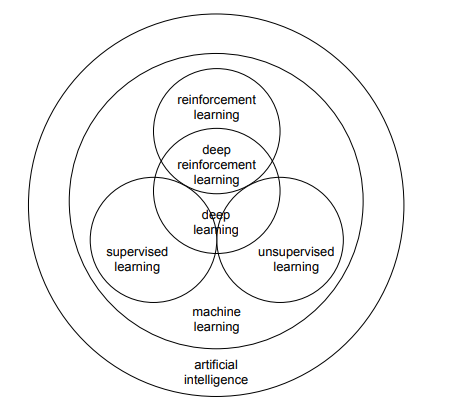
# 机器学习笔记

# 2025.1.2

机器学习是利用“经验”（数据）改善系统自身性能，重要的理论基础是概率近似正确（Probably Approximately Correct，PAC）理论：

机器学习、深度学习和人工智能的关系：

[](https://arxiv.org/pdf/1810.06339)

* 机器学习相关概念：

数据集、训练集、验证集、测试集、样本、记录、特征（属性）。

泛化（Generalization）：指模型在新的、未见过的数据上的表现能力。过拟合（Overfitting）

是指模型在**训练数据上表现得非常好**，**但在新的、未见过的数据上表现不佳**；欠拟合（Underfitting）是指模型在**训练数据上表现不佳**，无法捕捉数据的基本结构和规律。

模型是得到了一个规律，可以高概率解释输入与输出的映射，模型得出的是假设，不一定对。事实的真相是一定对的。

监督学习（supervised learning）：有“导师”的学习。数据集中给出期待结果。典型问题：分类、回归。

无监督学习（unsupervised learning）：无“导师”的学习。数据集中没有给出期待结果，无法得到判断“西瓜好坏”的模型。典型问题：离散问题的聚类。

* 开发机器学习应用程序的步骤

1. **收集数据**。
2. **准备输入数据**：得到数据之后，还必须确保数据格式符合要求。此外还需要为机器学习算法准备特定的数据格式。
3. **分析输入数据**：主要是人工分析得到的数据。查看是否有空值或异常值，可以用可视化工具进行展示数据。这一步是确保数据集中没有垃圾数据。
4. **训练算法**。
5. **测试算法**。
6. **使用算法**。

# 2025.1.4

错误率：m个样本中有a个样本分类错误。精度是“1 – 错误率”。

|  |  |
| --- | --- |
| 真正例 TP | 实际为正类的样本被模型预测为正类 |
| 假正例 FP | 实际为反类的样本被模型预测为正类 |
| 真反例 TN | 实际为反类的样本被模型预测为反类 |
| 假反例FN | 实际为正类的样本被模型预测为反类 |

查准率和查全率是矛盾的，一般来说，查准率高，则查全率低。

查准率P：

查全率R：

* 评估方法

只有一个包含m个样例的数据集D，既要训练又要测试。此时可以通过对D进行适当的处理，从中产生训练集S和测试集T。

|  |  |  |  |
| --- | --- | --- | --- |
| **评估方法** | **步骤** | **优点** | **缺点** |
| 留出法 | 将数据集分为两个不相交的子集，一个是训练集，一个为测试集 | 简单；  只需要一次训练和测试 | 数据集小时，不准确，方差大 |
| 交叉验证法 | 将数据集分成K个大小相似的互不重叠的子集，其中K-1个子集用于训练模型，剩下的一个子集用于测试模型，重复K次，每次使用不同的子集作为测试集，最后取K次测试结果的平均值作为模型的评估结果 | 可以充分利用数据集；  适合小到中等的数据集 | 计算成本高  由于每个样本都被作为一次验证集，可能引入噪声；  K过大会导致模型保守，K过小会引入更多噪声 |
| 自助法 | 从原始数据集中进行有放回的抽样，生成与原始数据集大小相同的新数据集，作为训练集，然后使用原始数据集中未被抽样的样本作为测试集，重复多次，对模型进行训练和评估，最后取多次评估结果的平均值作为模型的最终评估结果. | 数据集小时很有用；  能产生多个不同的训练集；  有放回使模型更加稳健 | 改变了初始数据集的分布，会引入估计误差；  可能会过拟合 |

调参与算法选择没有本质区别，调参：对每种参数配置都训练出模型，然后把对应最好模型的参数作为结果。

* 一元线性回归

一元线性回归是一种统计方法，用于分析一个自变量和一个因变量之间的线性关系。估计一元线性回归模型的最常用方法是**最小二乘法**，它是一种常用的参数估计方法，用于求解一元线性回归模型的斜率w和截距b。

