# 1900 页 34 章数学方法精要笔记，

# ——深入数学建模， 机器学习和深度学习的数学基础

## 书籍介绍

在信息爆炸的当今，大到企业巨头的经营方向， 小到和日常生活相关的人工驾驶等领域，数学建模和人工智能都对信息数据的收集、处理、解释以及做出决策将起到至关重要的作用。负责开发模型和算法的一线科学家和工程师， 都需要有坚实的数学基础。 相信有许多所有对数学建模，机器学习和深度学习深感兴趣的小伙伴，有一定的基础却常常被繁杂的定理和错综的模型所困—— 那么这本书就是一部可供随时查阅，帮助大家融会贯通的宝典。

本书有以下几大亮点：

1. 理论与实践相结合，学以致用。内容详尽，涵盖范围广。

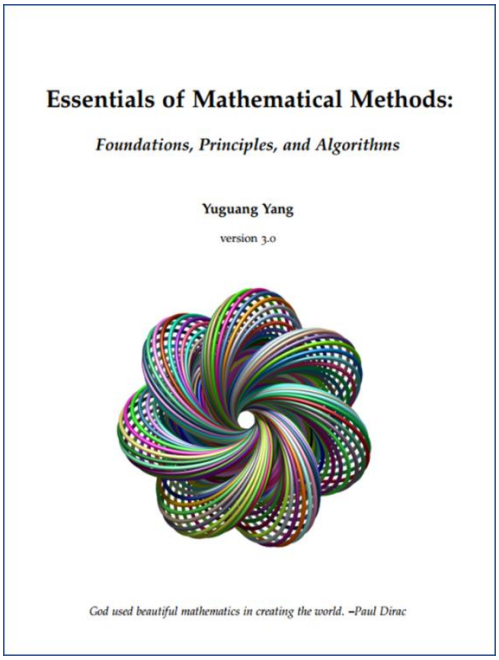
    a. 全书页数近两千页覆盖范围广， 包含~100 个核心算法， 约 3 00 个示意图。例子丰富，且绝大部分定理都有证明。

    b. 本书凝聚了作者多年数学建模和机器学习研究和实战经验。根据应用领域， 本书总结并深入讲述传统方法到前沿的深度学习和强化学习算法，帮助读者迅速抓住重点，减少弯路。

2. 便于学习查找，由浅入深，步步为营，多用示意图以助读者理解

    a. 本书的算法和定理证明中常常引用相关的其他章节, 循序渐进，有助于读者建立树状知识脉络，一网打尽相关知识点。

    b. 本书例子详实并多伴有示意图，清晰易懂。作者基于多年实践，总结并对易混淆的概念进行比对，帮助读者更加扎实掌握相关内容。




全书 GitHub 地址： https://github.com/yangyutu/EssentialMath

全书总共 34 章分成六个部分：

**I Mathematical Foundations （数学基础）**

**II Mathematical Optimization Methods （数学优化方法）**

**III Classical Statistical Methods （经典统计方法）**

**IV Dynamics Modeling Methods （动力系统建模方法）**

**V Statistical Learning Methods （统计学习方法）**

**VI Optimal Control and Reinforcement Learning Methods （最优控制和强化学习方法）**

作者对一些热门章节进行章节归类打包下载

- **Linear Algebra and Matrix Analysis**
- **Mathematical Optimization**
- **Probability and Statistical Estimation**
- **Stochastic Process**
- **Markov Chain and Random Walk**
- **Linear Regression Analysis**
- **Statistical Learning**
- **Neural Network and Deep Learning**
- **(Deep) Reinforcement Learning**

整体目录如下：

**I Mathematical Foundations**

- Sets, Sequences and Series
- Metric Space and Topological Space
- Advanced Calculus
- Linear Algebra and Matrix Analysis
- Function Sequences, Series and Approximation
- Basic Functional Analysis

**II Mathematical Optimization Methods**

- Unconstrained Nonlinear Optimization
- Constrained Nonlinear Optimization
- Linear Optimization
- Convex Analysis and Convex Optimization
- Basic Game Theory

**III Classical Statistical Methods**

- Theory of Probability
- Statistical Distributions
- Statistical Estimation Theory
- Multivariate Statistical Methods
- Linear Regression Analysis
- Monte Carlo Methods

**IV Dynamics Modeling Methods**

- Models and estimation in linear dynamical systems
- Stochastic Process
- Stochastic Calculus
- Fokker-Planck Equation
- Markov Chain and Random Walk
- Time Series Analysis

