be pragmatic（讲究实效的）. 'Almost' give up on inference.

Notations

Notations	Description
$\mathbb{R}$	set of all real numbers
$\mathbb{R}^n$	set of all real vectors of dimension $n \times 1$
$\mathbb{R}^{n \times p}$	set of all real matrices of dimension $n \times p$
a, b, c, i, j, k	lower-case letter as a scalar
$\mathbf{a}, \mathbf{b}, \mathbf{c}$	bold, lower-case letter as a vector (列向量)
a_i	ith element of vector $\mathbf{a}$
$\mathbf{A}, \mathbf{B}, \mathbf{C}$	bold, upper-case letter as a matrix

Two ways to describe a matrix $\mathbf{X}$ (矩阵的两种描述形式):

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}^{(1)T} \\ \mathbf{x}^{(2)T} \\ \vdots \\ \mathbf{x}^{(n)T} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \cdots & \mathbf{x}_p \end{pmatrix} \in \mathbb{R}^{n \times p}$$

What is linear regression?

Linear regression is a supervised learning problem, a regression (not classification) task.

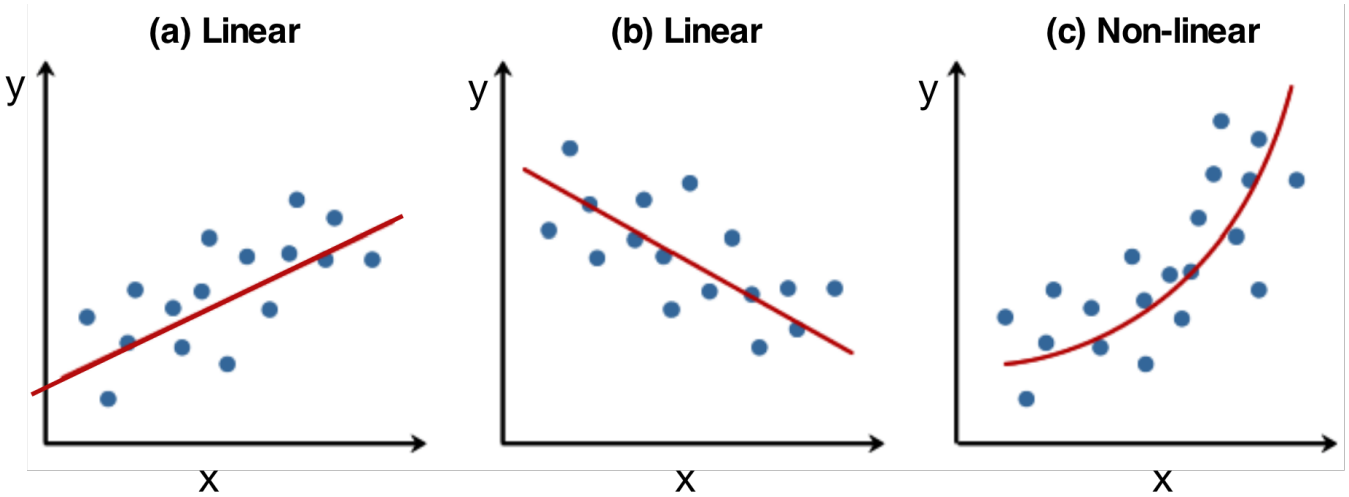


图 2: Linear regression vs Nonlinear regression

Q: Can you take some examples to use linear regression in real applications?

Linear regression

- Data: $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$
  $x^{(i)} \in \mathbb{R}^{p\times 1}$ (p features in a data); $y^{(i)} \in \mathbb{R}^{1\times 1}$ (label)
  $x = [x_1\ x_2 \dots x_p]^T$ is a subject's data (*e.g.*, $x^{(i)}$, ith sub)

Linear regression

- Data: $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$
  $x^{(i)} \in \mathbb{R}^{p\times 1}$ (p features in a data); $y^{(i)} \in \mathbb{R}^{1\times 1}$ (label)
  $x = [x_1\ x_2 \dots x_p]^T$ is a subject's data (*e.g.*, $x^{(i)}$, ith sub)
- Model: a linear function $f(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$
  Parameters: $\beta = [\beta_0\ \beta_1 \dots \beta_p]^T$
  Loss function (aka cost function): We use mean-squared-error (MSE) loss here.

$$\begin{aligned}\mathcal{L}(\beta) &= \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - f(x^{(i)})\right)^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \beta_0 - \beta_1 x_1^{(i)} - \cdots - \beta_p x_p^{(i)}\right)^2\end{aligned} \tag{1}$$

