

# 人工智能现代方法：机器学习 🤖

- [Intro. \(一些琐碎的东西\)](#)
- [ML Basics \(一份漫游指南 & 一份缩略图\)](#)
- [Linear Models](#)
- [Kernel Methods & SVM](#)
- [Bayesian Classification & Probabilistic Graphical Models](#)
- [Ensemble Learning](#)
- [前馈神经网络](#)
- [卷积神经网络](#)
- [循环神经网络](#)
- [无监督学习与聚类](#)
- [采样方法](#)

## Intro. (一些琐碎的东西)

### 算法主要考查简述

用机器学习把之前学过的知识串起来

记忆、概念、推导、计算（带计算机）、理解类

不考填空

SVM的对偶问题（优化）

SVM用于回归

维特比算法

会对概念定义进行考察

会要求理解性阐释

正文中出现的所有 $\log$ 特指自然对数 $\ln$

## ML Basics (一份漫游指南 & 一份缩略图 🗺)

学好《机器学习》这门课，与其说是学会使用一种工具，倒不如说是将一种思想内化于心——机器学习的思想。

我被告知，这些思想并非凭空产生，亦非前辈们“一拍脑门”想出的，更不是带有强烈主观色彩的人为规定。因此，这带给我们一个好消息：要真正领会这些思想，并知晓它们从何而来，我们还得从它们的数学根基入手（这是一个相当简明的抓手）。

这是因为，**数学**给予我们形式化定义一门语言的能力，而机器学习正是这样一门语言，它充当着沟通数据科学与人工智能“两岸”的桥梁。

ps.第一章主要考查选择题

## 什么是机器学习

### 三大要素：T、P、E：

对于某种**任务T**、**性能度量P**，一个及其程序被认为可以从**经验E**中学习指：利用经验E，它在任务T上由性能度量P衡量的性能提升。

- T：智能系统执行的、实现目标的工作或智能系统处理一个样本（对象中已量化特征）的工作。  
e.g.分类、输入缺失分类、回归、转录、翻译、结构输出（输出变量之间关系）、异常检测、合成和采样、缺失值填补、去噪、密度估计
- P：性能度量用来描述机器学习算法能力，与任务相关，不一定能精确定义其性能度量。  
e.g.分类正确率，概率估计，
- E：经验是人们知识积累、经验就是数据集，数据点的集合

**两大流程：学习（训练）、推理**

# ML vs. DL

深度学习的“五宗罪”（误）：

- 深度学习并不能对**所有领域**的问题都取得好的效果
- 深度学习方法的实现**依赖**机器学习方法
- 深度学习存在**鲁棒性低、可解释性差、处理不完全信息能力弱**的缺陷
- 深度学习依赖**大量数据和计算资源**、计算量大、成本高昂
- 传统机器学习方法的**思想**对未来研究具有**启发**意义

## 前置知识 & Toolbox

- 高等数学
- 线性代数
- 概率论
- 矩阵论
- 计算方法

## 矩阵与优化

- 解析解（求矩阵方程）

关于矩阵求导的快速技巧：

- 近似解（梯度下降法/牛顿迭代法）

第三章线性模型中的梯度的求解与迭代公式中的最后一项转置问题存在前后不一致的问题

详见[线性模型用于回归问题](#)

## 数据的维度（tensorflow中的tensor）

高维数据的低维可视化方案

## 数据的性态

核心是用[距离/范数](#)表征的相似度(similarity)

距离也是[聚类](#)问题中的一个关键设定

## 一些不那么常见的距离

马氏距离

$$D_M = \sqrt{(\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y})}$$

$\Sigma$  : covariance matrix

Tanimoto测度

$$T(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T \mathbf{y}}{\mathbf{x}^T \mathbf{x} + \mathbf{y}^T \mathbf{y} - \mathbf{x}^T \mathbf{y}}$$

## 概率与信息

### 我们为什么需要引入概率

分类问题本质上是求解各目标类的概率分布

### 概率论，你熟练掌握了吗？（基于概率模型的一点补充）

Notation:

PDF 概率密度函数

PMF 概率质量函数  $\xrightarrow{\text{multiple variables}}$  JPD 联合概率分布

CDF 累积分布函数

先祭出来正态分布的公式：

若 $x \sim N(\mu, \sigma^2)$ , 则 $x$ 的PDF表示为 $\frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{(x-\mu)^2}{2\sigma^2})$

再祭出来柯西分布的公式：  
(柯西分布具有"长尾"效应)

再祭出来Gamma分布的公式：

**关于（条件）独立性的一点强调**

$$\text{If } p(X = x, Y = y|Z = z) = p(X = x|Z = z)p(Y = y|Z = z),$$
  
$$\text{then } X \text{ and } Y \text{ are conditional independent about } Z, \text{ denoted by } X \perp Y|Z$$

**从贝叶斯公式到贝叶斯模型**

$p(y)$ : 先验概率——样本空间中各类样本的先验分布

$p(y|x)$ : 后验概率——所求目标的概率分布

$p(x|y)$ : 似然函数——样本特征属性的联合概率

详见[贝叶斯模型](#)

以下内容重在数学推导与求解：

**MLE：最大似然估计**

$$\theta_{MLE} = \operatorname{argmax}_{\theta} p(x|\theta)$$

MLE 选择使得观测数据在给定模型下最有可能发生的参数值

MLE 是 MAP 的一个特例。**当先验分布  $p(\theta)$  是均匀分布时，MAP 估计等同于 MLE。**

**CMLE：条件最大似然估计**

$$\theta_{CMLE} = \operatorname{argmax}_{\theta} p(x|y, \theta)$$

CMLE 主要应用于涉及条件分布的情景下，如条件概率图模型等

**在无条件信息时，CMLE 退化为 MLE。**

**MAE：最大后验估计**

$$\theta_{MAP} = \operatorname{argmax}_{\theta} p(x|\theta)p(\theta)$$

最大后验估计结合了**似然函数和先验分布**，通过贝叶斯定理，寻找最大化后验概率的参数值。MAP 估计考虑了先验知识，有助于在**数据稀缺或噪声较大**的情况下提高估计的稳定性。

**信息论（一门学科的入门水平）**

在机器学习中用来描述概率分布或量化概率分布之间的相似性

信息熵是信息量的期望，就是平均而言发生一个事件我们得到信息量的大小

**交叉熵：定义于两个概率分布之上，反映两个概率分布的差异程度。**

交叉熵是用来衡量在给定的**真实分布** $p(x)$ 下，使用**非真实分布** $q(x)$ 所指定的策略消除系统的不确定性所需要付出的努力的大小。即 $q(x)$ 是对 $p(x)$ 的预测(估计)值，有时也被称为 $\hat{p}(x)$

$$H(p, q) = - \sum_x p(x) \ln q(x)$$

- 非对称

**机器学习时候要拟合分布，所以可以用交叉熵构造损失函数。**

**kullback leibler(KL)散度/相对熵/信息增益：用于衡量两个概率分布之间的差距。**

相对熵 = 某个策略的交叉熵（ $p \rightarrow q$ 付出努力大小） - 信息熵( $p$ 平均信息量)

**信息增益**

交叉熵是相对熵的一种特殊情况,即 $p(x)$ 分布是已知的,因而导致公式的后半部分为常数项。

$$D_{KL}(p||q) = \sum_x p(x) \ln \frac{p(x)}{q(x)}$$

- 非对称

- 相对熵 = 交叉熵 - 自信息
- 非负性

JS散度

$$D_{JS}(p||q) = \frac{1}{2}D_{KL}(p||m) + \frac{1}{2}D_{KL}(q||m)$$
$$m(x) = \frac{1}{2}[p(x) + q(x)]$$

- 对称

讲ML都会讲的一些东西

过/欠拟合分析

拟合现象	原因
过拟合	1. 模型过于复杂 <b>(模型的容量：拟合函数的能力)</b> 2. 训练样本太少 3. 样本噪声太大
欠拟合	1. 模型过于简单 2. 特征数太少