**V Statistical Learning Methods**

- Supervised Learning Principles and Methods
- Linear Models for Regression
- Linear Models for Classification
- Generative Models
- K Nearest Neighbors
- Tree Methods
- Ensemble and Boosting Methods
- Unsupervised Statistical Learning
- Neural Network and Deep Learning

**VI Optimal Control and Reinforcement Learning Methods**

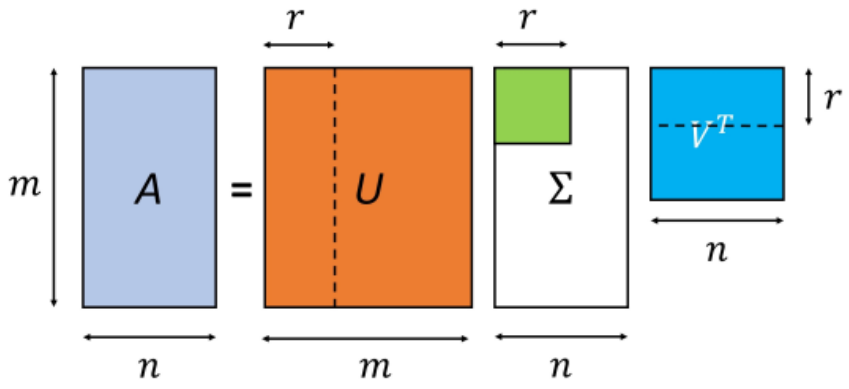
- Classical Optimal Control Theory
- Reinforcement Learning

**Appendix: Supplemental Mathematical Facts**

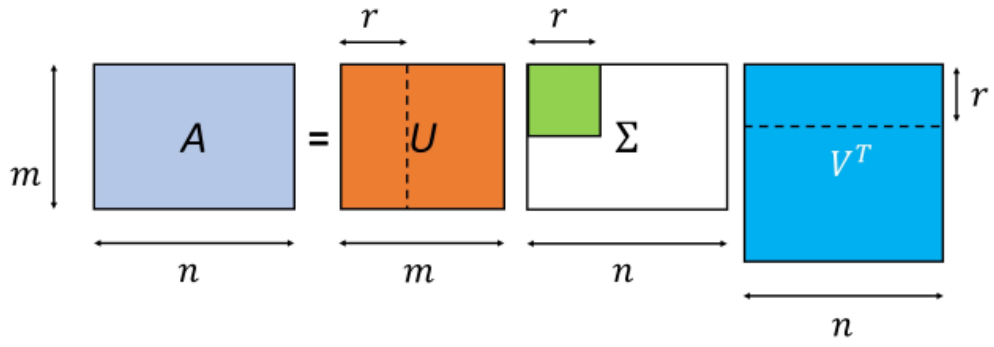
## 内容展示

### 线性代数篇

SVD (矩阵奇异值分解) 是线性代数中最重要工具之一， 经常在各类统计以及重要机器学习方法中出现。作者用如下图示和定理对 SVD 的性质进行总结和证明。该证明简洁扼要，且所用到的其它辅助定理与证明都在本书中 。作者使用一个图示来分别指出 complete form SVD 和 compact form SVD 的结果和原矩阵的关系。




**(a)** Demonstration of SVD for a tall and skinny matrix.




**(b)** Demonstration of SVD for a short and fat matrix.

**Figure 5.9.1:** Demonstration of SVD for matrices of different shapes. The dashed lines highlight the compact form SVD.

**Theorem 5.9.1 (complete form SVD).** *Any matrix $A \in \mathbb{R}^{m\times n}$ has a factorization given by [Figure 5.9.1]*

$$A = U\Sigma V^T$$

*where $U \in \mathbb{R}^{m\times m}, V \in \mathbb{R}^{n\times n}, \Sigma \in \mathbb{R}^{m\times n}$ and $\Sigma$ is rectangle diagonal matrix. The diagonal entries in $\Sigma$, known as **singular values**, consist of first $r = rank(A)$ **non-zero, positive, decreasing entries** $(\sigma_1, ..., \sigma_r)$, and other zeros.*
*Moreover, $\sigma_i^2$ is a eigenvalue of matrix $AA^T$ and $A^TA$; $u_i$ and $v_i$ (columns in $U$ and $V$) are eigenvectors of $AA^T$ and $A^TA$, respectively.*