Linear regression

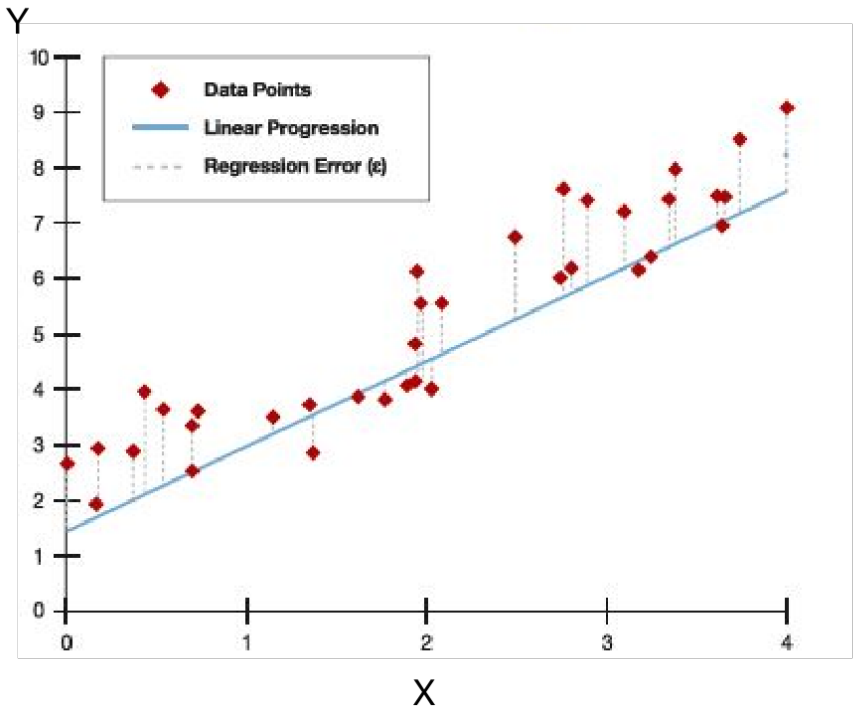
- Data: $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$
  $x^{(i)} \in \mathbb{R}^{p\times 1}$ (p features in a data); $y^{(i)} \in \mathbb{R}^{1\times 1}$ (label)
  $x = [x_1\ x_2 \dots x_p]^T$ is a subject's data (*e.g.*, $x^{(i)}$, ith sub)
- Model: a linear function $f(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$
  Parameters: $\beta = [\beta_0\ \beta_1 \dots \beta_p]^T$
  Loss function (aka cost function): We use mean-squared-error (MSE) loss here.

$$\begin{aligned}\mathcal{L}(\beta) &= \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - f(x^{(i)})\right)^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \beta_0 - \beta_1 x_1^{(i)} - \cdots - \beta_p x_p^{(i)}\right)^2\end{aligned} \tag{1}$$

- Optimization: finding the best β to fit the model to the data.

Visualize loss

We want to fit the model to the data.
"To fit the model" means to find a β so that $f(x^{(i)}) \approx y^{(i)}$ for all i, which means to "minimize the MSE loss".