正则化方法

$$J(\Theta) = L(\Theta; X, Y) + \frac{\lambda}{2}\Theta^T\Theta$$

$\lambda$ : regularization factor  $\Theta$ : parameter

评价指标

P-R曲线

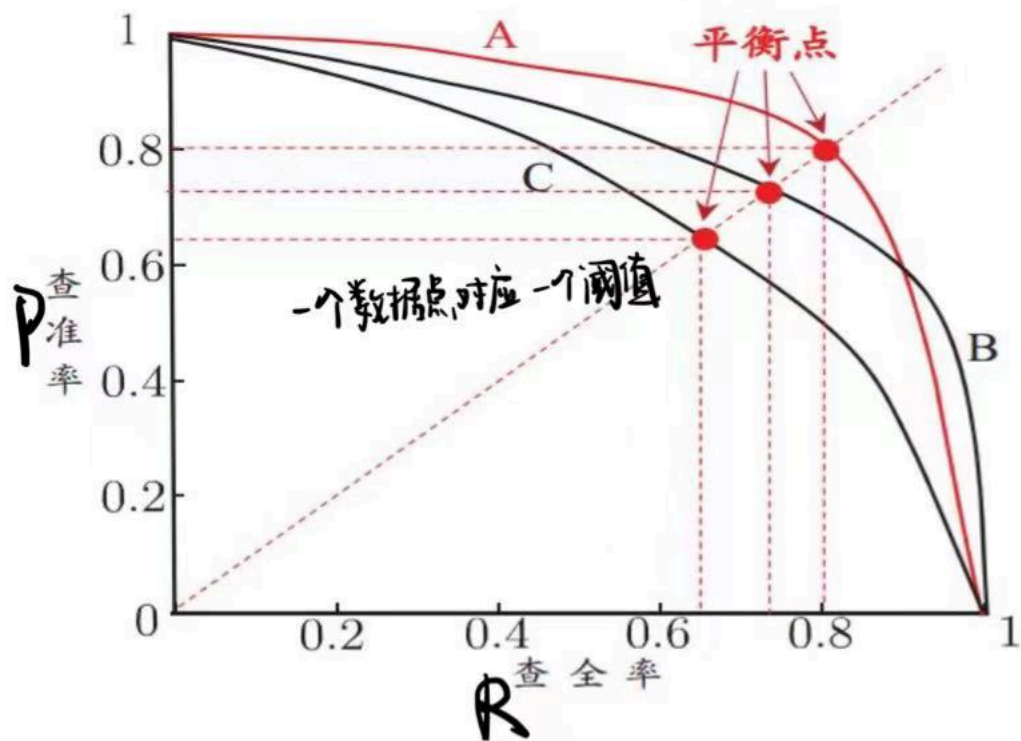
P is for Precision（查准率）：你认为对的样本，有多少是对的  
R is for Recall（查全率）：所有对的样本，你找出来了多少

P & R：反相关；我们希望二者都保持在较高的水平  
因此，取 P = R 作为BEP（平衡点），作为综合考虑二者的依据

$F_\beta - Score$ ：  
 $\beta$ 用于（加权）调和P和R对于整体的贡献程度

$$\frac{1}{F_\beta} = \frac{1}{1 + \beta^2}(\frac{1}{P} + \frac{\beta^2}{R})$$

出每次的P、R (曲线围络面积反应全面性能)



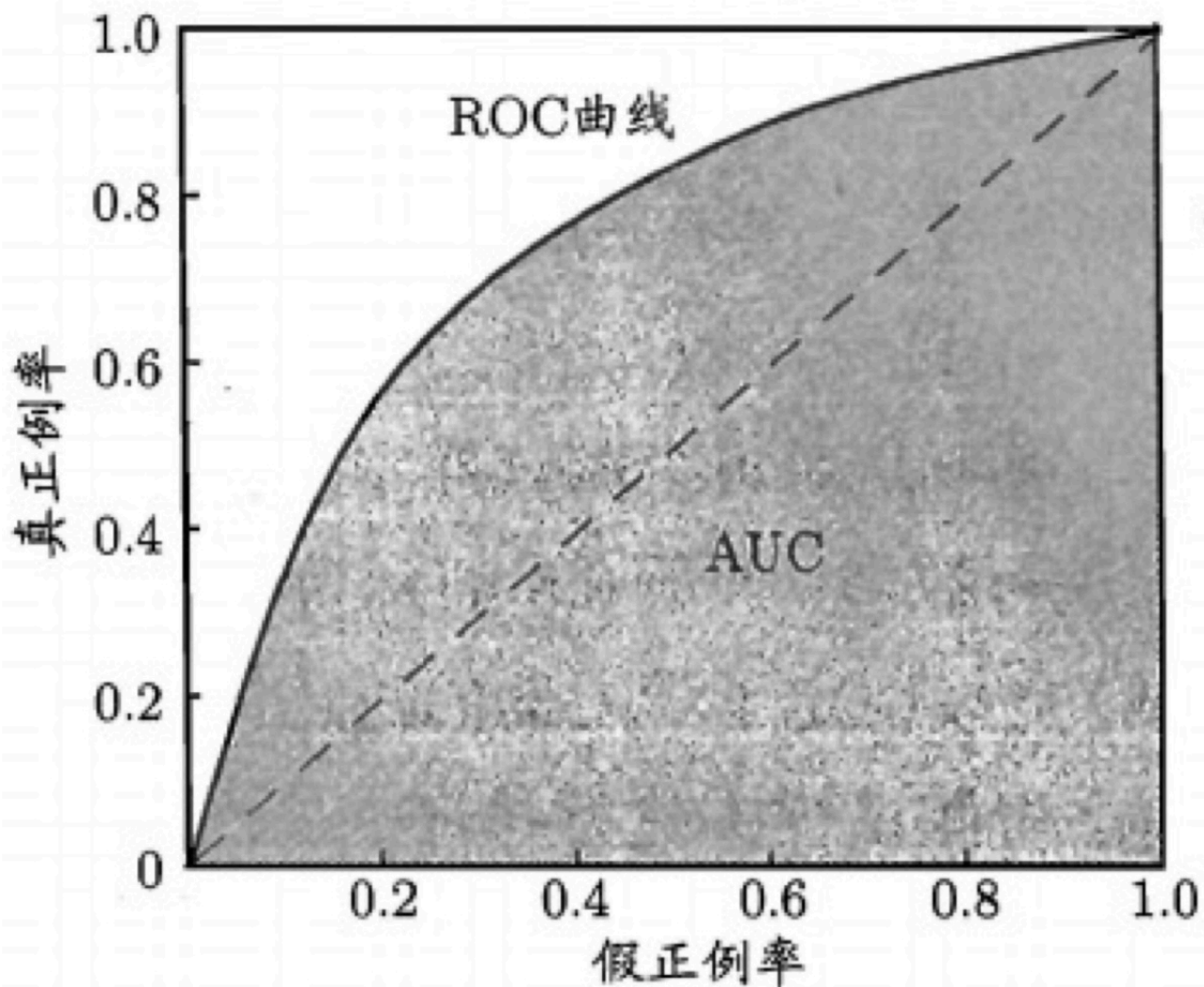
59

#### ROC (受试者工作特征曲线)

TPR is for True Positive Rate (真正例率) : 与查全率R相等!!!

