课件和书都在汪荣贵的个人主页
http://ci.hfut.edu.cn/2020/1208/c11504a245626/page.htm
https://pan.baidu.com/s/1iaMwelwAQie2-U-xSvc5OA

第一节

2021年1月3日 10:42

这种在同等条件下选择简单事物的倾向性原则称为奥卡姆剃刀原则

逻辑推理——专家系统（知识工程）——机器学习

机器学习具体地说，对于给定的任务和性能度量标准，使用先验信息E，通过某种计算方式T改进初始模型M_0，获得一个性能更好的改进模型M_p，即有：
$M_p = T(M_0, E)$

分类

回归：使模型输出接近真实值

聚类的类别由不同样本之间的某种相似性确定，因而聚类类别所表达的含义通常是不确定的，聚类样本也不带特定的标注表示样本所属的类别。

示例集：在聚类任务中，所有输入示例的集合——不带标注样本，聚类的先验

簇：被划分为同一类别的示例所构成的集合

监督学习

——利用一组带标注样本调整模型参数，提升模型性能的学习方式。

基本思想是通过标注值告诉模型在给定输入的情况下应该输出什么值，由此获得尽可能接近真实映射方式的优化模型。

强化学习

根据反馈信息来调整机器行为以实现自动决策的一种机器学习方式

强化学习主要由智能体和环境两个部分组成。智能体是行为的实施者，由基于环境信息的评价函数对智能体的行为做出评价，若智能体的行为正确，则由相应的回报函数给予智能体正向反馈信息以示奖励，反之则给予智能体负向反馈信息以示惩罚。

整体误差$R_S(f) = E[L(y, f(x))] = \frac{1}{n}\sum_{i=1}^{n} L\left(y_i, f(X_i)\right)$