损失函数：

分别对w和b求偏导，并令其等于0，之后进行求解：

# 2025.1.5

单位阶跃函数：

但是单位阶跃函数不连续，需要找到一个单调可微的函数。

对数几率函数：

*Sigmoid 函数即形似S的函数。对率函数是Sigmoid的重要代表函数。*

几率反应了x作为正例的相对可能性。将y视为样本x作为正例的可能性，1 – y 则是其反例的可能性，两者比值即为几率：。对几率取对数，即为对数几率：。

* 梯度下降

梯度下降是一个优化算法，用于通过不断调整参数来最小化目标函数（如损失函数）。

**原理**：通过计算损失函数关于模型参数的梯度，然后沿着梯度的反方向（即最陡峭的下降方向）更新参数。这样，每次迭代都会使损失函数值减小（至少在局部上是这样的），从而逐渐接近损失函数的最小值。

步骤：

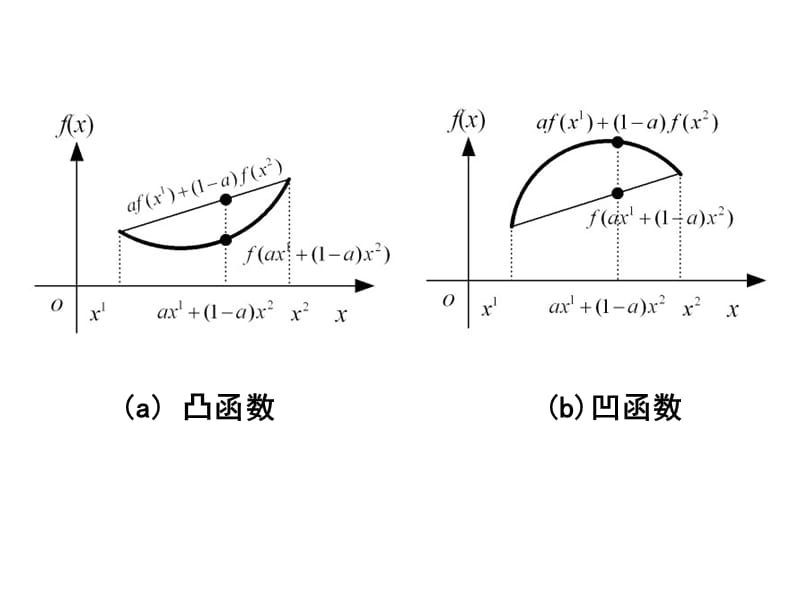
Step 1） 初始化参数：选择一个初始的模型参数向量。

Step 2） 计算梯度：计算损失函数对参数的梯度（或者称为导数），表示损失函数在当前参数值处的变化率。

Step 3） 更新参数：沿着梯度的反方向，以一定的步长（学习率）更新参数的值，使得损失函数逐渐减小。

Step 4） 迭代：重复步骤2和步骤3，直到满足终止条件（如达到最大迭代次数、损失函数收敛等）。

**当目标是凸函数时，梯度下降的解法是全局最优的。**

[](https://image.baidu.com/search/detail?ct=503316480&z=0&ipn=d&word=%E5%87%B8%E5%87%BD%E6%95%B0&hs=0&pn=0&spn=0&di=7448669010159206401&pi=0&rn=1&tn=baiduimagedetail&is=0%2C0&ie=utf-8&oe=utf-8&cl=2&lm=-1&cs=3839994261%2C1929460091&os=1466128256%2C2573398384&simid=3839994261%2C1929460091&adpicid=0&lpn=0&ln=30&fr=ala&fm=&sme=&cg=&bdtype=0&oriquery=%E5%87%B8%E5%87%BD%E6%95%B0&objurl=https%3A%2F%2Ffile1.renrendoc.com%2Ffileroot2%2F2019-12%2F14%2Fe130d581-b233-4755-96b0-b63de0d88ee0%2Fe130d581-b233-4755-96b0-b63de0d88ee04.gif&fromurl=ippr_z2C%24qAzdH3FAzdH3Fooo_z%26e3B6jg6jg15v_z%26e3Bv54AzdH3Frwrj6AzdH3Flnddcabd_z%26e3Bip4s&gsm=&islist=&querylist=&dyTabStr=MCwxMiwzLDEsMiwxMyw3LDYsNSw5)

凸函数性质：

1. 单一全局最优点：局部极小值就是全局极小值
2. 梯度指向函数递减方向：梯度表示函数在点x的变化率，梯度的反方向是函数值减小最快的方向

Python实现：

import numpy as np

# 目标函数和其梯度

def function(x):

    return (x - 3)\*\*2  # 目标函数

def gradient(x):

    return 2 \* (x - 3)  # 目标函数的梯度

# 梯度下降参数

learning\_rate = 0.1  # 学习率

iterations = 100  # 最大迭代次数

x = np.random.rand() \* 10  # 初始化 x 为一个随机值

# 梯度下降过程

for i in range(iterations):

    grad = gradient(x)  # 计算梯度

    x = x - learning\_rate \* grad  # 更新 x

# 输出结果

print(f"最终 x: {x:.4f}, 最终 f(x): {function(x):.4f}")

