第二部分 无监督学习 / Unsupervised Learning

翻译&校正 | 韩信子@ShowMeAI　编辑 | 南乔@ShowMeAI　原文作者 | https://stanford.edu/~shervine

本节原文超链

[1]无监督学习简介/ Introduction to Unsupervised Learning

动机 Motivation

无监督学习的目标是，通过对无标签数据集$\{x^{(1)}, \ldots, x^{(m)}\}$的学习，揭示数据的内在分布特性及规律。

琴生不等式 Jensen's inequality

对凸函数 f 和随机变量 X，以下不等式成立：

$$E[f(X)] \geq f(E[X])$$

[2]聚类 / Clustering

2.1 E-M 算法 / Expectation-Maximization

隐变量 Latent variables

隐变量不可观测的特性，为估测增加了难度。隐变量写作 z。以下是隐变量常见设定：

设定	隐变量z	$x\|z$	评论
k元混合高斯分布	$\text{Multinomial}(\phi)$	$\mathcal{N}(\mu_j, \Sigma_j)$	$\mu_j \in \mathbb{R}^n, \phi \in \mathbb{R}^k$
因子分析	$\mathcal{N}(0, I)$	$\mathcal{N}(\mu + \Lambda z, \psi)$	$\mu_j \in \mathbb{R}^n$

算法 Algorithm

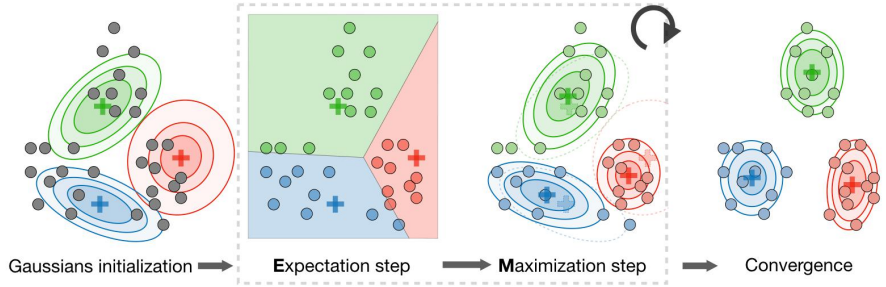
E-M 算法（**Expectation-Maximization Algorithm**）能够高效地估计参数 θ ——通过重复构建似然函数的下界（**E-步**）和最优化下界（**M-步**）进行极大似然估计：

E-步：计算后验概率 $Q_i(z^{(i)})$，其中每个数据点 $x^{(i)}$ 来自特定的簇 $z^{(i)}$，过程：

$$Q_i(z^{(i)}) = P(z^{(i)}|x^{(i)};\theta)$$

M-步：使用后验概率 $Q_i(z^{(i)})$ 作为簇在数据点 $x^{(i)}$上的特定权重来分别重新估计每个簇模型，过程：

$$\theta_i = \underset{\theta}{\text{argmax}} \sum_i \int_{z^{(i)}} Q_i(z^{(i)}) \log\left(\frac{P(x^{(i)}, z^{(i)};\theta)}{Q_i(z^{(i)})}\right) dz^{(i)}$$



备注：Gaussians initialization[高斯初始化] → E 步 → M 步 → Convergence[收敛]。

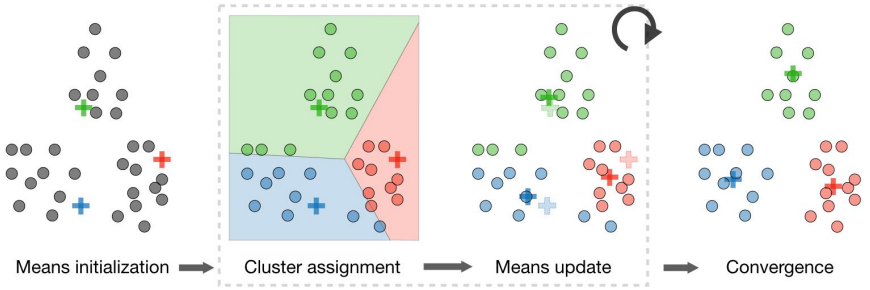
2.2 k-均值聚类 / k-means Clustering

记 $c^{(i)}$ 为数据点 i 的簇，μ_j 是簇 j 的中心。

算法 Algorithm

在随机初始化簇中心 $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$后，k-均值算法重复下列步骤直至收敛：

$$c^{(i)} = \arg\min_j \|x^{(i)} - \mu_j\|^2 \quad \text{和} \quad \mu_j = \frac{\sum_{i=1}^m 1_{\{c^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^m 1_{\{c^{(i)}=j\}}}$$



备注：Means initialization[初始化中心] → Cluster assignment[分类聚类类别] → Means update[更新中心] → Convergence[收敛]。