在与该任务相关的所有样本的集合D上整体误差称为泛化误差

$R_{exp}(f) = E_{P(D)}[L(y, f(X))]$

训练误差或称经验风险 训练集上的整体误差

$$R_{emp}(f) = \frac{1}{n}\sum_{k=1}^{n} L(y_k, f(X_k))$$

根据经验风险最小化方法得到优化模型：

$\hat{f} = \arg_{f\in F} \min R_{\text{emp}}(f)$

测试误差 测试集上整体误差

$$R_{test} = \frac{1}{v}\sum_{k=1}^{v} L(y_k^t, f(X_k^t))$$

模型的学习能力或模型的容量：机器学习模型这种适应训练数据变化的能力。

使用模型输出在不同训练样本集合下的综合偏差对其进行度量，这种综合偏差称为模型输出的偏差，简称为偏差。

$\text{bias}[F(X)] = E[F(X)] - y$

泛化误差：

$R_{exp}(f) = E[L(y, F(X))] = E\left[(F(X) - y)^2\right]$

$= \text{var}[\text{F}[(\text{X})] + [\text{bias}(\text{F}(\text{x}))]^2$

过拟合：同时拟合训练样本的共性特征和个性特征导致模型泛化能力较弱

欠拟合：未能充分拟合训练样本共性特征造成模型泛化误差较大而导致模型泛化能力较弱

专家系统弊端

普适性差，专家主观性、错误、意见不一致

决策树

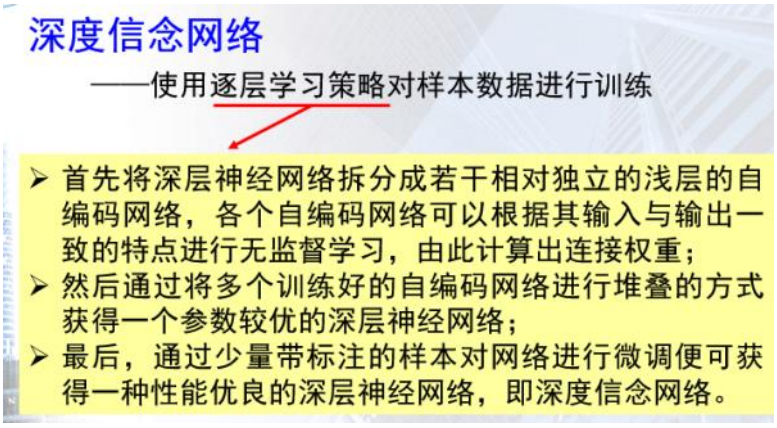
符号学习——归纳学习

SVM——统计学习 核方法 低维线性不可分映射到高维线性可分

优点

最大间隔思想使得分类器模型只取决于支持向量，模型计算复杂度只与支持向量数目有关，有效避免了维数灾难问题并使得支持向量机对训练样本的变化具有较强的鲁棒性。

支持向量机的核方法在一定程度上避免了直接在高维空间中处理问题，有效降低了问题求解的难度



特征提取

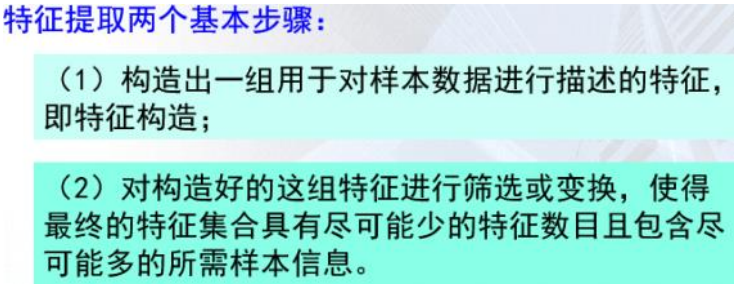
从样本中学习。对于任意一个给定的样本对象ξ，一般需要对其提取若干属性形成对该样本的数据描述或表征，并将这些属性值作为机器学习模型的输入。令：

$$x_1 = \psi_1(\xi), x_2 = \psi_2(\xi), \dots, x_m = \psi_m(\xi)$$

为样本ξ的m个属性提取函数，则可通过这些函数将样本ξ映射成一个m元**表征向量**X，即：

$$X = X(\xi) = (x_1, x_2, \dots, x_m)^T$$

其中x_i为样本ξ的第i个属性值，$i = 1,2,\dots,m$。



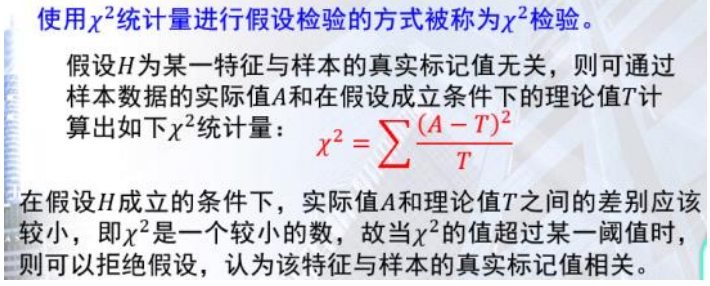
传统特征 词袋词频LBP特征、Canny特征、颜色直方图、Haar特征、SIFT特征等。