# 2025.1.6

* 逻辑回归

多元线性回归方程的一般形式如下：

其矩阵形式为：

使用Sigmoid函数，将x值转换为一个接近0或1的y值：

可以转为：

若将y视为类后验概率估计p(y = 1 | x)，则为：

根据 可以分别得到p\_1 p\_0的计算公式。

采用“**最大似然法**”，求解w,b。

*似然性是一个事件实际已经发生，反推在什么参数条件下，这个事件发生的概率最大。*

采用梯度下降进行求解。

# 2025.1.10

* 决策树

决策树是一种分类和回归方法，是基于各种情况发生的所需条件构成决策树，以实现期望最大化的一种图解法。

**熵**（entropy）是表示随机变量不确定性的度量，熵越大，随机变量的不确定性就越大：

条件熵H(Y|X)表示在已知随机变量X的条件下随机变量Y的不确定性：

*当熵和条件熵中的概率有数据估计得到时，所对应的熵与条件熵分别称为经验熵和经验条件熵。*

**信息增益**（information gain）表示得知特征X的信息而使得类Y的信息不确定性减少的程度。

**决策树学习应用信息增益准则选择特征。给定训练数据集D和特征A，经验熵H(D)表示对数据集D进行分类的不确定性；经验条件熵H(D|A)表示在特征A给定条件下对数据集D进行分类的不确定性。它们的差即为信息增益。**

*根据信息增益准则的特征选择方法是：对训练数据集D，计算其每个特征的信息增益，权责信息增益最大的特征。*

*以信息增益作为划分训练数据集的特征，存在偏向于选择取值较多的特征。使用信息增益比可以对此问题进行矫正。特征A对训练数据集D的信息增益比定义为其信息增益比g(D,A)于训练数据集D关于特征A的值的熵之比：*

|  |  |
| --- | --- |
| **算法名称** | **特征选择准则** |
| ID3 | 信息增益 |
| C4.5 | 信息增益比 |

# 2025.1.12

* 神经网络

**神经网络是由具有普适性的简单单元组成的广泛并行互连的网络，它的组织能够模拟生物神经系统对真实世界物体所作出的交互反应。**机器学习中谈到的神经网络指的是“神经网络学习”，或者说是机器学习和神经网络这两个学科的交叉部分。

神经网络中最基本的成分是神经元模型：如果神经元的电位超过“阈值”，就会被激活，向其他神经元发送信息。

*在简单模型中，神经元接收到来自n个其他神经元传递过来的输入信号, 这些输入信号通过带权重的连接(connection)进行传递, 神经元接收到的总输入值将与神经元的阙值进行比较，然后通过“激活函数”(activation function)处理以产生神经元的输出.*

[图示, 示意图

描述已自动生成](https://blog.csdn.net/weixin_39910711/article/details/100775918?ops_request_misc=&request_id=&biz_id=102&utm_term=%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0%EF%BC%9A%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-3-100775918.142%5ev101%5epc_search_result_base1&spm=1018.2226.3001.4187)

理想激活函数是阶跃函数，将输入映射为0或1，分别对应神经元抑制和兴奋。但其不连续、不光滑，故常用Sigmoid函数。该函数可将大范围输入值挤压到(0,1)输出值范围内，也称“挤压函数”。

[图示

描述已自动生成](https://blog.csdn.net/weixin_48136911/article/details/132024431?ops_request_misc=%257B%2522request%255Fid%2522%253A%25225adcdcb7835209ede23363899603af1d%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=5adcdcb7835209ede23363899603af1d&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-4-132024431-null-null.142%5ev101%5epc_search_result_base1&utm_term=%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0%EF%BC%9A%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C&spm=1018.2226.3001.4187)

感知机是由两层神经元所构成。输入层接收输入信号之后传输给输出层，输出层即阈值逻辑单元（MP神经元）。

[图示

低可信度描述已自动生成](https://blog.csdn.net/weixin_39910711/article/details/100775918?ops_request_misc=&request_id=&biz_id=102&utm_term=%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0%EF%BC%9A%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-3-100775918.142%5ev101%5epc_search_result_base1&spm=1018.2226.3001.4187)

给定训练数据集，权重w以及阈值可以通过学习得到。为设置的学习率。

神经元经过激活函数处理，即为功能神经元。

输出层与输入层之间的一层神经元，被称为隐层或隐含层。

神经网络中，每层神经元与下一层神经元互连，神经元不存在同层连接，也不存在跨层连接，这样的神经网络叫做“多层前馈神经网络”。

[地图上有字

描述已自动生成](https://blog.csdn.net/weixin_48136911/article/details/132024431?ops_request_misc=%257B%2522request%255Fid%2522%253A%25225adcdcb7835209ede23363899603af1d%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=5adcdcb7835209ede23363899603af1d&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-4-132024431-null-null.142%5ev101%5epc_search_result_base1&utm_term=%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0%EF%BC%9A%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C&spm=1018.2226.3001.4187)