失真函数 Distortion function

为了看到算法是否收敛，失真函数定义如下：

$$J(c,\mu)=\sum_{i=1}^{m}\left\|x^{(i)}-\mu_{c^{(i)}}\right\|^2$$

2.3 层次聚类 / Hierarchical Clustering

算法 Algorithm

层次聚类也是聚类算法，采用自底向上逐步聚合的方法，构建嵌套的层次化聚类结果。

类型 Types

不同类型的层次聚类算法，用以优化不同的目标函数优化问题，总结如下表：

内链	均链	全链
最小化簇内距离	最小化簇对平均距离	最小化簇对最大距离

2.4 聚类评估指标 / Clustering Assessment Metrics

与监督学习相比，无监督学习中的模型性能通常难以评估，因为无监督学习没有标准答案（ ground truth labels ）。

轮廓系数 Silhouette coefficient

a 为某一样本与同一簇中其他所有点的平均距离，b 为此样本与最近簇中其他所有点的平均距离。则该样本的轮廓系数 s （ Silhouette coefficient ）定义为：

$$s=\frac{b-a}{\max(a,b)}$$

CH 指标 Calinski-Harabaz index

k 为簇的数目。B_k 为簇间弥散矩阵， W_k 为簇内弥散矩阵，定义如下：

$$B_k=\sum_{j=1}^{k}n_{c^{(i)}}\left(\mu_{c^{(i)}}-\mu\right)\left(\mu_{c^{(i)}}-\mu\right)^T$$

$$W_k=\sum_{i=1}^{m}\left(x^{(i)}-\mu_{c^{(i)}}\right)\left(x^{(i)}-\mu_{c^{(i)}}\right)^T$$

CH 指标（ Calinski-Harabazindex ），表示一个聚类模型对簇的定义程度。指标得分越高，表示簇越稠密且分隔性能越好。记作 s(k) ，表示如下：

$$s(k)=\frac{\mathrm{Tr}(B_k)}{\mathrm{Tr}(W_k)}\times\frac{N-k}{k-1}$$

[3]降维 / Dimension Reduction

3.1 主成分分析/ PCA

是一种降维技术，可以找到方差最大化的方向，并将数据投影到该方向上。

特征值 & 特征向量 Eigenvalue, eigenvector

给定矩阵 $A\in\mathbb{R}^{n\times n}$。若存在特征向量 $z\in\mathbb{R}^n\setminus\{0\}$ 满足下方公式，则λ 为矩阵 A 的一个特征值。

$$Az=\lambda z$$

谱定理 Spectral theorem

给定矩阵 A $\in\mathbb{R}^{n\times n}$。如果 A 是对称阵，那么 A 可以被一个实正交矩阵 U $\in\mathbb{R}^{n\times n}$对角化。记$\Lambda$ = diag$(\lambda_1,\ldots,\lambda_n)$，则有：

$$\exists\Lambda\text{为对角矩阵，}\quad A=U\Lambda U^T$$

备注：与最大特征值对应的特征向量，被称为矩阵 A 的主特征向量。

算法 Algorithm

主成分分析（**Principal Component Analysis**，**PCA**）的是一个降维技术，通过最大化数据方差，将数据投影到 **k** 维上：

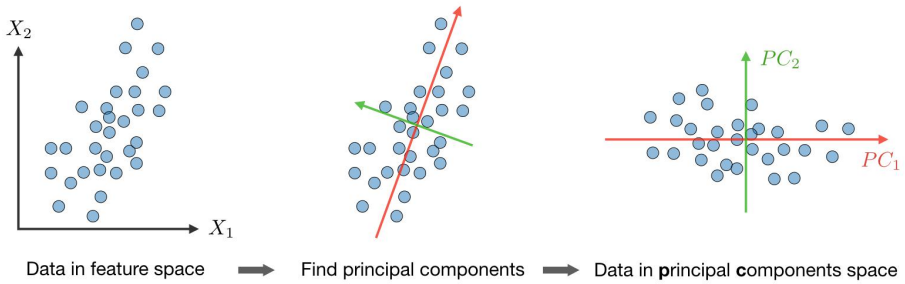
步骤 1：数据标准化，使均值为 **0**，标准差为 **1**。

$$x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{\sigma_j} \quad \text{其中} \quad \mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)} \ , \ \sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)} - \mu_j\right)^2$$

步骤 2：计算 $\Sigma = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} x^{(i)T} \in \mathbb{R}^{n\times n}$，其为有实特征值的对称阵。

步骤 3：计算 Σ 的 **k** 个正交主特征向量 $u_1, \ldots, u_k \in \mathbb{R}^n$，即 **k** 个最大特征值对应的正交特征向量。

步骤 4：将数据投影到 $\text{span}_{\mathbb{R}}(u_1, \ldots, u_k)$ 上。在此过程中，将所有 **k** 维空间的方差最大化。