深度学习特征：卷积网络

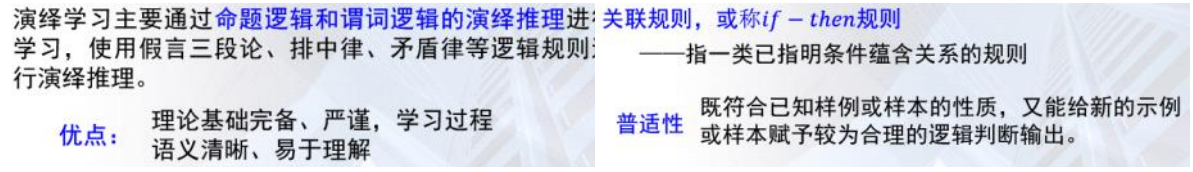
特征选择：子集搜索或相关性评估

相关性评估常用的统计量有χ^2统计量、信息熵等

相关性评估方法首先假设某一特征与样本的真实标记值无关，然后对该假设进行假设检验，判断该假设是否成立



规则构建



行演绎推理。

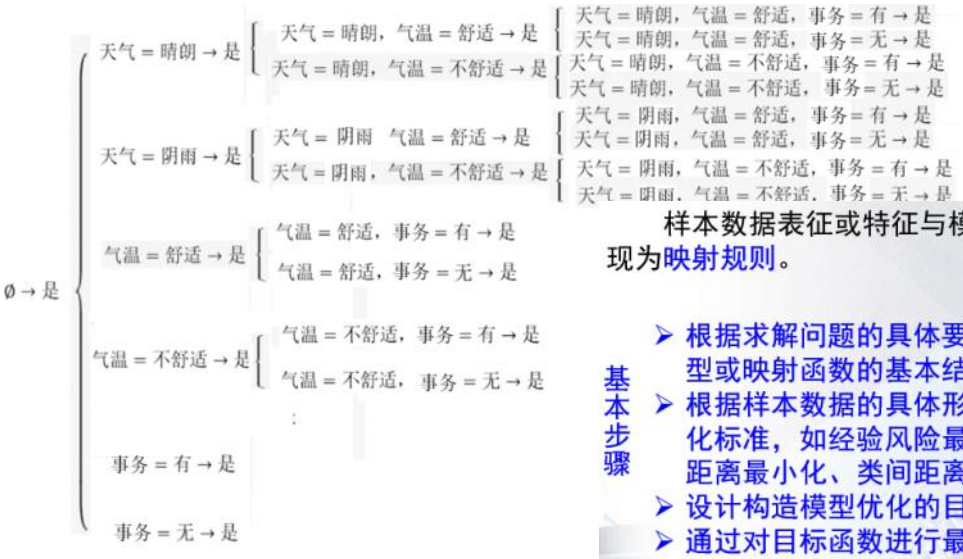
优点：理论基础完备、严谨，学习过程语义清晰、易于理解

缺点：难以处理不确定性信息，对复杂问题的求解会出现难以解决的组合爆炸问题

普适性：既符合已知样例或样本的性质，又能给新的示例或样本赋予较为合理的逻辑判断输出。

$X \to Y$表示一个具体的关联规则，意为如果命题X成立，则命题Y成立。
其中X称为前件或条件，Y称为后件或结论。

序列覆盖算法递归地归纳出单条关联规则去逐步覆盖训练样本集中的正样例，当训练样本集中所有正例均被已归纳的关联规则所覆盖时，此时对应的关联规则集就是所求规则集。然后按适当标准对所求规则进行排序，确定规则使用的优先级。



样本数据表征或特征与模型输出之间的关系通常表现为映射规则。

基本步骤：

- 根据求解问题的具体要求确定机器学习模型的基本类型或映射函数的基本结构；
- 根据样本数据的具体形式和模型特点确定合适模型优化标准，如经验风险最小化、结构风险最小化、类内距离最小化、类间距离最大化等；
- 设计构造模型优化的目标函数；
- 通过对目标函数进行最值优化获得所需映射函数，完成映射规则构建。

结构化风险：就是正则化

模型评估

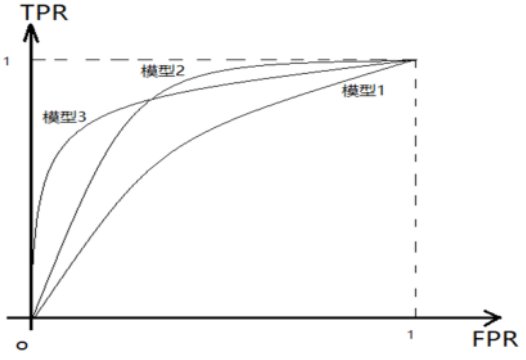
正确率错误率

查准率 精确率 P=TP/TP+FP

查全率 召回率 W=TP/TP+FN

F1值，查全率和查准率调和平均

$$\mathrm{F1} = \frac{1}{\frac{1}{P} + \frac{1}{W}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{NP}}$$