# 2025.1.13

* 误差逆传播算法

多层网络的学习能力更加强大，预训练多层网络，需要更加强大的学习算法。误差逆传播（error BackPropagation，BP）算法是其中杰出代表，现实任务中使用神经网络时，大多都是使用BP算法进行训练，同时不仅适用于多层前馈神经网络，还可适用于其他类型的神经网络。

**BP算法的核心思想是通过计算网络输出与目标输出之间的误差，然后将误差从输出层向输入层逐层反向传播，以此来调整网络中各连接权重和偏置，使网络输出逐渐逼近目标输出，从而实现对神经网络的训练。**

[图表, 雷达图

描述已自动生成](https://blog.csdn.net/weixin_48136911/article/details/132024431?ops_request_misc=%257B%2522request%255Fid%2522%253A%25225adcdcb7835209ede23363899603af1d%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=5adcdcb7835209ede23363899603af1d&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-4-132024431-null-null.142%5ev101%5epc_search_result_base1&utm_term=%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0%EF%BC%9A%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C&spm=1018.2226.3001.4187)

白板上的文字

描述已自动生成

|  |  |
| --- | --- |
| **标准BP算法** | 每次仅针对一个训练样例更新连接权和阈值 |
| **累积BP算法** | 基于累积误差最小化 |

BP算法具有强大的表示能力，但也经常**过拟合**：训练误差降低，但是测试误差增加。两种策略缓解过拟合：

|  |  |
| --- | --- |
| **早停** | 将数据分成训练集和测试集，训练集用来计算梯度、更新连接权和阈值，验证集用来估计误差，若训练集误差降低，但验证集误差升高，则停止训练，同时返回最小验证集误差的连接权和阈值。 |
| **正则化** | 在误差目标函数中增加一个用于描述网络复杂度的部分 |

* 全局最小和局部最小

基于梯度的搜索是使用最为广泛的参数寻优方法。在此类方法中，从某些初始解出发，迭代寻找最优参数值。每次迭代中，计算误差函数在当前点的梯度，然后根据梯度确定搜索方向。负梯度方向是函数值下降最快的方向，因此梯度下降法就是沿着负梯度方向搜索最优解。

如果误差函数仅有一个局部极小，那么找的的局部极小就是全局极小；否则则不能保证找到的解是全局最小。

在现实任务中，常用下面策略试图“跳出”局部极小，从而进一步接近全局最小：

|  |
| --- |
| 以多组不同参数初始化多个神经网络，按标准方法训练后，取其中误差最小的解作为最终参数。 |
| 使用“模拟退火”技术，模拟退火在每一步都以一定概率接受比当前解更差的结果，从而有助于“跳出”局部极小。 |
| 使用**随机梯度下降**。与标准梯度下降精确计算梯度不同，随机梯度下降法在计算梯度时加入了随机因素，即使陷入局部极小点，它计算出的梯度仍可能不为零，这样就有机会跳出局部极小继续搜索。 |

除此之外，遗传算法也常用来训练神经网络以更好地逼近全局最小。**上述跳出局部极小地技术大多是启发式。**

* **深度学习**

参数越多地模型复杂度越高、“容量”越大，能够完成更复杂的学习任务，但训练效率低，易陷入过拟合。随着云计算、大数据时代的到来，可以缓解模型训练的低效性，此时，以“深度学习”为代表的复杂模型开始受到关注。

典型的深度学习模型就是很深层的神经网络。可以**增加隐层的数目**或**隐层神经元的数目**，显示增加隐层的数目比增加隐层神经元的数目更有效，因为**增加隐层数不仅增加了拥有激活函数的神经元数目，还增加了激活函数嵌套的层数。多隐层神经网络难以直接用经典算法（BP算法）进行训练，因为误差在多隐层内逆传播时，往往会“发散”而不能收敛到稳定状态。**

无监督逐层训练是多隐层网络训练的有效手段：每次训练一层隐结点，训练时将上一层隐结点的输出作为输入，而本层隐结点的输出作为下一层隐结点的输入。这也叫做“**预训练**”。

在预训练全部完成后，再对整个网络进行“微调”训练。

“预训练＋微调”可视为将大量参数分组，对每组先找到局部看起来比较好的设置，然后再基于这些局部较优的结果联合起来进行全局寻优。

另一种节省训练开销的是“**权共享**”：让一组神经元使用相同的连接权。

*另一个角度理解深度学习：多层堆叠、每层对上一层的输出进行处理的机制，看作是在对输入信号进行逐层加工，从而把初始的、与输出目标之间联系不太密切的输入表示，转化成与输入目标联系更密切的表示，使得原来仅基于最后一层输出映射难以完成的任务转为可能。*

# 2025.1.16

支持向量机（Support Vector Machine）是一种二类分类模型，其基本模型定义为特征空间上的间隔最大的广义线性分类器，其学习策略便是间隔最大化，最终可转化为一个**凸二次规划问题**的求解。