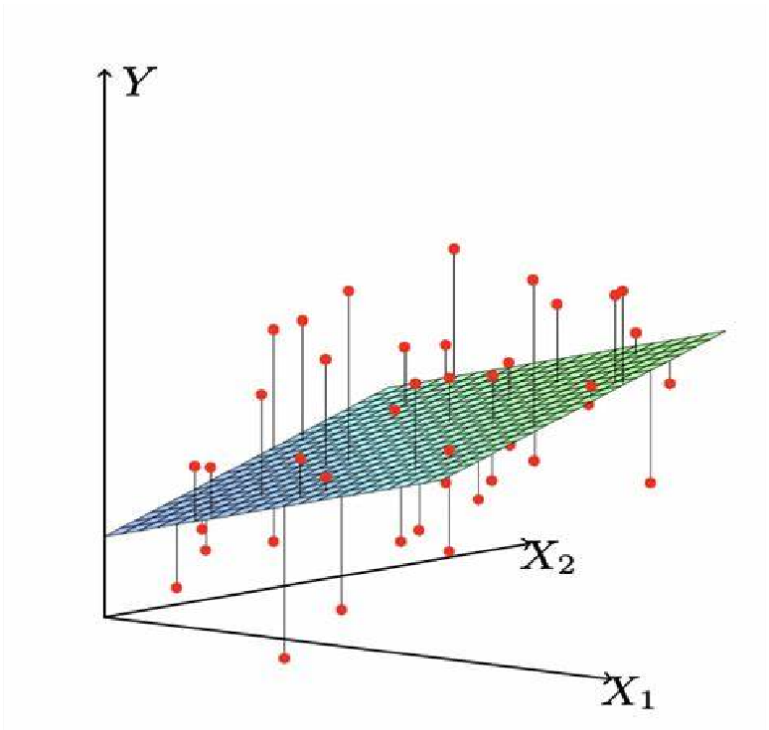


图 3: Visualization of MSE loss

The simplest linear regression

The simplest linear regression model: $y = \beta x$

Parameter: the slope β

Loss function:

$$\mathcal{L}(\beta) = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \beta x^{(i)}\right)^2 \tag{2}$$

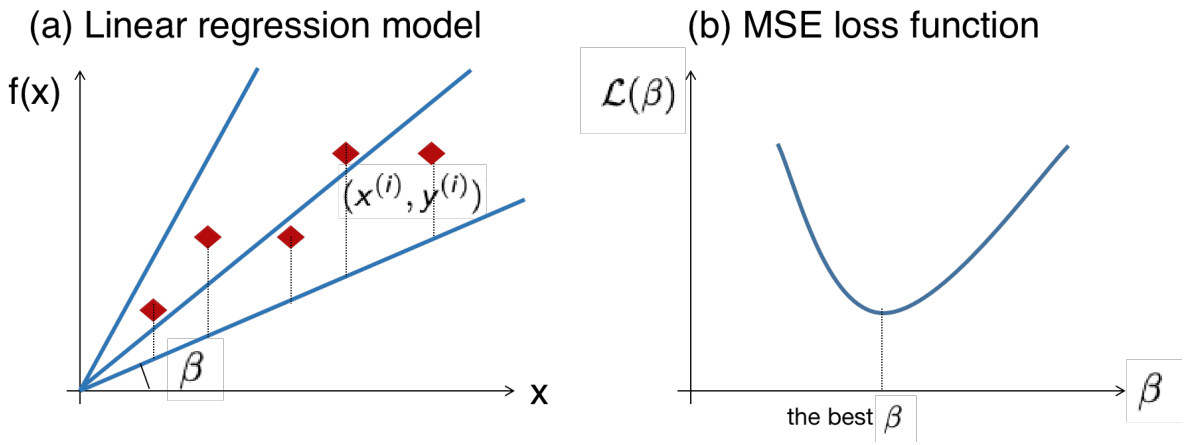


图 4: The simplest linear regression with one parameter

Compute the derivative of the loss

The simplest linear regression model: $y = \beta x$
Parameter: the slope β
Loss function:

$$\mathcal{L}(\beta) = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \beta x^{(i)}\right)^2.$$

The derivative of the loss:

$$\mathcal{L}'(\beta) = \frac{1}{n}\sum_{i=1}^{n} 2\left(y^{(i)} - \beta x^{(i)}\right)\left(-x^{(i)}\right) = -\frac{2}{n}\sum_{i} x^{(i)}\left(y^{(i)} - \beta x^{(i)}\right). \tag{3}$$

Analytical solution

The derivative of the loss:

$$\mathcal{L}'(\beta) = \frac{1}{n}\sum_{i=1}^{n} 2\left(y^{(i)} - \beta x^{(i)}\right)\left(-x^{(i)}\right) = -\frac{2}{n}\sum_{i} x^{(i)}\left(y^{(i)} - \beta x^{(i)}\right).$$

At the minimum:

$$\mathcal{L}'(\beta) = 0,$$
$$-\frac{2}{n}\sum_{i} x^{(i)}\left(y^{(i)} - \beta x^{(i)}\right) = 0.$$

We obtain the analytical solution of the simple linear regression, $\hat{\beta}$,

$$\hat{\beta} = \frac{\sum_{i=1}^{n} x^{(i)} y^{(i)}}{\sum_{i=1}^{n} \left(x^{(i)}\right)^2} \tag{4}$$

Gradient Descent

Update rule:

$$\beta^{(j+1)} \longleftarrow \beta^{(j)} - \eta \frac{d\mathcal{L}(\beta)}{d\beta}. \tag{5}$$

Here η is the *learning rate*.
Choosing a good η is important: (i) too small – slow convergence; (ii) too large – divergence.