ROC曲线下方面积指标称为AUC指标。在一般情况下，模型所对应的AUC值越大，则该模型的平均性能就越好。

回归

MSE

决定系数R

$$R^2 = 1 - \frac{\sum_{i=1}^{s}\left(y_i - f(X_i)\right)^2}{\sum_{i=1}^{s}(y_i - \bar{y})^2}$$

自助法：当D中样本数量较少

——通过对D中样本进行可重复随机采样的方式构造训练集和测试集

假设数据集D中包含n个样本，自助法对数据集D中样本进行n次有放回的采样，并将采样得到的样本作为训练样本生成一个含有n个样本的训练样本集S，所有未被得到的样本则作为测试样本构成测试集T。

对于D中的任一样本，该样本在自助采样中未被采样的概率为：

$$\lim_{n\to\infty}\left(1-\frac{1}{n}\right)^{n}=\frac{1}{e}\approx 0.368$$

第二节

2021年1月3日 13:21

最大后验

bayes公式 $f(\boldsymbol{\beta}|\mathrm{X})=\frac{f(X|\boldsymbol{\beta})g(\boldsymbol{\beta})}{p(X)}$

$f(X|\boldsymbol{\beta})$：现有样本所表现出的信息

$g(\boldsymbol{\beta})$：先验信息

$f(\boldsymbol{\beta}|\mathrm{X})$：后验信息

最大似然估计：认为待求参数是某个固定取值；仅考虑当前的样本出现情况。

最大后验估计：认为待求参数服从某一概率分布；同时考虑过往经验和已经出现的样本，过往经验表现为先验概率。

最大后验估计形式为：

$\widehat{\boldsymbol{\beta}}=\arg_{\boldsymbol{\beta}}\max f(\boldsymbol{\beta}|\mathrm{X})=\arg_{\boldsymbol{\beta}}\max[f(\mathrm{X}|\boldsymbol{\beta})g(\boldsymbol{\beta})/p(X)]$

$p(X)$为参数无关且恒大于零，可省略，得：

$\widehat{\boldsymbol{\beta}}=\arg_{\boldsymbol{\beta}}\max f(\mathrm{X}|\boldsymbol{\beta})g(\boldsymbol{\beta})$

正定二次型梯度下降

当选择最优步长时，每步搜索方向均与上步搜索方向正交，

$X_{k+1}=X_k-\frac{{P_k}^T P_k}{{P_k}^T \boldsymbol{A} P_k}P_k$

梯度下降法缺陷之一：靠近极小值时收敛速度通常会减慢

共轭梯度下降法基本思想：对搜索方向进行修正

共轭的概念：设$\boldsymbol{A}$为$R^{n\times n}$上对称正定矩阵，Q_1,Q_2为R^n上两个非零向量，若有${Q_1}^T\boldsymbol{A}Q_2=0$，则称$Q_1$与$Q_2$关于矩阵$\boldsymbol{A}$**共轭**，向量$Q_1$与$Q_2$的方向为一组**共轭方向**。

第一次更新：

选取初始点X_1，计算目标函数在该点梯度值$\nabla F(X_1)$；

根据$X_{k+1}=X_k+\alpha_k P_k$计算下一点X_2；

步长α_k为

$\mathrm{argmin}_{\alpha\geq 0}F(X_k+\alpha_k P_k)$

的优化值；

第二次更新：

搜索到X_2后，计算该点对应的梯度值$\nabla F(X_2)$，并按下式调整搜索方向：

$$P_{k+1}=-\nabla F(X_{k+1})+step_k P_k$$

$step_k$为调整搜索方向时的步长，上式两侧同时乘以$\boldsymbol{A}P_k$可得：

$$P_{k+1}^T\boldsymbol{A}P_k=-\nabla F(X_{k+1})\boldsymbol{A}P_k+step_k P_k^T\boldsymbol{A}P_k$$

当P_{k+1}和P_k共轭时可得：

$$step_{k+1}=\frac{\nabla F(X_{k+1})\boldsymbol{A}P_k}{{P_k}^T\boldsymbol{A}P_k}$$