*Proof.* We use following three steps to prove SVD. (1) Consider the matrix $AA^T$, which is real-valued symmetric and therefore diagonalizable [Theorem 5.8.3]. Let $AA^T = \sum_{i=1}^r \lambda_i u_i u_i^T$, where $\{\lambda_i\}$, $\{u_i\}$ are reversely sorted $r$ non-zero, positive eigenvalues and their eigenvectors of $AA^T$. Note that $A$ and $A^TA$ have the same rank $r$ [Lemma 5.4.3]; therefore, $AA^T$ only has $r$ non-zero eigenvalues. (2) For each $u_i$, construct $v_i = \frac{A^Tu_i}{\sqrt{\lambda_i}}$. Now we show that $v_i$ is a unit eigenvector associated with eigenvalue $\lambda_i$ of $A^TA$. Note that

$$A^TAv_i = \frac{A^TAA^Tu_i}{\sqrt{\lambda_i}} = \frac{A^T\lambda_i u_i}{\sqrt{\lambda_i}} = \lambda_i v_i,$$

and

$$v_i^Tv_i = \frac{u_i^TA^TAu_i}{\lambda_i} = 1.$$

Therefore, we can write $A^TA = \sum_{i=1}^r \lambda_i v_i v_i^T$. (3) Let $U$ consist of columns $u_1, ..., u_r, u_{r+1}, ..., u_m$, where $u_{r+1}, ..., u_m$ are the basis spanning the $\mathcal{N}(A^T)$ (or $\mathcal{N}(A^T)$, which is the same since $\mathcal{N}(A^T) = \mathcal{N}(A^TA)$, Lemma 5.4.3). Similarly, let $V$ consist of columns $v_1, ..., v_r, v_{r+1}, ..., v_n$, where $v_{r+1}, ..., v_n$ are the basis spanning $\mathcal{N}(A)$. Note that $u_i^TAv_j = \delta_{ij}\sqrt{\lambda_i}$, therefore

$$U^TAV = \Sigma \implies A = U\Sigma V^T,$$

where $\Sigma$ is a diagonal matrix with entries of $\sqrt{\lambda_1}, ..., \sqrt{\lambda_r}, 0, ..., 0$. □

作者同时指出新手常混淆的一个知识点：

**Remark 5.9.2** (relationship between $U$ and $V$). It is a common mistake to think that $U$ and $V$ are orthogonal to each other, i.e. $U^TV = I$. Actually, $U$ and $V$ are orthogonal to each other when $A$ is symmetric. Particularly, we have:

- $U$ consists of the eigenvectors of $AA^T$, and $V$ consists of the eigenvectors of $A^TA$.
- If $A$ is not square, $U$ and $V$ cannot even multiply together (incompatible sizes).
- If $A$ is symmetric, columns in $U$ and $V$ are eigenvectors of $A^2$ and $A$. Therefore, $U$ and $V$ are orthogonal to each other.

**统计篇**

多元高斯随机变量( multivariate random variable) 的 affine transformation 经常被用于证明高斯随机变量的一系列重要性质（比如加和， 条件等）。本书首先给出用矩函数对此定理的证明。

**Theorem 13.1.1 (affine transformation for multivariate normal distribution).** *Let $X$ be an $n$-dimensional random vector with $MN(\mu, \Sigma)$ distribution. Let $Y = AX + b, A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$. Then $Y$ is an $m$-dimensional random vector having a $MN(A\mu + b, A\Sigma A^T)$ distribution.*

*Proof.* Use moment generating function to prove. Let $Y = AX + b$, then from Lemma 12.7.5

$$M_Y(t) = e^{t^T b} M_X(A^T t) = e^{t^T(A\mu+b)+\frac{1}{2}t^T A\Sigma A^T t}$$

which indicates $Y \sim MN(A\mu + b, A\Sigma A^T)$. □

然后本书给出此定理在多元高斯随机变量加和中的应用。值得一提的是， 作者用脚注强调 jointly normal 这一重要条件。

**Corollary 13.1.1.1 (sum of two multivariate normal random vectors).** *Let $X_1 \sim MN(\mu_1, \Sigma_1)$ and $X_2 \sim MN(\mu_1, \Sigma_1)$ be two $n$ dimensional multivariate normal random variable. It follows that*

- *If $X_1$ and $X_2$ are independent, then $Y = X_1 + X_2$ is a multivariate normal random vector with $MN(\mu_1 + \mu_2, \Sigma_1 + \Sigma_2)$.*
- *If $X_1$ and $X_2$ are dependent and $(X_1, X_2)$ are **jointly normal** [a] with covariance matrix given by*

$$\Sigma = \begin{bmatrix} \Sigma_1 & \Sigma_{12} \\ \Sigma_{12} & \Sigma_2 \end{bmatrix},$$

*then $Y = X_1 + X_2$ is a multivariate normal random vector with $MN(\mu_1 + \mu_2, \Sigma_1 + \Sigma_2 + 2\Sigma_{12})$.*

a If $X_1$ and $X_2$ are not jointly normal, $Y$ is not normal.