[图表, 散点图

描述已自动生成](https://blog.csdn.net/qq_42192693/article/details/121164645?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522e4298b29c352e13a7e8e8dbef62f1a70%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=e4298b29c352e13a7e8e8dbef62f1a70&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-121164645-null-null.142%5ev101%5epc_search_result_base1&utm_term=%E6%94%AF%E6%8C%81%E5%90%91%E9%87%8F%E6%9C%BA&spm=1018.2226.3001.4187)

假设两类数据可以被分离，垂直于法向量w，移动H直到碰到某个训练点，可以得到两个超平面H1和H2，两个平面称为支撑超平面，题目分别支撑两类数据。而位于H1和H2正中间的超平面是分离这两类数据的最好选择。**支持向量**就是离分隔超平面最近的那些点。

[图示

低可信度描述已自动生成](https://blog.csdn.net/qq_42192693/article/details/121164645?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522e4298b29c352e13a7e8e8dbef62f1a70%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=e4298b29c352e13a7e8e8dbef62f1a70&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-121164645-null-null.142%5ev101%5epc_search_result_base1&utm_term=%E6%94%AF%E6%8C%81%E5%90%91%E9%87%8F%E6%9C%BA&spm=1018.2226.3001.4187)

在求解最优化问题中，**拉格朗日乘子法**（Lagrange Multiplier）和**KKT**（Karush Kuhn Tucker）**条件**是两种最常用的方法。在有等式约束时使用拉格朗日乘子法，在有不等约束时使用KKT条件。

* 拉格朗日乘子法

拉格朗日乘子法是一种寻找多元函数在一组约束下的极值的方法。通过引入拉格朗日乘子，可将有d个变量与k个约束条件的最优化问题转化为具有d+k 个变量的无约束优化问题求解。

[文本, 信件

描述已自动生成](https://www.cnblogs.com/mo-wang/p/4775548.html)

[文本, 信件

描述已自动生成](https://www.cnblogs.com/mo-wang/p/4775548.html)

[图形用户界面, 文本, 应用程序

描述已自动生成](https://www.cnblogs.com/mo-wang/p/4775548.html)

* 二次规划

二次规划是一类典型的优化问题，包括凸二次优化和非凸二次优化。在此类问题中，目标函数是变量的二次函数，而约束条件是变量的线性不等式。

# 2025.1.18

硬分隔SVM

图表, 散点图

描述已自动生成

文本

描述已自动生成

文本, 信件

描述已自动生成

* **序列最小优化（Sequential Minimal Optimization，SMO）**

序列最小优化是将大优化问题分界成多个小优化问题来求解。

**SMO算法工作原理**：每次循环中选择两个变量进行优化处理。一旦找到一对合适的变量，那么就增大其中一个同时减小另一个。这里的“合适”指的是两个变量必须要符合一定的条件，条件之一就是这两个变量必须要在间隔边界之外，而其第二个条件则是这两个变量还没有进行过区间化处理或者不在边界上。

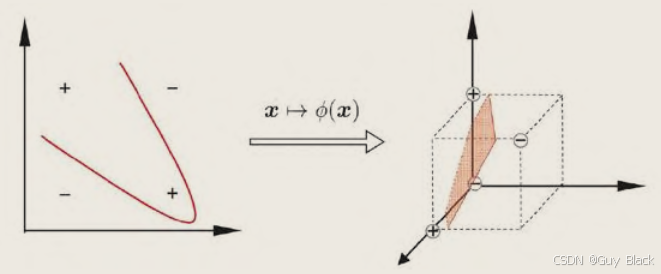
文本, 信件

描述已自动生成

# 2025.1.19

* **核函数与软间隔**

若原始样本不是**线性可分**的，可将样本从原始空间映射到一个更高维的特征空间，使得样本在这个特征空间内线性可分。

[](https://blog.csdn.net/weixin_61986755/article/details/142692095?ops_request_misc=&request_id=&biz_id=102&utm_term=%E6%94%AF%E6%8C%81%E5%90%91%E9%87%8F%E6%9C%BA%20%E5%91%A8%E5%BF%97%E5%8D%8E&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-1-142692095.142%5ev101%5epc_search_result_base1&spm=1018.2226.3001.4187)

设想函数：

即x\_i 与 x\_j 在特征空间的内积等于它们在原始空间中通过函数K(·,·)计算的结果。这里K(·,·)就是“核函数”。只要一个对称函数所对应的核矩阵半正定（，总能找到一个与之对应的映射f，核函数选择不唯一。

|  |  |  |
| --- | --- | --- |
| 名称 | 表达式 | 参数 |
| 线性核 |  |  |
| 多项式核 |  | D大于等于1，为多项式的次数 |
| 高斯核 |  | Sigma大于0，为高斯核的带宽 |
| 拉普拉斯核 |  | Sigma大于0 |
| Sigmoid核 |  | Tanh为双曲正切函数，beta 大于0，theta小于0 |