最速下降法的迭代方向为什么相互垂直

证明：

最速下降法的迭代公式：

$$x_{k+1} = x_k - \lambda \Delta f(x)$$

其中，迭代步长 lambda 由一维搜索得到，lambda 满足

$$\lambda = \arg \min_{\lambda} f(x_k - \lambda \Delta f(x))$$

一阶导数为 0 时成立，即

$$\frac{\partial f(x_k - \lambda \Delta f(x))}{\partial \lambda} = -\Delta f(x_k - \lambda \Delta f(x)) \Delta f(x) = -\Delta f(x_{k+1}) \Delta f(x) = 0$$

第三章

2021年1月3日　　23:55

决策树

ID3　经验熵越大越杂乱

$$H(D|A)=\sum_{i=1}^{m}\frac{|D_i|}{|D|}H(D_i)$$

经验条件熵

$$H(D|A)=\sum_{i=1}^{m}\frac{|D_i|}{|D|}H(D_i)$$

信息增益

$$G(D,A)=H(D)-H(D|A)$$

若信息增益值越大则表示使用属性A划分后的样本集合纯度提升越大，使得决策树模型具有更强的分类能力。

C4.5 信息增益率

信息增益率在信息增益基础上引入一个校正因子，消除属性取值数目的变化对计算结果的干扰。

信息增益率定义：

$$G_r(D,A)=\frac{G(D,A)}{Q(A)}$$

其中$Q(A)$为校正因子，由下式计算：

$$Q(A)=-\sum_{i=1}^{m}\frac{|D_i|}{|D|}log_2\frac{|D_i|}{|D|}$$

属性A的取值状态数m值越大，则$Q(A)$值也越大，可以减少信息增益的不良偏好对决策树模型构建所带来的影响，使得所建决策树拥有更强的泛化能力。

Cart Gini系数

$$\text{Gini}(D)=1-\sum_{k=1}^{m}\left(\frac{|C_k|}{|D|}\right)^2$$

其中C_k是D中属于第k类的样本子集，m是类别数。

ID3算法：以信息增益最大的属性为分类特征，基于贪心策略自顶向下地搜索遍历决策树空间，通过递归方式构建决策树。

ID3算法缺点：不能处理连续值和缺失值。

解决方法：C4.5

避免过拟合：剪枝：从决策树上裁剪掉一些子树或者叶子结点，并将其根节点或父节点作为新的叶结点，从而简化分类树模型。

决策树剪枝策略有预剪枝和后剪枝两种

预剪枝：在决策树生成过程中，对每个子集在划分前先进行估计，若当前结点的划分不能带来泛化性能的提升，则停止划分并将当前结点标记为叶结点。

后剪枝：先从训练样本集生成一颗完整决策树，然后自底向上或自上而下考察分支结点，若将该结点对应子树替换为叶结点能提升模型泛化性能，则进行替换。

后剪枝可以保证剪枝操作不会降低决策树模型的泛化性能，因此通常采用后剪枝策略。

悲观错误剪枝（PEP）算法的基本思想：

1、考察每个内部结点对应子树所覆盖训练样本的误判率与剪去该子树后所得叶结点对应训练样本的误判率；

2、比较二者间的大小关系确定是否进行剪枝操作；

3、当剪枝后所获得叶子节点的误判率小于所对应的子树误判率时，进行剪枝。

注：由于PEP算法直接通过训练样本集对子树和叶结点进行评估，导致子树的误判率一定小于叶结点的误判率，得到在任何情况下都无需剪枝的错误结论。为避免出现这种错误结论，在计算误判率时添加了一个经验性惩罚因子。

ID3算法和C4.5算法：构建的决策树模型都属于分类树；

CART算法：既可构造的分类树，也可构造回归树。

CART算法构造的决策树必是一颗二叉树，因此，对于特征为多余两个取值的需要进行二元划分。如下所示：

则在属性A为划分属性的条件下，集合D的基尼指数定义为：

$$\text{Gini(D, A)} = \frac{|D_1|}{|D|}\text{Gini}(D_1) + \frac{|D_2|}{|D|}\text{Gini}(D_2)$$