*Proof.* (1) Consider a $2n$-dimensional multivariate normal random variable $Z$ with distri-

**机器学习篇**

在机器学习的线性分类模型中，三种常见模型 SVM, logistic regression 和 Perceptron learning 可以统一在同一个数学优化框架下，每种方法对应不同的 loss function。作者对如何把这三种模型转化成同一个框架进行了详细的阐述和证明。

### 26.5.5 A unified perspective from loss functions

In this section, we show that Logistic regression, Perceptron learning, and SVM for binary classification problem are unified under the same optimization problem given by

$$\min_{\beta_0,\beta} \sum_{i=1}^{N} L(y_i, f(x)) + \lambda \beta^T \beta,$$

where $L$ is the loss function taking different forms, respectively, and $f(x) = \beta^T x + \beta_0$.

The formulation of SVM into this framework relies on the following Lemma.

**Lemma 26.5.2 (equivalent form of soft margin SVM).** *Let the binary classification training data consist of $N$ pairs $(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$, with $x_i \in \mathbb{R}^p, y_i \in \{1, -1\}$. The **soft margin SVM classification** optimization*

$$\min_{\beta,\beta_0,\eta} \|\beta\|^2 + C \sum_{i=1}^{N} \eta_i$$

*under the constraints:*

$$y_i(x_i^T \beta + \beta_0) \geq 1 - \eta_i, i = 1, 2, ..., N$$
$$\eta_i \geq 0, i = 1, ..., N$$

*where the $C$ is the regulation parameter, is equivalent to*

$$\min_{\beta_0,\beta} \sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda \|\beta\|^2,$$

*where*

$$L(y_i, f(x_i)) = \max(0, 1 - y_i f(x_i)).$$

**强化学习篇**

Value iteration 值迭代是强化学习的基石型定理之一，然而目前很多教材资料中并没有给出证明。本书通过 contraction mapping 和 fixed point theorem 得出简明的证明。contraction mapping 和 fixed point theorem 的知识点则在在本书 Part I 有详细介绍。

**Theorem 34.1.3 (convergence property of value iteration).** *For a finite state MDP, we can write the optimal value function recursive relationship as*

$$V^*(s) = \max_a \sum_{s' \in \mathcal{S}} P(s'|s,a)[R(s',a) + \gamma V^*(s')], \forall s \in \mathcal{S}.$$

*We can express the recursive relationship as a matrix form given by*

$$V = T(R + \gamma V),$$

*where* $R, V \in \mathbb{R}^{|\mathcal{S}|}, T \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$.

*We further define* $H(V) \triangleq \max_a T(R + \gamma V)$ *as the value iteration operator.*

*We have*

- *H is a* ***contraction mapping****.*
- *In iterative policy evaluation,* $V^k(s)$ *will converge to the unique optimal value function* $V^*$*. Or equivalently,* $V^*$ *is the* ***fixed point*** *of the contraction mapping* $H$*, and*

$$\lim_{n \to \infty} H^n(V) = V^*.$$

- *(error bound) If* $\left\| H^k(V) - H^{k-1}(V) \right\|_\infty \leq \epsilon$*, then*

$$\left\| H^k(V) - V^* \right\|_\infty \leq \frac{\epsilon}{1 - \gamma}.$$

之后作者给出基于 value iteration 的算法。

**Algorithm 70:** Value iteration algorithm for a finite state MDP

**Input:** MDP, small positive number $\epsilon$ as tolerance

**Output:** Value function $V \approx V^*$ and policy $\pi \approx \pi^*$.

Initialize $V$ arbitrarily. Set ($V(s) = 0$ for all $s \in \mathcal{S}^T$.)

**repeat**

  $\Delta = 0$

  **for** $s \in \mathcal{S}$ **do**

    $v = V(s)$

    $V(s) = \max_{a \in \mathcal{A}(s)} \sum_{s' \in \mathcal{S}} P(s'|s,a)(R(s',a) + \gamma V(s'))$

    $\Delta = \max(\Delta, |v - V(s)|)$

  **end**

**until** $\Delta < \epsilon$;

Compute the policy $\pi(s) = \max_{a \in \mathcal{A}(s)} \sum_{s' \in \mathcal{S}} P(s'|s,a)(R(s',a) + \gamma V(s'))$.

**return** *V and* $\pi$