FPR is for False Positive Rate (假正例率) : 所有反例中被你错误地预测为正例的比例

TPR & FPR: 正相关; 我们希望TPR维持在较高水平而FPR维持在较低水平



共同点：

1. 曲线与坐标轴间的包络面积反映模型的整体性能(AUC)
2. 动态调整阈值以控制对于曲线上每一坐标点的plot

不同点：

1. P-R曲线更适用于不平衡数据集；ROC更适用于平衡数据集

### 评价方法（技巧）

#### 交叉验证：k-fold

- 样本均分为k份
- 取(k-1)份训练，1份测试
- k批次并行训练
- 结果取性能平均值

#### 自助采样法

### 关于ML的一些分类

#### 不同分类标准

#### 有监督学习算法的分类

#### 判别模型

对于判别式模型来说求得 $P(Y|X)$ ，对未见示例 $X$ ，根据 $P(Y|X)$ 可以求得标记 $Y$ ，即可以直接判别出来，如上图的左边所示，**实际是就是直接得到了判别边界**，所以传统的、耳熟能详的机器学习算法如线性回归模型、支持向量机SVM等都是判别式模型，这些模型的特点都是输入属性 $X$ 可以直接得到 $Y$ （对于二分类任务来说，实际得到一个score，当score大于threshold时则为正类，否则为反类）~（根本原因个人认为是对于某示例 $X_1$ ，对正例和反例的标记的条件概率之和等于1，即 $P(Y_1|X_1)+P(Y_2|X_1)=1$ ）

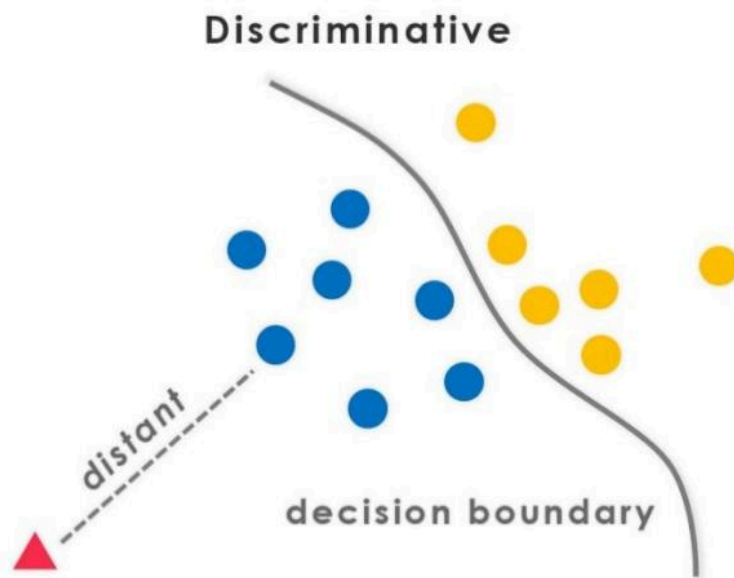
### 生成模型

而生成式模型求得 $P(Y,X)$ ，对于未见示例 $X$ ，你要求出 $X$ 与不同标记之间的联合概率分布，然后大的获胜，如上图右边所示，并没有什么边界存在，对于未见示例（红三角），求两个联合概率分布（有两个类），比较一下，取那个大的。机器学习中朴素贝叶斯模型、隐马尔可夫模型HMM等都是生成式模型，熟悉Naive Bayes的都知道，对于输入 $X$ ，需要求出好几个联合概率，然后较大的那个就是预测结果~（根本原因个人认为是对于某示例 $X_1$ ，对正例和反例的标记的联合概率不等于1，即 $P(Y_1,X_1)+P(Y_2,X_1)<1$ ，要遍历所有的 $X$ 和 $Y$ 的联合概率求和，即 $\sum(P(X,Y))=1$ ）

### Sample

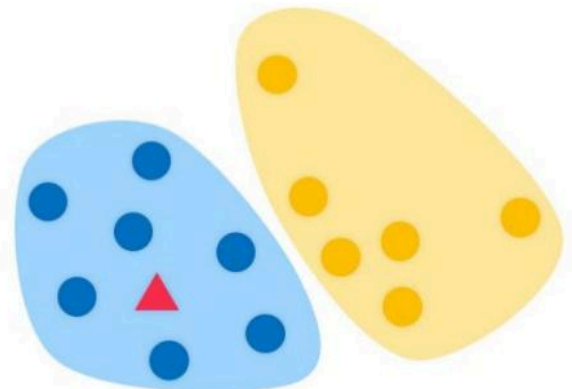
判别式模型举例：要确定一个羊是**山羊** 🐐 还是**绵羊** 🐑，用判别模型的方法是从历史数据中学习模型，然后通过提取这只羊的特征来预测出这只羊是山羊的概率，是绵羊的概率。生成式模型举例：利用生成模型是根据山羊的特征首先学习出一个山羊的模型，然后根据绵羊的特征学习出一个绵羊的模型，然后从这只羊中提取特征，放到山羊模型中看概率是多少，在放到绵羊模型中看概率是多少，哪个大就是哪个。细细品味上面的例子，**判别式模型是根据一只羊的特征可以直接给出这只羊的概率（比如logistic regression，这概率大于0.5时则为正例，否则为反例），而生成式模型是要都试一试，最大的概率的那个就是最后结果~**

# Discriminative vs. Generative



- Only care about estimating the conditional probabilities
- Very good when underlying distribution of data is really complicated (e.g. texts, images, movies)

## Generative



- Model observations  $(x,y)$  first, then infer  $p(y|x)$
- Good for missing variables, better diagnostics
- Easy to add prior knowledge about data

### 要求掌握一些具体例子

Discriminative:

决策/回归等经典算法  
神经网络  
KNN  
条件随机场

Generative:

无监督学习算法的分类

- 聚类
- 数据降维（e.g.PCA）

弱监督学习分类

1. 不完全监督（部分数据标签缺失）
2. 不确切监督（标签粗粒度，不够精细）
3. 不精确监督（部分标签错误，指鹿为马）

其他学习方式

1. 跨域学习 ≈ 迁移学习
2. 序贯学习 ≈ 在线学习
3. 青年大学习（误

按照机器学习任务进行分类

1. 回归任务  
（股价预测、位置估计）
2. 分类任务  
（目标识别、疾病诊断）
3. 转录任务  
（语音识别、字符识别）
4. 机器翻译  
（汉译英、英译汉）
5. 结构输出  
（语句解析、图像内容解析、场景综合理解）
6. 异常检测  
（产品缺陷检测、信用卡欺诈检测）
7. 合成和采样  
（图像生成、数据合成）
8. 缺失值填补  
（深度补全、图像修复）
9. 去噪  
（图片编辑软件：美图秀秀）
10. 密度估计  
（）

Linear Models

参数的存在形式为线性的模型

允许引入**非线性基函数**对输入**x**进行变换

区分线性模型与线性函数这组概念

基函数的作用

- 提高模型的表征能力
- 对原始数据进行某种特征提取 / 特征变换

基函数	形式
恒等基函数	$\phi(x) = x$
幂基函数	$\phi_j(x) = x^j$



基函数	形式
高斯基函数	$\phi_j(x) = \exp\{-\frac{(x-\mu_j)^2}{2s^2}\}$
反曲基函数	$\phi_j(x) = \frac{1}{1+\exp(-\frac{x-\mu}{s})}$

## 线性模型用于回归问题

- 模型的求解

### 解析法

利用矩阵进行最大似然估计

- 似然函数为  $p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^N \mathcal{N}(t_n|\mathbf{w}^T \phi(\mathbf{x}_n), \beta^{-1})$
- 参数的最大似然估计为  $(\mathbf{w}, \beta)_{ML} = \mathit{argmax}_{\mathbf{w}, \beta} \ln p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta)$
- 对数似然函数为  $\frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta E_D(\mathbf{w})$   $E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2$
- 对w求导得  $= \frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\} \phi(\mathbf{x}_n)$
- 令导数为0得到  $\mathbf{w}_{ML} = (\phi^T \phi)^{-1} \phi^T \mathbf{t}$

ps. 平方和误差函数的构造的一致性

### 迭代法

随机梯度下降法（SGD法）

递推公式

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E(\mathbf{w})$$

$E$ 为误差函数

## 拓展：多输出

两种思路：

- 不同基函数进行独立单输出回归
- 联合回归模型 + 多维高斯模型

如果输出之间相对独立，或者每个输出的数据特性差异较大，第一种方法可能更合适。而如果输出之间存在较强的相关性，或者希望模型具有较高的计算效率，第二种方法则可能更优。