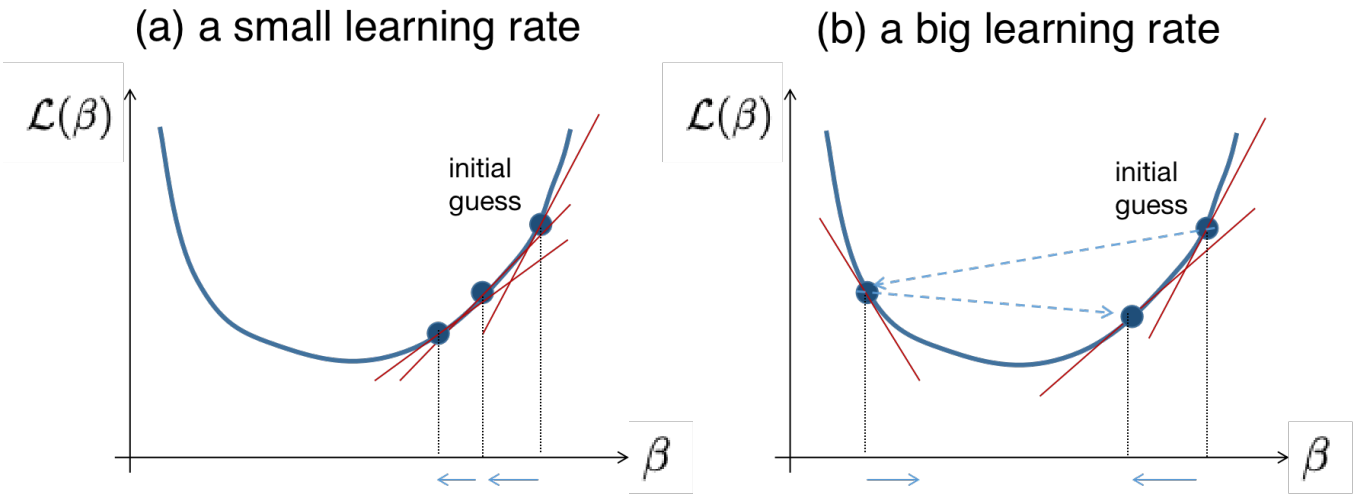


图 5: Effects of learning rate

Stopping criterion

Convergence:

$$\beta^{(j+1)} - \beta^{(j)} < \epsilon. \tag{6}$$

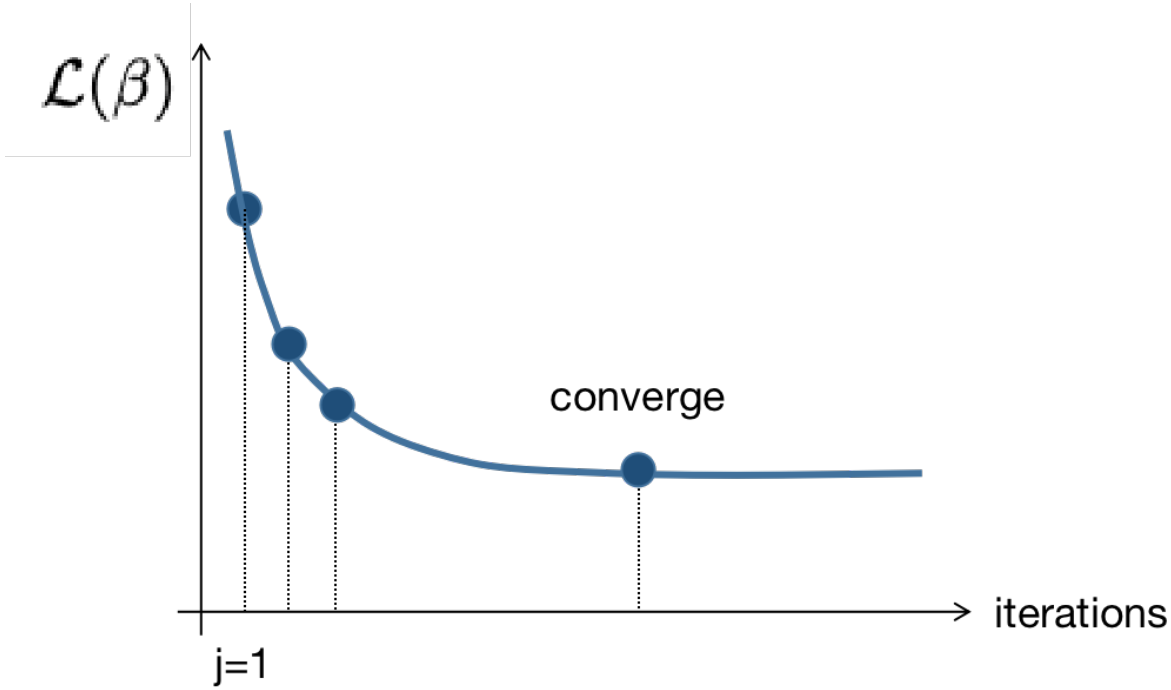


图 6: Stop when the loss function converges.

Multiple linear regression

Data: $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}, \quad \mathbf{x}^{(i)} = [x_1\ x_2 \dots x_p]^T$

Model: $f(\mathbf{x}^{(i)}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$

Parameters: $\beta = [\beta_0\ \beta_1 \dots \beta_p]^T$

It is convenient to define $x_0 \equiv 1, \quad \mathbf{x} = [x_0\ x_1 \dots x_p]^T$.

Then the model can be written as:

$$f(\mathbf{x}, \beta) = \beta^T \mathbf{x} = (\beta_0 \cdots \beta_p) \begin{pmatrix} x_0 \\ \vdots \\ x_p \end{pmatrix} \in \mathbb{R}^{1\times 1} \tag{7}$$

The loss and the gradient

Using this notation, the MSE loss function becomes:

$$\mathcal{L}(\beta) = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \beta^T \mathbf{x}^{(i)}\right)^2 \tag{8}$$

Partial derivatives of the loss function w.r.t. β_k:

$$\frac{\partial \mathcal{L}}{\partial \beta_k} = -\frac{2}{n}\sum_{i=1}^{n}\left(y^{(i)} - \beta^T \mathbf{x}^{(i)}\right) x_k^{(i)} \tag{9}$$

Gradient of the loss:

$$\nabla \mathcal{L} = \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial \beta_1} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial \beta_p} \end{pmatrix} = -\frac{2}{n}\sum_{i=1}^{n}\left(y^{(i)} - \beta^T \mathbf{x}^{(i)}\right) \mathbf{x}^{(i)} \tag{10}$$

Design matrix: $\mathbf{X} \in \mathbb{R}^{n \times p}$

Design matrix is a collection of feature vectors $\mathbf{x}^{(i)}$ for different data points $i = 1, \ldots, n$.

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}^{(1)T} \\ \mathbf{x}^{(2)T} \\ \vdots \\ \mathbf{x}^{(n)T} \end{pmatrix} = \begin{pmatrix} x_0^{(1)} & x_1^{(1)} & \cdots & x_p^{(1)} \\ x_0^{(2)} & x_1^{(2)} & \cdots & x_p^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_0^{(n)} & x_1^{(n)} & \cdots & x_p^{(n)} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \cdots & \mathbf{x}_p \end{pmatrix}$$

Each row of the matrix is one data point (i.e., one feature vector), and each column represents the values of a given feature across all of the data points.

Response vector: $\mathbf{y} \in \mathbb{R}^{n \times 1}$

Let us collect all $y^{(i)}$ values into a *response vector*:

$$\mathbf{y} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{pmatrix} \in \mathbb{R}^{n \times 1}$$

The design matrix $\mathbf{X}$ and response vector $\mathbf{y}$ are the basic data objects on which machine learning algorithms operate.

Model as Matrix multiplication

Let us rewrite the linear regression model in a matrix multiplication manner:

$$\mathbf{y} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{pmatrix} = \begin{pmatrix} x_0^{(1)} & x_1^{(1)} & \cdots & x_p^{(1)} \\ x_0^{(2)} & x_1^{(2)} & \cdots & x_p^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_0^{(n)} & x_1^{(n)} & \cdots & x_p^{(n)} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} \tag{11}$$

That is,

$$\mathbf{y} = \mathbf{X}\beta \tag{12}$$

Loss function as Matrix calculus

L2 norm: On the n-dimensional Euclidean space $\mathbb{R}^n$, the length of the vector $\mathbf{y}$ is its *L2 norm*, which is defined as

$$\|\mathbf{y}\|_2 := \sqrt{\sum_{i=1}^{n} y_i^2} = \sqrt{y_1^2 + y_2^2 + \cdots + y_n^2} = \sqrt{\mathbf{y}^T \mathbf{y}}$$