常用核函数

还可以通过函数组合得到。

**软间隔**：这主要是缓解近似线性可分与过拟合的手段。在特征空间或样本空间做SVM时，其实很难出现完全线性可分的情形，此时我们不能断言它是线性不可分的，大多数情形可能是近似线性可分。而我们若强行升维核函数处理，有可能导致过拟合现象。此时，我们就应当允许一些点跨入间隔区域。

[图表, 散点图

描述已自动生成](https://blog.csdn.net/weixin_61986755/article/details/142692095?ops_request_misc=&request_id=&biz_id=102&utm_term=%E6%94%AF%E6%8C%81%E5%90%91%E9%87%8F%E6%9C%BA%20%E5%91%A8%E5%BF%97%E5%8D%8E&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-1-142692095.142%5ev101%5epc_search_result_base1&spm=1018.2226.3001.4187)

优化目标为：

手机屏幕的截图

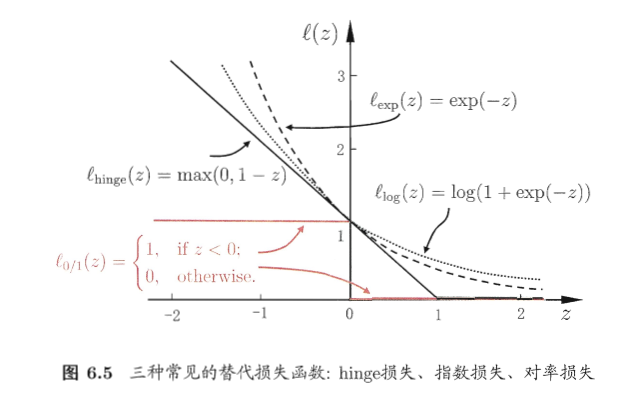
中度可信度描述已自动生成

由于0/1损失函数非凸非连续，使得上式不宜求解。一般使用其他函数来代替，这称为“替代损失“，替代损失函数具有良好的数学性质。

Hinge损失：

指数损失：

对率损失：



**正则化**通常通过在优化目标函数中加入一个正则项来实现。这个正则项通常是模型参数的某种范数，例如L2范数（也称为岭回归）或L1范数（也称为Lasso）。

手机屏幕的截图

描述已自动生成

其中，*w*是模型的权重向量，*b*是偏置项，*ξi*​是松弛变量，用于处理不可分的情况，*C*是正则化参数，控制正则项和经验风险之间的平衡。

# 2025.2.3

* **贝叶斯决策论**

贝叶斯决策论是概率框架下实施决策的基本方法,对分类任务来说,在所有相关概率都已知的理想情形下,贝叶斯决策论考虑如何基于这些概率和误判损失来选择最优的类别标记。

贝叶斯定理是贝叶斯决策论的基础，描述了如何根据新的证据更新先验概率，**贝叶斯定理**：

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 后验概率 | P(A|B) | 在观测B的条件下A的概率 | 指在观测到数据之后，对某个假设的概率估计。它是通过贝叶斯定理计算得到的，结合了先验概率和似然概率。（**事情已经发生，求这件事情发生的原因是由某个因素引起的可能性的大小。**） |
| 似然概率 | P(B|A) | 在A发生的条件下B的概率 | 指在某个假设下，观测数据的概率。**它描述了在某个假设下，数据出现的可能性**。 |
| 先验概率 | P(A) | 没有观测到B的条件下A的概率 | 指在没有观测到任何数据之前，对某个事件的概率估计。它通常基于先验知识或经验。（**根据以往经验或经过数据统计得到的概率。**） |
| 边际概率 | P(B) | B的总概率 |  |