## 线性模型用于分类问题

不同之处在于引入**判别函数**进行后处理

## 判别函数的作用：划定决策边界

## 判别函数（学习）方法（K类判别式法）

### 多类别分类

one-versus-the-rest	one-versus-one
$k-1$	$k(k-1)/2$ 【“二维遍历”（单循环赛）】

多分类SVM

### 最小平方和误差

令导数为0得到  $\widetilde{\mathbf{W}} = (\widetilde{\mathbf{X}}^T \widetilde{\mathbf{X}})^{-1} \widetilde{\mathbf{X}}^T \mathbf{T}$

求导优化结果类似

### Fisher线性判别式

引入三个概念：

1. 向量均值
2. 均值投影
3. 类内方差

**Fisher准则：** $J(\mathbf{w})$ 越大越好

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} = \frac{\text{类间距}}{\text{类内距}}$$

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$$

$$\mathbf{S}_W = \sum_{n \in C_1} (x_n - \mathbf{m}_1)(x_n - \mathbf{m}_1)^T + \sum_{n \in C_2} (x_n - \mathbf{m}_2)(x_n - \mathbf{m}_2)^T$$

**Perceptron Algorithm**

**感知机准则：** $E_p(\mathbf{w})$ 越小越好

$\mathcal{M}$ 为错分样本集合

$$E_p(\mathbf{w}) = - \sum_{n \in \mathcal{M}} \mathbf{w}^T \phi(x_n) t_n$$

## 概率模型的引入

概率模型按照x与y的分布划分为：判别式模型（给定x条件下y的概率）和生成式模型（估计x和y的联合分布）

## 逻辑回归模型

1. 解析法（最大似然估计法）  
负对数似然： $E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = -\sum_{n=1}^N \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}$  越小越好
2. 迭代法

**最难算法：IRLS**

Iterative Reweighted least Squares算法（牛顿-拉夫森迭代）

补充知识：海森矩阵

$$\mathbf{H} = \frac{\partial^2 E(\mathbf{w})}{\partial \mathbf{w}^2}$$

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \mathbf{H}^{-1} \nabla E(\mathbf{w})$$

$$\nabla E(\mathbf{w}) = \sum_{n=1}^N (y_n - t_n) \phi_n = \Phi^T (y - t)$$

**拓展：多类回归**

使用softmax函数

## 生成模型的引入

类似于Metropolis-Hasting算法

- 二分类问题  
 $a = \ln \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)}$  连续输入特征 =  $\mathbf{w}^T \mathbf{x} + w_0$
- 多分类问题  
 $a_k = \ln p(\mathbf{x}|C_k)p(C_k)$  连续输入特征 =  $\mathbf{w}_k^T \mathbf{x} + w_k0$

似然函数： $p(\mathbf{t}|\pi, \mu_1, \mu_2, \Sigma) = \prod_{n=1}^N [\pi \mathcal{N}(\mathbf{x}_n|\mu_1, \Sigma)]^{t_n} [(1 - \pi) \mathcal{N}(\mathbf{x}_n|\mu_2, \Sigma)]^{1-t_n}$

## Kernel Methods & SVM

**决策边界（最大间隔超平面）的进一步优化：最大间隔分类器**

基本定义：

1. 间隔：决策边界和任意样本点之间的最小距离  
推导：任意向量 $\mathbf{x}$ 到决策平面的有符号垂直距离

$$r = \frac{\mathbf{w}^T \mathbf{x} + w_0}{\|\mathbf{w}\|}$$

2. 支撑向量：确定间隔位置的关键样本点

## SVM的对偶问题：最大间隔优化

对偶问题就是把求导得到的最优解代入原拉格朗日方程

- 二分类器的求解（拉格朗日方程法）
  - i. 优化函数：最大化间隔

$$\arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n [t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b)] \right\}$$

在 $n$ 个数据点中，找到距离最小的那个点，并尽可能使该点距离最大  
通过提出来一个 $\alpha$ 进行归一化

- ii. 约束条件：确保全部分类正确
  - iii. 拉格朗日方程： $L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{n=1}^N a_n \{t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) - 1\}$
- 分类器形式

$$y(x) = \sum_{n=1}^N a_n t_n k(x, x_n) + b$$

## 实际情况：样本交错的处理

两种思路：

- 软间隔：松弛变量 + 惩罚因子（penalty）

$$+ C \sum_{i=1}^N \epsilon_i$$

目的：减少过拟合（类似正则化）

- 核函数 + 非线性变换（详见后文）

## 拓展：多分类SVM

将样本分为 $k$ 类

多分类线性模型

两种实现方式：

- $k$  个分类器（存在问题：分类器独立；样本不平衡）
- $k(k-1)/2$  个分类器【“二维遍历”（单循环赛）】

## SVM用于回归

引入  $\epsilon - insensitive$

$$E_{\epsilon}(y(x) - t) = \begin{cases} 0 & \text{if } |y(x) - t| < \epsilon, \\ |y(x) - t| - \epsilon & \text{if } otherwise \end{cases}$$

钝感的带阈值整流器

## 核方法的引入

从原始空间到特征空间的升维映射

核函数具有对称性

核函数对应于一个未知特征空间的内积

Kernel	形式
多项式核	$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + c)^M$
高斯核	$k(\mathbf{x}, \mathbf{y}) = \exp[-\frac{\ \mathbf{x}-\mathbf{y}\ ^2}{2\sigma^2}]$
拉普拉斯核	$k(\mathbf{x}, \mathbf{y}) = \exp[-\frac{\ \mathbf{x}-\mathbf{y}\ }{\sigma}]$
反曲核	$k(\mathbf{x}, \mathbf{y}) = \tanh(a\mathbf{x}^T \mathbf{y} + b)$

### 从零构建一个核函数

$$\begin{aligned}
 k(\mathbf{x}, \mathbf{y}) &= \phi(\mathbf{x})^T \phi(\mathbf{y}) \\
 \mathbf{x} &= (x_1, x_2, \dots, x_n)^T \\
 \mathbf{y} &= (y_1, y_2, \dots, y_n)^T
 \end{aligned}$$

### 基于已有核构造核函数的系列方法

- $k(\mathbf{x}, \mathbf{y}) = ck_1(\mathbf{x}, \mathbf{y})$
- $k(\mathbf{x}, \mathbf{y}) = f(\mathbf{x})k_1(\mathbf{x}, \mathbf{y})f(\mathbf{y})$
- $k(\mathbf{x}, \mathbf{y}) = q(k_1(\mathbf{x}, \mathbf{y}))$
- $k(\mathbf{x}, \mathbf{y}) = k_1(\mathbf{x}, \mathbf{y}) + k_2(\mathbf{x}, \mathbf{y})$
- $k(\mathbf{x}, \mathbf{y}) = k_1(\mathbf{x}, \mathbf{y})k_2(\mathbf{x}, \mathbf{y})$
- $k(\mathbf{x}, \mathbf{y}) = k_3(\phi(\mathbf{x}), \phi(\mathbf{y}))$
- $k(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{A} \mathbf{y}$
- $k(\mathbf{x}, \mathbf{y}) = k_a(\mathbf{x}_a, \mathbf{y}_a) + k_b(\mathbf{x}_b, \mathbf{y}_b)$
- $k(\mathbf{x}, \mathbf{y}) = k_a(\mathbf{x}_a, \mathbf{y}_a)k_b(\mathbf{x}_b, \mathbf{y}_b)$

### 核方法的推广

# Bayesian Classification & Probabilistic Graphical Models

关于贝叶斯分类器的前置知识

## 分类器的两种实现

- 朴素实现——朴素贝叶斯分类器
- 高级实现——概率图模型

Motivation:随机变量多且复杂；概率图模型直观简便且计算可视化  
 also 条件独立性的重要意义

关于图论的前置知识：

- 有向图(a.k.a. 时间关联的*贝叶斯网络*)
- 无向图 (a.k.a. 空间或上下文关联的*MRF-马尔可夫随机场*)
- 团
- 最大团
- 势函数
- 条件独立性