基尼系数找小的！！

CART算法构造回归决策树：

决策树f在这两个区域上整体误差平方和为：

$$\sum_{X_i \in l_1}(y_i - \hat{y}_1)^2 + \sum_{X_i \in l_2}(y_i - \hat{y}_2)^2$$

目标选择不同的切分变量和切分点对上述整体误差平方和进行优化：

$$\min_{x_j, b}\left(\sum_{X_i \in l_1}(y_i - \hat{y}_1)^2 + \sum_{X_i \in l_2}(y_i - \hat{y}_2)^2\right)$$

过拟合问题：CART算法使用代价复杂度剪枝（CCP）方法。

$$R_\alpha(T) = R(T) + \alpha|T|$$

CCP方法的步骤：

首先：对T_0中每一内部结点t计算$\alpha(t)$值；

然后：在T_0中剪去$\alpha(t)$最小的T_t并将所得子树记为T_1，同时将$\alpha(t)$的最小值记为α_1

最后：递归上述过程，直至得到根结点构成的单节点子树。

在这一过程中，得到一个取值递增的参数序列$\{\alpha_1, \alpha_2, \cdots, \alpha_n\}$和子树序列$\{T_0, T_1, \cdots, T_n\}$

使用独立测试集从$\{T_0, T_1, \cdots, T_n\}$中选取最优子树$T_\alpha$

贝叶斯

事件A机器学习任务中样本的取值状态为X 事件B机器学习模型参数θ的取值为θ_i

公式可化为：$P(\theta_i|X) = P(\theta_i)P(X|\theta_i)/P(X)$

根据全概率公式可以得到概率$P(X)$：$P(X) = \sum_k P(X|\theta_k)P(\theta_k)$

贝叶斯记不住了不写了

构造贝叶斯分类器步骤：

1、计算各种情况下的误判损失值λ_{ij}，建立混淆矩阵；

2、计算训练样本X被分为不同类别的条件风险$R(y_i|X)$；

3、最小化每个训练样本条件风险$R(y_i|X)$的方式构建分类模型。

示：

CART算法构造回归决策树：

某个特征x_j的取值b作为切分点，则训练子集划分为：

$$l_1(x_j, b) = \{X|x_j \le b\} \quad l_2(x_j, b) = \{X|x_j > b\}$$

朴素贝叶斯分类器条件独立性假设：认为样本的每个特征之间是相互独立的，不存在依赖关系。

半朴素贝叶斯分类器 独依赖估计(ODE) 假设样本的每个属性都可单独依赖且仅依赖另外一个属性，或者说样本的每个属性都可关联且仅关联一个对其产生一定影响的另一属性
关键：如何确定每个属性的依赖属性。

SPODE方法：指定某个属性是所有其它属性的依赖属性，被指定依赖属性名为超父。

SPODE实现思路：首先分别让每个属性当一次超父，然后通过带标签的训练集找出使得预测误差最小的超父，并将其作为所求模型的超父。

贝叶斯回归

支持向量机

SVM模型：分离超平面使得两类样本数据与该分离超平面形成的间隔均为最大。SVM只输出样本类别而不输出样本属于某一类别的概率。

几何间隔：对于给定的数据集 T 和超平面 $w \cdot x + b = 0$ ，定义超平面关于样本点 (x_i, y_i) 的几何间隔为

$$\gamma_i = y_i \left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} \cdot \boldsymbol{x_i} + \frac{b}{\|\boldsymbol{w}\|} \right)$$

超平面关于所有样本点的几何间隔的最小值为

$$\gamma = \min_{i=1,2\ldots,N} \gamma_i$$

实际上这个距离就是我们所谓的**支持向量**到超平面的距离。

根据以上定义，SVM模型的求解最大分割超平面问题可以表示为以下约束最优化问题

$$\max_{\boldsymbol{w},b} \ \gamma$$

$$s.t. \quad y_i \left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} \cdot \boldsymbol{x_i} + \frac{b}{\|\boldsymbol{w}\|} \right) \geq \gamma \, , i = 1, 2, \ldots, N$$