贝叶斯决策论通过结合先验知识和观测数据，使用贝叶斯定理计算后验概率，从而做出最优决策。即：

P(c)是样本空间中各类样本所占的比例，根据**大数定理**，当训练集包含充足的独立同分布样本时，P(c)可以通过各类样本所占的比例来进行估计。

但是对于类条件概率P(x|c)来说，涉及了关于x所有属性的联合概率，因此很难进行估计。

例如：每个样本具有d个属性，每个属性都有10个属性值，那么样本空间将有种可能值，导致**组合爆炸**。

**朴素贝叶斯对条件概率分布做了条件独立性假设**

即，现在种可能值。

先验概率：

**属性值是离散情况下**，条件概率为：

**属性值是连续情况下**，条件概率为：

假设概率密度P(x|c)服从正太分布，通过极大似然估计得到的正态分布均值就是样本均值，方差就是。

例子：



基于后验概率P( c\_i | x)可以获得样本x分类为c\_i所产生的期望损失（条件风险）

**任务就是寻找一个判定准则最小化所有样本的条件风险总和**，因此就有了贝叶斯判定准则：为最小化总体风险，只需要在每个样本上选择那个条件风险最小的类标。

* “判别式”模型和“生成式”模型

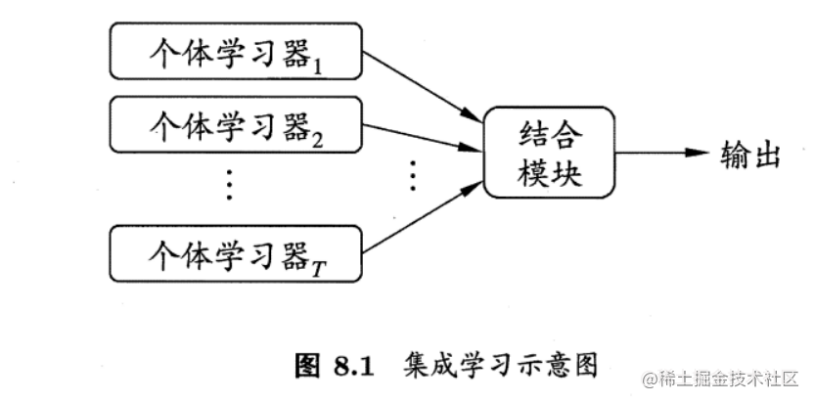
“判别式“模型：给定x，可通过直接建模P(c | x)来预测c。例如：决策树、BP神经网络、支持向量机。

“生成式“模型：先对联合概率分布P(x, c)建模，然后由此得到P(c | x)。

*“判别式“是根据特定数据集进行处理，得到模型；”生成式“是根据特定数据集找到所隐含的分布，根据分布进而找到模型。*

# 2025.2.5

**集成学习**（ensemble learning）：通过构建多个学习器来完成学习的任务。可以分成同质集成/异质集成。同质集成（homogeneous）：个体学习器都是同种类型的。该类型中个体学习器称为“基学习器”。异质集成（heterogeneous）：包含不同类型的个体学习器。该类型中个体学习器称为“组件学习器”。



集成方法主要包括Bagging和Boosting，Bagging和Boosting都是将已有的分类或回归算法通过一定方式组合起来，形成一个更加强大的分类。两种方法都是把若干个分类器整合为一个分类器的方法，只是整合的方式不一样，最终得到不一样的效果。常见的基于Baggin思想的集成模型有：随机森林、基于Boosting思想的集成模型有：Adaboost、GBDT、XgBoost、LightGBM等。

根据**个体学习器的生成方式**，目前的集成学习方法大致可分为两大类：

个体学习器间存在强依赖关系、必须串行生成的序列化方法：Boosting。

个体学习器间不存在强依赖关系、可同时生成的并行化方法：Bagging和“随机森林”(Random Forest)。

**Boosting**是一族可将弱学习器提升为强学习器的算法。这族算法的工作机制类似：先从初试训练集训练出一个基学习器，再根据基学习器的表现对训练样本分布进行调整，使得先前基学习器做错的训练样本在后续受到更多关注，然后基于调整后的样本分布来训练下一个基学习器。

为了得到泛化性能强的集成，集成中的个体学习器应尽可能相互独立，但是“独立”在现实任务中很难做到，不过可以使基学习器尽可能具有较大差异。另一种可能的做法是对训练样本采样，产生出若干个不同的子集，再从每个数据子集中训练出一个基学习器。为此可以采用相互有交叠的采用子集，这就是**Bagging**。**随机森林**是Bagging的一个扩展变体，随机森林在以决策树为基学习器构建的Bagging集成的基础上，进一步在决策树的训练过程中引入了随机属性选择。

Baggin和Boosting的区别总结如下：

|  |  |  |
| --- | --- | --- |
|  | **Boosting** | **Bagging** |
| 样本选择 | 需要每一轮的训练集不变，只是训练集中每个样本在分类器中的权重发生变化。而权值是根据上一轮的分类结果进行调整 | 训练集是从原始集中有放回的选取，所以从原始集中选出的各轮训练集之间是独立的 |
| 样例权重 | 根据错误率不断调整样本的权值，错误率越大则权重越大 | 使用均匀取样，所以每个样本的权重相等 |
| 预测函数 | 每个弱分类器都有相应的权重，对于分类误差小的分类器会有更大的权重 | 所有预测函数的权重相等 |
| 并行计算 | 各个预测函数只能顺序生成，因为后一个模型参数需要前一轮模型的结果。 | 各个预测函数可以并行生成 |