重点考察方法：D 分离 & 条件移除

尾到尾节点（分支）

头到尾节点（串联）

头到头节点（合流）

一句话理解 D 分离

倘若，我们（AB）之间的所有联系都有x-尾节点属于C或有自己与后代都不属于C的头-头节点  
那么，我们对C条件独立

图模型中推理概念的理解

计算某些节点的边缘分布（以一些节点为条件计算其他节点的后验概率）

局部信息在图中的传播

马尔可夫随机场

关于随机场的前置知识：

- 位点空间
- 相空间
- 随机场
- 邻域系统

吉布斯分布

- 能量函数
- 配分函数

引入状态空间——走向隐马尔可夫模型

有亿点点类似于FSM有限状态机

解决三个基本问题

HMM相关算法主要分为三类。

- 还是知道骰子有几种（隐含状态数量），每种骰子是什么（转换概率），根据掷骰子掷出的结果（可见状态链），我想知道掷出这个**结果的概率**。
- 已知骰子有几种（隐含状态数量），每种骰子是什么（转换概率），根据掷骰子掷出的结果（可见状态链），我想知道每次掷出来的都是哪种骰子（**隐含状态链**）。
- 知道骰子有几种（隐含状态数量），不知道每种骰子是什么（转换概率），观测到很多次掷骰子的结果（可见状态链），我想反推出每种骰子是什么（**转换概率**）。

其中第三种情况，知道隐含状态数量，知道可见状态链，反推转换概率，是最常见的情况。

前向算法（问题一）

定义前向变量：t时刻的状态与t时刻及其之前全部观测序列的联合概率

初始化

递推公式：

$$a_t = p(y_1, y_2, \cdots, y_t, s_t) = p(y_t | s_t) \sum_{s_{t-1}} a_{t-1} p(s_t | s_{t-1})$$

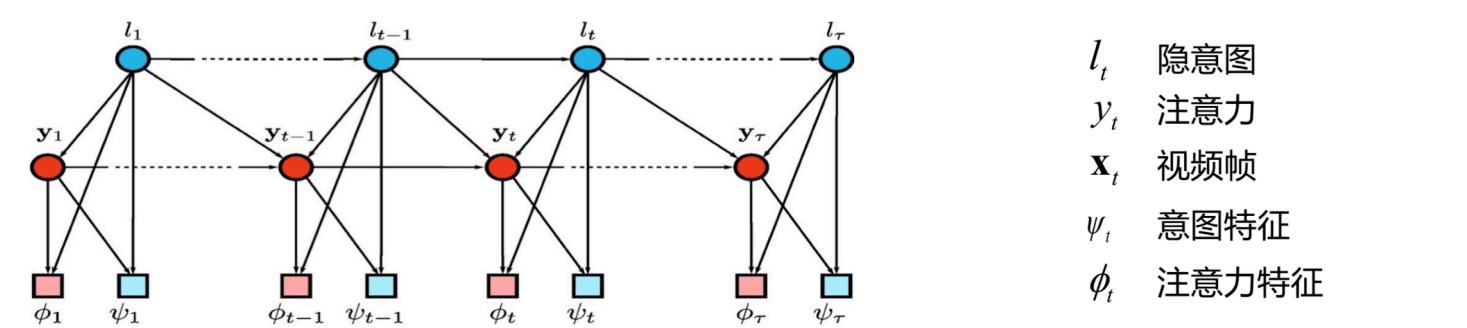
$p(y_t | s_t)$ 为由状态j生成观测 $y_t$ 的概率 $q_{jy_t}$   
 $p(s_t | s_{t-1})$ 为由状态i转移为状态j的概率 $p_{ij}$

算法复杂度：直接求和算法 $O(2n|S|^n)$ ->前向算法 $O(n|S|^2)$

维特比算法（问题二）

EM算法（问题三）

1. 求期望值
2. 最大化



$p(\mathbf{X}, \mathbf{l}, \mathbf{Y}) =$

$$\underbrace{p(l_1)p(y_1|l_1)}_{\text{初始分布}} \underbrace{\prod_{t=1}^{\tau} p(\phi(\mathbf{x}_t)|l_t, y_t)}_{\text{意图匹配}} \underbrace{\prod_{t=2}^{\tau} p(l_t|l_{t-1})}_{\text{意图转移}} \underbrace{\prod_{t=1}^{\tau} p(\phi(\mathbf{x}_t)|y_t, l_t)}_{\text{注意力回归}} \underbrace{\prod_{t=2}^{\tau} p(y_t|y_{t-1}, l_t, l_{t-1})}_{\text{注意力转移}}$$

$\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_\tau)$  输入视频数据

$\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_\tau)$  注意力序列

$\mathbf{l} = (l_1, \dots, l_\tau)$  隐意图序列

该公式要求看图能写出来

背景

这个公式的目的是用来描述在一个视频中，我们如何根据观察到的内容（比如一个人的行为）推断出这个人注意力的变化过程。

公式的各个部分

1. 初始状态:
- $p(l_1)$  是对一开始隐藏状态  $l_1$  的猜测。隐藏状态可能表示一些我们不能直接看到的东西，比如人的心理状态。
  - $p(y_1|l_1)$  是在隐藏状态  $l_1$  下，注意力状态  $y_1$  的概率。注意力状态是我们关心的东西，比如这个人一开始在看哪里。
2. 观测概率:
- 视频帧  $x_t$  是我们在第  $t$  时刻看到的画面。
  - $\phi(x_t)$  是视频帧的特征，比如说画面中的人正在做什么。
  - $p(\phi(x_t)|l_t, y_t)$  是在隐藏状态  $l_t$  和注意力状态  $y_t$  下，看到这些特征的概率。这部分告诉我们，在某个特定的心理状态和注意力状态下，我们有多大可能看到这个人正在做的事情。
3. 状态转移:
- $p(l_t|l_{t-1})$  是前一个隐藏状态  $l_{t-1}$  转移到当前隐藏状态  $l_t$  的概率。
  - $p(y_t|y_{t-1}, l_t, l_{t-1})$  是在前一个注意力状态  $y_{t-1}$  和隐藏状态  $l_{t-1}$  和  $l_t$  下，转移到当前注意力状态  $y_t$  的概率。这部分描述了注意力和隐藏状态如何随时间变化。

组合这些概率

将这些部分组合起来，我们得到整个过程的联合概率：

$$p(\mathbf{X}, \mathbf{l}, \mathbf{y}) = p(l_1)p(y_1|l_1) \prod_{t=1}^T p(\phi(x_t)|l_t, y_t) \prod_{t=2}^T p(l_t|l_{t-1}) \prod_{t=2}^T p(y_t|y_{t-1}, l_t, l_{t-1})$$

这个公式的每一部分都在说明一个方面的情况：

- 初始状态和初始注意力状态：一开始的猜测。
- 每一帧的观测：我们看到的每一帧视频在给定状态下有多大可能性。

- **状态转移**：状态和注意力状态如何从一个时间步转移到下一个时间步。

## 举个例子

假设我们在看一个视频，视频中的人从走到水杯  那里、检查水杯的状态、然后查看书  的状态。这三个动作依次发生。

1. **初始状态**：我们猜测这个人最开始在寻找水杯。
2. **观测概率**：我们看到视频帧中这个人确实在看水杯，这符合我们对它注意力的猜测。
3. **状态转移**：从看水杯到检查水杯的状态，然后再转移到查看书的状态。

通过这种方法，我们可以一步步地根据视频中的画面来推测出人的注意力状态变化，并计算出每一步的可能性。这就是这个公式的主要作用：帮助我们理解和计算一个复杂过程中的各个状态和观察结果的关系。

## Ensemble Learning

Classification：

- 同质：基学习器
- 异质：组件学习器

集群与性能的关系：

不同弱分类器能够正确分类的样本有所重叠又有所不同

## Boosting

依赖 / 串行

### 机制

特点：

- 重赋权法：  
通过更改上一轮中分错样本的权重（重赋权法），使得其**损失函数的函数值增大**，**基分类器**就会倾向于将权重大的这些样本（即错误样本）学对
- 重采样法：

## 重采样:

设我们有五个样本  $(x_1, y_1)$   $(x_2, y_2)$   $\dots$   $(x_5, y_5)$ .

$h_0$  初始权重  $w_0 = [0.2, 0.2, 0.2, 0.2, 0.2]$ .