$\min_{\boldsymbol{w},b} \frac{1}{2}||\boldsymbol{w}||^2$； $\mathrm{s.t.}\, y_i(\boldsymbol{w}^T X_i + b) - 1 \geq 0$注意这里最小化$w^2$与最大化$\frac{1}{w}$等价

SMO算法较复杂每次选择两个参数α_i和α_j进行优化，在完成当前参数优化计算之后再重新选取另外两个参数进行，直至所有参数收敛。

软间隔SVM模型：允许对少量训练样本出现分类错误。引入一个松弛变量ξ_i，将约束条件转化为：$y_i(\boldsymbol{w}^T X_i + b) \geq 1 - \xi_i$

目标函数min $\frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{n}\xi_i \, s.t. \; y_i(\boldsymbol{w}^T X_i + b) \geq 1 - \xi_i$其中$C > 0$是为惩罚因子。

映射函数$\varphi(X)$作用于D中所有样本数据，将D映射到高维空间，使得D在高维空间的映像$D' = \varphi(D)$满足线性可分性

假设线性可分数据集D'的分离超平面为$\boldsymbol{w}^T\varphi(X) + b = 0$

由此可将优化求解SVM模型的目标函数定义为：min $\frac{1}{2}||\boldsymbol{w}||^2 s.t. \; y_i(\boldsymbol{w}^T\varphi(X_i) + b) \geq 1$

核函数$K(X_i, X_j)$的取值：映射函数$\varphi(X)$所得新数据$\varphi(X_i)$和$\varphi(X_j)$的内积

核函数的充分条件（Mercer定理）：任意半正定函数都可以作为核函数。非必要条件

核函数也可以通过线性组合、直积$k(\boldsymbol{u}, \boldsymbol{v}) = g(\boldsymbol{u})k_1(\boldsymbol{u}, \boldsymbol{v})g(\boldsymbol{v})$等得到

训练样本数足够多时：

此时经验风险大体能够代表泛化误差，使用经验风险来代替泛化误差作为模型泛化性能的度量指标；

训练样本数目较少时：经验风险和泛化误差之间通常会有较大差别，采用结构风险。

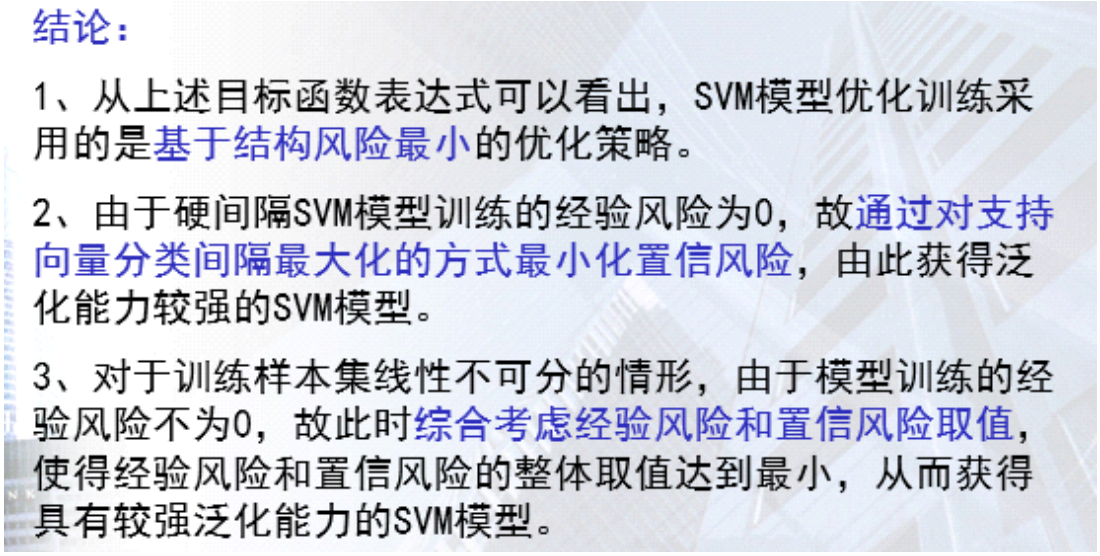
结构风险：经验风险+置信风险。$R_{\text{srm}}(f) = R_{\text{emp}}(f) + \alpha\lambda(f)$