Now we can write the loss function as:

$$\mathcal{L}(\beta) = \frac{1}{n} \sum_{i=1}^{n} \left(y^{(i)} - \beta^T \mathbf{x}^{(i)} \right)^2 = \frac{1}{n} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 \tag{13}$$

Another way to write the loss function is,

$$\mathcal{L}(\beta) = \frac{1}{n} (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) \tag{14}$$

Gradient of loss function

Recall Eq. (10), the gradient of loss function is:

$$\nabla \mathcal{L} = -\frac{2}{n} \sum_{i=1}^{n} \left(y^{(i)} - \beta^T \mathbf{x}^{(i)} \right) \mathbf{x}^{(i)} \in \mathbb{R}^{p \times 1}$$

We re-write the gradient as:

$$\nabla \mathcal{L} = -\frac{2}{n} \begin{pmatrix} \mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \cdots & \mathbf{x}^{(n)} \end{pmatrix} \begin{pmatrix} y^{(1)} - \beta^T \mathbf{x}^{(1)} \\ y^{(2)} - \beta^T \mathbf{x}^{(2)} \\ \vdots \\ y^{(n)} - \beta^T \mathbf{x}^{(n)} \end{pmatrix}$$

$$= -\frac{2}{n} \mathbf{X}^T (\mathbf{y} - \mathbf{X}\beta)$$

Analytical solution

The gradient of loss function is:

$$\nabla \mathcal{L} = -\frac{2}{n}\mathbf{X}^T\left(\mathbf{y} - \mathbf{X}\beta\right) \tag{15}$$

Let us set $\nabla \mathcal{L} = \mathbf{0}$ to derive the analytical solution $\hat{\beta}$:

$$-\frac{2}{n}\mathbf{X}^T\left(\mathbf{y} - \mathbf{X}\hat{\beta}\right) = \mathbf{0}$$
$$\mathbf{X}^T\mathbf{y} - \mathbf{X}^T\mathbf{X}\hat{\beta} = \mathbf{0}$$

The analytical solution of multiple linear regression is:

$$\hat{\beta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} \tag{16}$$

Q: What are the possible problems to calculate the analytical solution?

Gradient descent

Update the parameter vector $\beta \in \mathbb{R}^{p\times 1}$

$$\beta_0^{(j+1)} \longleftarrow \beta_0^{(j)} - \eta \frac{\partial \mathcal{L}}{\partial \beta_0}$$

$$\cdots$$

$$\beta_p^{(j+1)} \longleftarrow \beta_p^{(j)} - \eta \frac{\partial \mathcal{L}}{\partial \beta_p}$$

Rewrite it in vector form:

$$\beta^{(j+1)} \longleftarrow \beta^{(j)} - \eta \nabla \mathcal{L} \tag{17}$$

Recall Eq. (15), the gradient of loss function is:

$$\nabla \mathcal{L} = -\frac{2}{n}\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta)$$

Summary of Lecture 2 - Linear regression

- Simple linear regression
  1. Data: $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, $x^{(i)} \in \mathbb{R}^1$, $y^{(i)} \in \mathbb{R}^1$
  2. Model: $y = \beta x$ with single parameter β;
     Loss function $\mathcal{L}(\beta) = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \beta x^{(i)}\right)^2$
  3. Optimization algorithms
     Analytical solution: $\hat{\beta} = \frac{\sum_{i=1}^{n} x^{(i)} y^{(i)}}{\sum_{i=1}^{n}\left(x^{(i)}\right)^2}$
     Gradient descent: $\beta^{(j+1)} \longleftarrow \beta^{(j)} - \eta \frac{d\mathcal{L}(\beta)}{d\beta}$
- Multiple linear regression
  1. Data: $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, $\mathbf{x}^{(i)} = [x_1\ x_2 \ldots x_p]^T$
  2. Model: $\mathbf{y} = \mathbf{X}\beta$;
     Loss function $\mathcal{L}(\beta) = \frac{1}{n}\left\|\mathbf{y} - \mathbf{X}\beta\right\|^2 = \frac{1}{n}\left(\mathbf{y} - \mathbf{X}\beta\right)^T\left(\mathbf{y} - \mathbf{X}\beta\right)$
  3. Optimization algorithms
     Analytical solution: $\hat{\beta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$
     Gradient descent: $\beta^{(j+1)} \longleftarrow \beta^{(j)} - \eta \frac{\partial\mathcal{L}(\beta)}{\partial\beta}$