弱分类器<sup>y</sup>分类错误: 2.4 样本, 2.4 样本被赋更高权重:

归一化后  $w_1 = [0.14, 0.29, 0.14, 0.29, 0.14]$

中生成训练下一个弱分类器样本:

第一个样本以  $w_1$  为概率分布抽取样本.

第二, 三, 四, 五同理.

$\Rightarrow (x_1, y_1)$   $(x_2, y_2)$   $(x_3, y_3)$   $(x_4, y_4)$   $(x_5, y_5)$

为新样本集.

且抽概率  $h_1$

如果  $h_1$  性能比随机猜测差, 则不生成新权重, 用上一次有

效权重再一次重采样生成新的  $h_1$ .

### 典型算法: Adaboost

#### 算法口述

1. 先用初始权重下的样本全体训练一个弱分类器
2. 计算出分类误差, 若整体分类误差大于某个阈值则重新训练
3. 若误差满足阈值条件, 则根据该分类误差计算出一个权重因子, 结合规范化因子对样本个体的权重进行调整, 并为把权重因子记录为该弱分类器的分类器权重
4. 用调整权重后的样本全体再次训练一个弱分类器, 并再次进行步骤2和3
5. 迭代步骤4
6. 对所有训练完成的弱分类器按照分类器权重表进行加权得到一个强分类器

### 应用

#### Bagging

##### 自主采样法: bootstrap sampling

- 采样出  $T$  个 (每个含  $m$  个样本) 的采样集
- 每次采样后放回, 故  $T$  个采样集间可能有重叠 (确保概率分布一致性)
- 基学习器



- 简单投票法/简单平均法

k-fold交叉验证

不依赖 / 并行

机制

算法口述

特点：

典型算法：Decision Tree 🌲

算法口述

构建分类树

算法口述

1. 构建根节点，将所有训练数据放在根节点，选择一个**最优特征**，按照这一特征将训练数据集分割成子集，使得各个子集在当前条件下最好的分类
2. 如果这些子集已经能够被基本正确分类，那么构建叶节点，并将这些子集分到所对应的叶节点中去
3. 如果还有子集不能被基本正确分类，那么就对这些子集选择新的**最优特征**，继续对其进行分割，构建相应的节点
4. 如此**递归**地进行下去，直至所有训练数据子集被基本正确分类，或者没有合适的特征为止

选择最优划分属性

算法口述

度量纯度：信息增益方法

$$Gain(D,a) = Ent(D) - \sum_{v=1}^V \frac{|D^v|}{|D|} Ent(D^v)$$

Ent here is the entropy function measuring the purity of one sample set

相对熵（信息增益）

Random Forest

由Decision Tree 到Random Forest的过程才是Bagging思想的核心体现

算法口述

1. Bagging
2. 在每个节点上所有的样本特征中**随机选择**K个样本特征，并从这K个样本特征中选择一个最优的进行决策树的子过程

应用

Boosting & Bagging 对照表

分类	Boosting	Bagging	Random Forest
结构	多个弱学习器串行生成， 后一个学习器根据前一个学习器的错误率进行调整	多个弱学习器并行生成， 每个学习器都在不同的样本子集上训练	基于Bagging的思想， 结合随机特征选择的多决策树模型
特点	1. 个体学习器存在强依赖关系， 串行生成 2. 错分样本得到更多关注	1. 个体学习器不存在强依赖关系， 并行生成	1. 结合Bagging和随机特征选择 2. 每个树的训练样本和特征都是随机选择的
优点	1. 提高了弱学习器的准确率 2. 能够有效降低偏差	1. 时间复杂度低 （基学习器的计算复杂度与平均过程的复杂度） 2. 方便推广至多分类及回归任务	1. 简单、容易实现、计算开销小 2. 基学习器的多样性不仅来自 <b>样本扰动</b> ， 还来自 <b>属性扰动</b> 3. 训练效率优于Boosting 4. 能处理高维数据，防止过拟合

分类	Boosting	Bagging	Random Forest
缺点	1. 只适用于二分类任务 2. 计算复杂度高，训练时间长 3. 对噪声数据敏感	1. 需要更多的存储空间 2. 模型解释性较差	1. 需要更多的存储空间 2. 模型解释性较差 3. 对训练数据的依赖较大
考点	1. 弱学习器的选择与权重更新 2. 错误率的计算与调整机制	1. 样本重采样方法 2. 并行训练机制	1. 决策树的生成与剪枝 2. 特征选择与随机性 3. 多树结果的综合方法

## 前馈神经网络🧠

Feedforward NN / MLP  
 结构：网络中的变量和它们的拓扑关系  
 激励函数：神经元如何根据其他神经元的活动改变自己的激励值  
 学习规则：网络中权重如何随着时间推进而调整

## 构建 $\phi$ 的三种方式（ML关注的问题）

- 通过映射，将 $x$ 变换至无限维空间 $\phi(x)$
- 手动设计 $\phi$ ,以人的经验选择特征，如边缘、HOG、SIFT
- 自主学习 $\phi$ ,从数据中挖掘和学习实现某一任务最佳 $\phi(x)$ （因此DL属于ML的子类）

## 普遍近似原理

牢记几个关键词：

- “挤压”非线性激活函数 -> 线性输出层
- 足够数量的隐藏单元
- 任意精度近似
- 从一个**有限维空间**到另一个**有限维空间**

## 激活函数

- $Sigmoid(x) = \frac{1}{1+e^{-x}}$
- $tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- $ReLU(x) = max(0, x)$
- $softmax(\mathbf{z})_i = \frac{exp(\mathbf{z}_i)}{\sum_j(\mathbf{z}_j)}$  (习惯上把 $z$ 叫做logits)
- $LeakyReLU(x) = max(\alpha x, x)$
- $ExponentialLU(x) = \begin{cases} x & \text{if } x > 0, \\ a(e^x - 1) & \text{if } otherwise \end{cases}$

## 代价函数

- 范数代价函数（ $L_2$ 范数）
- 范数代价函数-平均绝对误差（ $L_1$ 范数）

## 输出单元

## 架构设计

默认：输入层即为神经网络的第0层  
 $f(x) = f^{(3)}(f^{(2)}f^{(1)}(x))$  -> 包含2个隐藏层和一个输出层的神经网络

# 卷积神经网络

理解：卷积神经网络引入卷积操作，可以**对于局部数据提取特征（局部感知）**，并且可以对**同一数据进行多种卷积得到不同特征（参数共享）**，最后对于得到的**特征在不改变特征情况下进行降维（池化）**  
'端到端'：卷积神经网络通过卷积操作、池化操作、非线性激活函数映射等一系列操作的层层堆叠，将高层语义信息逐层由原始数据输入层中提取出来，逐层抽象，直到完成目标任务。

## 结构

### Pipeline

CONV->ReLU->POOL->CONV->ReLU->POOL->.....->FC

输入层：

- 1. 去均值
- 2. 归一化

## 卷积单元

padding：对于“越卷越小”现象的处理

- 1. 复制边界值
- 2. 补“0”

### 卷积计算题

$$Output\ Size = \left\lceil \frac{Input\ Size - Kernel\ Size + 2 \times Padding}{Stride} \right\rceil + 1$$