备注：**Data in feature space**[特征空间的数据] → **Find principal components**[寻找主成分] → **Data in principal components space**[主成分空间的数据]。

3.2 独立成分分析 / Independent Component Analysis

这是一种寻找数据背后统计独立的信号源组合的技术。

假设 Assumptions

$s = (s_1, \ldots, s_n)$ 为 n-维源向量，s_i 为独立随机变量。A 为混合和非奇异矩阵**[mixing and non-singular matrix]**。数据 x 由以下方式产生：

$$x = As$$

目标是要找到分离矩阵 $W = A^{-1}$。

ICA 算法 Bell and Sejnowski ICA algorithm

该算法通过下列步骤，找到分离矩阵 **W**，：

1）$x = As = W^{-1}s$ 的概率为：

$$p(x) = \prod_{i=1}^{n} p_s\left(w_i^T x\right) \cdot |W|$$

2）训练数据为 $\{x^{(i)}, i \in [\![1, m]\!]\}$，**sigmoid** 函数为 **g**，对数似然函数如下：

$$l(W) = \sum_{i=1}^{m}\left(\sum_{j=1}^{n} \log\left(g'\left(w_j^T x^{(i)}\right)\right) + \log|W|\right)$$

因此，随机梯度下降学习规则是，对每个训练样本 $x^{(i)}$，按照下述方式更新 **W**：

$$W \leftarrow W + \alpha\left(\begin{pmatrix} 1 - 2g\left(w_1^T x^{(i)}\right) \\ 1 - 2g\left(w_2^T x^{(i)}\right) \\ \vdots \\ 1 - 2g\left(w_n^T x^{(i)}\right) \end{pmatrix} x^{(i)T} + (W^T)^{-1}\right)$$

Awesome AI Courses Notes Cheat Sheets

Machine Learning CS229	Deep Learning CS230	Natural Language Processing CS224n	Computer Vision CS231n	Deep Reinforcement Learning CS285	Neural Networks for NLP CS11-747	DL for Self-Driving Cars 6.S094	...
Stanford	Stanford	Stanford	Stanford	UC Berkeley	CMU	MIT	...

是 **ShowMeAI** 资料库的分支系列，覆盖最具知名度的 TOP20+门 **AI** 课程，旨在为读者和学习者提供一整套高品质中文速查表，可以点击【这里】查看。

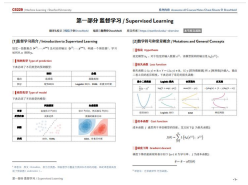
斯坦福大学（**Stanford University**）的 **Machine Learning**（**CS229**）和 **Deep Learning**（**CS230**）课程，是本系列的第一批产出。

本批两门课程的速查表由斯坦福大学计算机专业学生 **Shervine Amidi** 总结整理。原速查表为英文，可点击【这里】查看，**ShowMeAI** 对内容进行了翻译、校对与编辑排版，整理为当前的中文版本。

有任何建议和反馈，也欢迎通过下方渠道和我们联络 (*￣3￣)

CS229 | Machine Learning @ Stanford University

监督学习
Supervised Learning



中文速查表链接

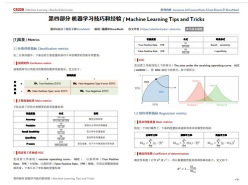
无监督学习
Unsupervised Learning

中文速查表链接

深度学习
Deep Learning

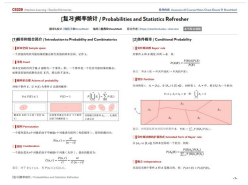
中文速查表链接

机器学习技巧和经验
Tips and Tricks



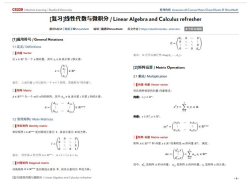
中文速查表链接

概率统计
Probabilities /Statistics



中文速查表链接

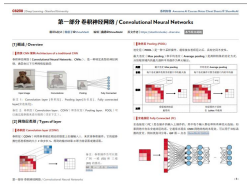
线性代数与微积分
Linear Algebra and Calculus



中文速查表链接

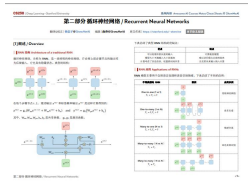
CS230 | Deep Learning @ Stanford University

卷积神经网络
CNN



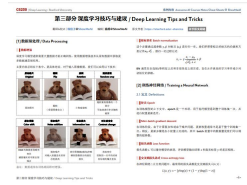
中文速查表链接

循环神经网络
RNN



中文速查表链接

深度学习技巧与建议
Tips and Tricks



中文速查表链接

GitHub

ShowMeAI

https://github.com/ShowMeAI-Hub/

ShowMeAI 研究中心

扫码回复"**速查表**"

下载**最新**全套资料