# 2025.2.7

* **聚类**

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规律，为进一步的数据分析提供基础，较为经典的是聚类。

**聚类试图将数据集中的样本划分为若干个通常是不相交的子集，每个子集称为一个“簇”。**聚类既能作为一个单独过程，用于找寻数据内在的分布结构，也可以作为分类等其他学习任务的前驱过程。

距离计算：

连续属性及离散有序属性：闵可夫斯基距离：

无序属性：VDM（Value Difference Metric）

**K均值算法（k-means）**：对于给定的样本集，按照样本之间的距离大小，将样本集划分为K个簇。让簇内的点尽量紧密的连在一起，而让簇间的距离尽量的大。给定样本集D，k-means算法针对聚类所得簇划分C最小化平方误差：

K-Means聚类算法步骤实质是EM算法（最大期望算法)的模型优化过程，具体步骤如下：

1. 随机选择k个样本作为初始簇类的均值向量
2. 将每个样本数据集划分到离它距离最近的簇
3. 根据每个样本所属的簇，更新簇类的均值向量
4. 重复（2）（3）步，当达到最大迭代次数或簇类的均值向量不再改变

# 2025.2.8

**K 邻近算法（K-Nearest Neighbors，简称 KNN）**是一种经典的机器学习算法，主要用于分类和回归任务。它的核心思想是：给定一个新的数据点，通过查找训练数据中最接近的 K 个邻居，并根据这些邻居的标签来预测新数据点的标签。

KNN 的优点在于其简单易懂、无需训练过程，并且适用于大多数任务。它能够处理复杂的非线性问题，不依赖数据分布假设，能够很好地适应复杂的决策边界。

然而，KNN 的缺点也很明显。它的计算开销大，因为每次预测都需要计算所有训练数据的距离，导致在大数据集上表现不佳。此外，KNN 需要存储所有训练数据，占用较大的内存空间，并且对异常值敏感，可能会影响预测结果的准确性。

KNN算法步骤：

1. 选择 K 个邻居的数量，K 值通常是一个奇数，以避免平票的情况。

2. 计算待预测数据点与训练数据集中每个点的距离。

3. 根据计算出的距离选择 K 个最接近的点。

4. 对于分类任务，返回 K 个邻居中最多的类别；对于回归任务，返回 K 个邻居标签的均值。

# 2025.2.10

降维是指在保持数据主要特征的情况下，减少数据的特征维度，从而提高计算效率、降低存储成本并减少维度灾难（Curse of Dimensionality）对模型的影响。

**主成分分析（PCA）是一种无监督的线性降维方法；监督线性降维方法最著名的是线性判别法（LDA）。**

|  |  |  |
| --- | --- | --- |
|  | **主成分分析（PCA）** | **线性判别分析（LDA）** |
| 核心 | 找到数据的主成分（方差最大的方向），然后对数据进行投影，从而降低维度。 | 利用类别标签来优化数据投影，使得不同类别的样本尽可能分开。 |
| 步骤 | 1. 将数据中心化（去均值）并标准化，以消除量纲影响。 2. 衡量不同特征之间的相关性。 3. 找到协方差矩阵的特征值和特征向量，特征向量代表数据的主方向，特征值表示该方向上的方差信息量。 4. 根据特征值大小选择前K KK个特征向量构成新的特征空间。 5. 用选择的特征向量将原始数据投影到新的低维空间。 | 1. 计算类内散度矩阵SW 2. 计算类间散度矩阵SB 3. 求解投影方向 4. 选择前k个特征向量组成投影矩阵W |

# 2025.2.11

特征选择是重要的“数据预处理”过程，在现实机器学习任务中，获得数据之后通常先进行特征选择再训练学习器，因为：

1. 维度灾难
2. 去除不相关特征往往会降低学习任务的难度

**降维（使用PCA/LDA）和特征选择是处理高维数据的两大主流技术。**

特征选择的目标是从原始特征集中选择出最优的特征子集，以提高模型的性能、减少计算成本，并降低过拟合风险。其中，“生成与搜索”是一种常见的特征选择方法，它通过不断生成不同的特征子集并进行评估，最终找到最优子集。

生成与搜索：特征选择通常包含搜索空间和评估策略：

**搜索空间**：所有可能的特征组合

**评估策略**：用于衡量某个特征子集的质量，比如分类准确率、均方误差（MSE）、等。

常见的特征选择方法大致分为三类：过滤式、包裹式和嵌入式。

|  |  |  |  |
| --- | --- | --- | --- |
|  | **概念** | **优点** | **缺点** |
| 过滤式 | 在特征选择时**不依赖**于具体的机器学习模型，而是**使用统计指标来衡量**特征的重要性。 | 计算快；不依赖具体模型 | 只考虑了单个特征和目标变量的关系；可能会选到对目标变量影响小但是与其他特征关联大的特征 |
| 包裹式 | 通过**训练机器学习模型来评估**不同特征子集的质量，然后选择最优子集 | 考虑了特征之间的相互关系；适用于特征数量少的情况 | 计算成本高；可能导致过拟合 |
| 嵌入式 | 在**模型训练过程中自动选择**最优特征，即特征选择和模型训练是同时进行的 | 效率高；同时进行特征选择和模型训练，防止了包裹式的过拟合 | 依赖具体模型；需要参数调优 |