### 局部感知域

- 1. 计算参数量减少
- 2. 只提取局部特征，保留局部不变性（不编码位置信息）

### 参数/权值共享

卷积是可学习的

### 激活函数

非线性映射层

### 多通道卷积

多个卷积模板并行

### 池化

- 1. 最大值池化
- 2. 平均池化

3	6	1	4
4	7	7	8
2	2	1	2
2	4	3	4

Max pooling

7	8
4	4

Average pooling

5	5
2.5	2.5

#### 作用

1. 特征不变性：只关注特征是否存在
2. 特征降维性：减少参数数量（保留抽象信息），避免过拟合

## 全连接层

### 实例

1. LeNet-5 7层
2. Alexnet 8层
3. VGG
4. Inception
5. ResNet
6. R-CNN
7. DeepID

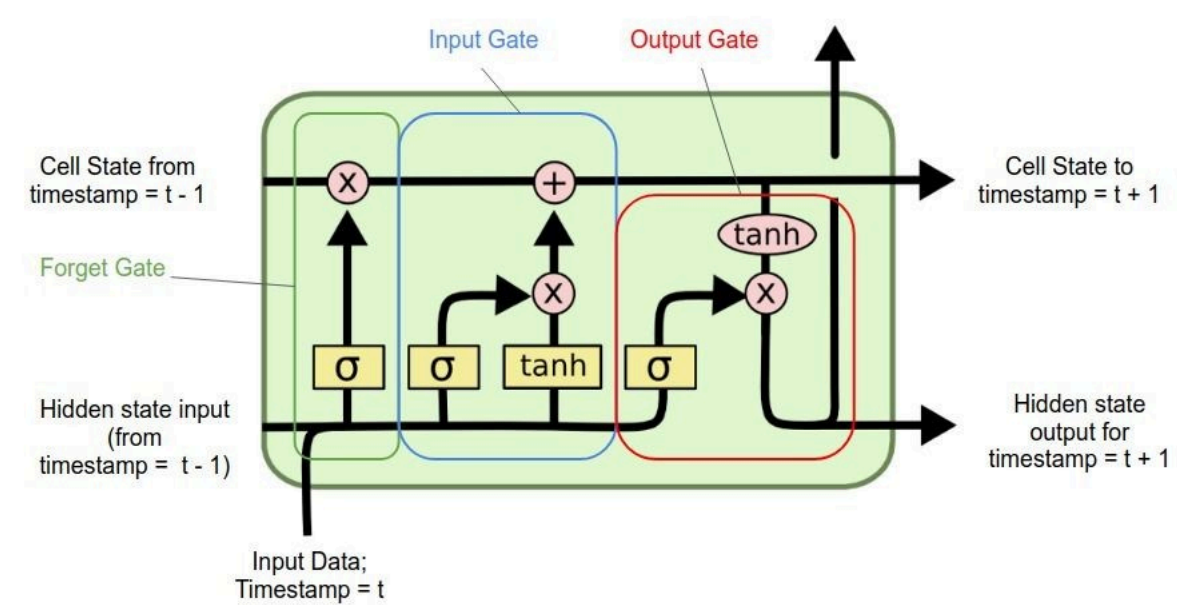
## 循环神经网络

定义：是一种具有从后续层到前面层反馈连接或者同层之间神经元连接的神经网络，常用于处理顺序数据。

理解：通过一个隐层来确定信息是否存储，考试的时候会考LSTM的内部结构

结构

LSTM



遗忘门

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

输入门

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

输出门

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = O_t \cdot \tanh(C_t)$$

状态更新

信息输出

无监督学习与聚类

无监督学习：在无监督学习中，训练样本的标记信息是未知的，目标是通过对无标记训练样本的学习来揭示数据的内在性质及其规律，为进一步的数据分析提供基础。

聚类

聚类：是无监督学习中应用最广、研究最多的一个内容，其目的是能够自动将未标记的数据根据自身的特点划分为若干个通常是不相交的子集（“簇”（cluster））。通过聚类算法，不仅可以自动组织数据，还能挖掘一些数据的隐藏结构和属性，如：“浅色瓜”...聚类算法也可以作为其他数据处理的基

础，如数据降维、可视化。

## 应用

通过话题聚类网页；根据表达式聚类蛋白质序列；根据消费记录对客户进行分类

## 定义

### 性能度量

外部指标：对比聚类结果和**参考模型**的簇划分、聚类结果的簇标记向量和参考模型的簇标记向量。（类似于baseline）

内部指标：基于**样本间的距离**和**簇中心点间的距离**。（样本间离得越近越好）

### 距离计算

闵可夫斯基距离：

- 欧氏距离
- 曼哈顿距离

### 性质

非负性

对称性

## K均值聚类

### 算法口述

1. 从样本集中随机选取k个均值向量
2. 遍历样本集中的每一个样本，计算出它们与所有均值向量的距离并选取最近的均值向量为他们所属的簇，记录它们的簇标记（**A**）
3. 对每个簇中的样本向量计算出它们的新（真正的）均值向量，更新均值向量（**B**）
4. 重复进行步骤**A** & **B**，直到均指向量不再出现变动

## 密度聚类

### 术语

#### 核心对象

#### 密度直达

单向性

#### 密度可达

单向性

#### 密度相连

双向性

#### 簇定义

1. 若两个样本点属于同一簇，则它们一定密度相连（逆命题成立）
2. 由给定簇中任一样本点密度可达的点同样属于该簇

## DBSCAN算法

### 算法口述

1. 遍历每个样本点，对每个样本点按照给定的 $\epsilon$ 邻域统计出邻域范围内的其他样本点数量，若达到 $MinPts$ 阈值，则标记该样本点为核心对象，并将其纳入**核心对象**集合
2. 再从任一**核心对象**出发，找出由其**密度可达**的样本生成聚类簇，直到所有**核心对象**均被访问过为止
3. 完成簇划分，簇之外为噪声点

# 层次聚类

试图在不同层次对数据进行划分，从而形成树形的聚类结构。数据集的划分可采用**自底向上**的聚合策略，也可采用**自顶向下**的分拆策略。（类似于归并排序）

## AGNES算法

### 算法口述

- 1. 以样本集的每个点单独为一个簇的形式初始化聚类簇
- 2. 通过“二维遍历”（单循环赛）的形式计算出每两个**聚类簇间的距离**，形成初始的距离矩阵
- 3. 设置当前聚类簇个数q
- 4. 按照归并排序的逻辑按距离由近及远依次找出最近的聚类簇
- 5. 对于聚类过后的聚类簇更新他们在距离矩阵中的占位
- 6. 直到聚类簇的个数减少至设置值

### 聚类簇距离定义方式

- 最小距离
- 最大距离
- 平均距离

## 聚类方法对照表

分类	K-Means聚类	密度聚类	层次聚类
结构	将数据分成K个簇， 每个簇由一个质心代表， 不断迭代以最小化簇内方差	基于数据点的密度， 将高密度区域的点聚类在一起，常用DBSCAN算法	创建一个层次结构， 通过不断合并或拆分簇来构建聚类树
特点	适合球状簇，快速收敛，需预设簇数	不需要预设聚类数目，能够发现任意形状的簇， 对噪声有鲁棒性	生成树状结构， 能展现数据的多级别聚类关系
优点	1. 简单易实现 2. 计算速度快，适合大规模数据集 3. 对球状聚类效果较好	1. 不用先验地设置聚类数目 2. 对数据集的凹凸性不做限制 3. 能够自发地发现异常点， 因此聚类结果不会受异常点干扰 4. 不具有初值依赖性	1. 不需要预设聚类数目 2. 能生成聚类树， 便于观察数据的多层次结构 3. 能处理噪声和异常值
缺点	1. 需要预设聚类数目K 2. 对初值敏感，容易陷入局部最优 3. 只适合球形聚类， 不能处理非凸形状的簇	1. 要求样本集密度均匀且样本点间距不要太大 2. 聚类收敛时间较长 3. 需要对 $\epsilon$ 和 MinPTs 联合调参， 对调参水平要求较高	1. 计算复杂度高，适合小规模数据 2. 无法自动确定聚类数目 3. 需要定义合并或拆分标准， 容易受噪声影响
考点	1. 初始质心选择 2. 簇内方差计算 3. 算法收敛条件	1. 密度阈值 $\epsilon$ 和最小点数 MinPTs 的选择 2. 算法复杂度 3. 噪声点处理	1. 聚类树的构建与剪枝 2. 距离和相似度度量 3. 合并和拆分标准的选择

# 采样方法

采样定义：从一个分布中生成一批服从该分布的样本。  
采样的本质上是对随机现象的模拟，根据给定的概率分布，来模拟产生一个对应的随机事件。采样可以让人们对随机事件及其产生过程有更直观的认识。

采样的作用：采样得到的样本集也可以看做是一种非参数模型，即用较少的样本点（经验分布）来近似总体分布，并刻画总体分布中的不确定性。从这个角度来讲采样其实也是一种信息的降维，起到简化问题的作用。  
在机器学习中，可能会遇到样本量过大或模型结构复杂导致的求解难度大、没有显示解析解等问题，这种情况下，可以利用采样方法进行模拟，从而对这些复杂模型进行近似求解或推理。一般会转化为**某些函数在特定分布下的积分或期望**，或者是求某些随机变量或参数**在给定数据下的后验分布**。

# 蒙特卡罗方法(MCMC)

基本问题：寻找某个定义在概率分布 $p(z)$ 上的函数 $f(z)$ 的期望  
(一类算法的统称)

## 常见的采样方法

### 均匀分布采样

从均匀分布  $U(0, 1)$  及其变体中采样  
一切采样的基础

### 逆变换采样

由万流归宗原理简单推导出的**积分+反函数**公式

#### 算法口述

- 1. 已知目标分布的CDF:  $z = \Phi(X) = \int_{-\infty}^X p(x)dx$
- 2. 从均匀分布  $U(0,1)$  产生一个随机数 $z_i$
- 3. 计算逆函数 $X_i = \Phi^{-1}(z_i)$ ，循环上述步骤

### 拒绝采样

术语：  
Proposal distribution：提议分布

#### 算法口述

- 1. 从提议分布 $q(z)$ 中采样生成一个样本 $z_0$
- 2. 生成区间 $[0, kq(z_0)]$ 上的均匀分布的一个样本 $u_0$  (注意此处的 $q(z_0)$ 特指对应的PDF上的函数值)
- 3. 若 $u_0 > \tilde{p}(z_0)$ ，则该样本被拒绝；否则 $z_0$ 被接受
- 4. 重复上述过程得到 $[z_0, z_1, ..., z_n]$ 即是对 $p(z)$ 的一个近似

### 重要（性）采样

当函数值较大的点对应的概率较小（采样方差较大，不稳定）时，为了保证尽可能估计出期望的近似值，通过引入提议分布参与构成重要性权重来修正采样引入偏差的一种方法

#### 公式推导

假如目前我们的分布是 $p(z)$ ， $f(z)$ 是我们的函数，则 $E(f) = \int_z f(z)p(z)dz$ ，但是这个分布不方便使用或者说该分布不适合采样，会引入较大的bias，那么我们就提供一种新的分布 $q(z)$ 。

接下来改写积分内的式子为： $p(z)f(z) = q(z)\frac{p(z)}{q(z)}f(z)$ ，那么根据期望的定义： $E(f) = \int_z q(z)\frac{p(z)}{q(z)}f(z)dz$ ，这个式子意思是 $\frac{p(z)}{q(z)}f(z)$ 在分布 $X \sim q(z)$ 上的数学期望。

我们在 $q(z)$ 上采样 $z^{(1)}, z^{(2)}, ..., z^{(l)}$ ，然后通过求取样本均值就可以估计出在 $q(z)$ 分布下随机变量 $f(X)$ 的数学期望。 $E(f) = \frac{1}{L} \sum_{l=1}^L \frac{p(z^{(l)})}{q(z^{(l)})} f(z^{(l)})$

其中 $\tilde{r} = \frac{p(z^{(l)})}{q(z^{(l)})}$ 叫重要性权重（修正因子）。

#### 缺陷

重要性采样可以改变原来的旧分布，用新的分布去采集样本，然后求出目标期望，上述证明显示两者理论上是等价的，但是等价他有个前提条件：就是2个分布不能相差太大。换句话说，如果2个分布相差过大，那么两者就不会相等，这就是重要性采样的缺陷。

重要性采样确实可以让2个分布产生一个期望，但是期望相等并不代表方差相等。如果二者相差过大，就会导致双方的方差过大。而当采样数据不足够时，方差相差太大会导致两者的样本均值相差很大！

### Metropolis-Hastings方法

通过引入马尔可夫链（考虑到其平稳性）来对上一采样点剩余区间进行动态采样，并在假设提议分布对称的基础上的改良方法  
详见【[Metropolis-Hastings - 可视化解释（中文字幕）-哔哩哔哩](#)】。



Metropolis算法口述（假设提议分布 $q(z)$ 是对称的）

注：最大迭代次数 $T$ ，根据需要截取尾部 $n$ 个样本  
减少初始样本的偏差对最终结果的影响，确保生成的样本集更好地反映目标分布的特性。

改进版（无需假设提议分布 $q(z)$ 是对称的）算法口述

Algorithm 1: MH 采样算法流程

```
Input: 目标分布  $p$ , 提议分布 (proposal distribution)  $q$ ,
采样数  $N$ 
Output: 符合目标分布的采样
1 初始化  $x_0, i \leftarrow 0$ ;
2 while  $i \leq N$  do
3    $x \leftarrow x_i$ ;
4   对  $x' \sim q(x'|x)$  进行采样;
5   计算接受概率  $\alpha_{ij}$ 
                                     
$$\alpha_{ij} = \frac{p(x')q(x|x')}{p(x)q(x'|x)}$$

6   计算  $r = \min(1, \alpha_{ij})$ ;
7   对  $u \sim U(0, 1)$  进行采样;
8    $i \leftarrow i + 1$ ;
9   if  $u < r$  then
10     $x_i = x'$ 
11  else
12     $x_i = x$ 
13  end
14  记录采样  $x_i$ 
15 end
```

Gibbs采样

Metropolis-Hastings方法的一个三变量特例

采样方法对照表

分类	均匀采样	拒绝采样	重要采样	Metropolis采样
结构	从均匀分布中直接采样	从候选分布中采样并根据接受概率筛选	从易于采样的分布中采样，并对样本赋予权重	基于当前样本生成候选样本，接受概率只依赖当前和候选样本的比
特点	简单直接，适用于已知分布	可用于复杂分布，但有时效率低	提高了稀有事件的采样效率	对目标分布的依赖较弱，适用于高维分布
优点	实现简单，无需额外计算	理论上适用于任何分布，概念直观	可减少方差，提高采样效率	简单易实现，不需要计算归一化常数
缺点	仅适用于均匀分布，无法处理复杂分布	低效，尤其是接受率低时	需要计算重要性权重，可能复杂	初始值影响较大，需较长烧入期

分类	均匀采样	拒绝采样	重要采样	Metropolis采样
考点	适用于简单均匀分布的采样方法	理解接受-拒绝机制，适用于复杂分布	理解权重计算，提高采样效率	理解Markov链的构建和收敛性

采样方法与ML的综合运用🦉

采样和模拟

估算未知量

贝叶斯推理

学习和模型估计

与神经网络结合

- 1. 受限玻尔兹曼机(RBM)
- 2. 大规模多分类任务

写在后面(小彩蛋🟡)

Kernel小专题

核函数的证明

从构建的角度证明

核函数的思想

- 基函数
- 核函数
- 激活函数

三种NN对比表

分类	前馈神经网络	卷积神经网络	循环神经网络（一般/LSTM）
结构	$f(x) = f^{(3)}(f^{(2)}f^{(1)}(x))$	CONV->ReLU->POOL->CONV->ReLU->POOL->.....->FC	遗忘门-输入门-输出门
特点	1. 前向传播 2. 各层全连接	1. 局部感知 2. 参数共享 3. 池化	1. 能处理序列数据 2. 有记忆功能
优点	1. 结构简单 2. 易于理解和实现	1. 处理图像数据效果好 2. 参数共享减少了参数数量 3. 局部感知降低了计算复杂度	1. 能处理时间序列数据 2. LSTM能有效缓解梯度消失问题 3. 能捕捉长期依赖关系
缺点	1. 较难处理序列数据 2. 缺乏长期记忆能力 3. 参数太多 4. 不满足局部不变性	1. 对平移和旋转不敏感 2. 需要大量计算资源	1. 梯度消失 2. 难以刻画长期依赖关系（一般RNN） 3. 训练复杂，时间长
考点	1. 前向传播和反向传播算法 2. 激活函数的选择	1. 卷积操作与池化操作 2. 参数共享和局部感知	1. 循环结构与序列数据处理 2. LSTM的门控机制