置信风险与模型复杂程度成正比；置信风险与训练样本数成反比

VC维：对于给定的决策函数候选集$\mathcal{F}$，通过$\mathcal{F}$能打散最大测试样本集的基数h称为$\mathcal{F}$的**VC**维

$$\Phi(\lambda) = \Phi(n/h) = \frac{h(\ln(2n/h) + 1) - \ln(\eta/4)}{n}$$

SVM的$\frac{w^2}{2}$就是置信风险



第四章

2021年1月4日 1:07

划分聚类的基本思想：对样本数据进行划分，实现对样本数据的聚类分析。
划分聚类方法首先需要**确定划分块的个数即聚簇的个数**，然后通过适当方式将样本数据聚集成指定个数的聚簇。

k–均值聚类 基于同类样本在特征空间中应该相距不远的基本思想，将集中在特征空间某一区域内的样本划分为同一个簇，其中区域位置的界定主要通过样本特征值的均值确定。
（1）令s=0，并从D中随机生成k个作为初始聚类中心的数据点$u_1\^0,u_2\^0,\ldots,u_k\^0$;
（2）计算D中各样本与各簇中心之间的距离w，并根据w值将其分别划分到簇中心点与其最近的簇中；
(3）分别计算各簇中所有示例样本数据的均值，并分别将每个簇所得到的均值作为该簇新的聚类中心$u_1\^(s+1),u_2\^(s+1),\ldots,u_k\^(s+1)$；
（4）若$u_j\^(s+1)=u_j\^s$，则终止算法并输出最终簇，否则令$s=s+1$，并返回步骤（2)。

k–均值聚类的不足：算法要求每个样本数据点在一次迭代过程中**只能被划分到某个特定的簇**中。然而，在很多实际应用中样本数据并非都满足这种**非此即彼的刚性划分**，

模糊c–均值聚类
使用模糊数学中属于[0,1]区间的隶属度指标度量单个样本隶属于各个簇的程度，并规定每个样本到所有簇的隶属度之和均为1
加权欧式距离w_{ij}度量样本X_i与簇C_j之间的相关性：

$$w_{ij}=\alpha_{ij}\left(\sum_{t=1}^{m}\left(x_{it}-u_{jt}\right)^2\right)^{1/2}$$

p为控制隶属度影响的参数，通常取p=2

$$\arg_{\alpha_{ij}}\min J\left(\alpha_{ij}\right);\ \text{s.t.}\sum_{j=1}^{c}\alpha_{ij}=1 \qquad (4-5)$$

$$\hat{J}\left(\alpha_{ij}\right)=\sum_{j=1}^{c}\sum_{i=1}^{n}\alpha_{ij}^{p}\sum_{t=1}^{m}\left(x_{it}-u_{jt}\right)^2+\sum_{i=1}^{n}\lambda_i\left(\sum_{j=1}^{c}\alpha_{ij}-1\right) \qquad (4-6)$$

还可以将目标函数$\hat{J}\left(\alpha_{ij}\right)$看成是聚类中心$u_{jt}$的函数，即$\hat{J}\left(u_{jt}\right)$，并由此通过对目标函数$\hat{J}\left(u_{jt}\right)$作最小值优化计算进一步得到各簇最优聚类中心坐标$U_j$。

类似kmeans迭代，终止条件若$J^s\geq\varepsilon$或$\left|J^s-J^{s-1}\right|\geq\varepsilon$

密度聚类
密度聚类算法：将聚簇看作是数据空间中被稀疏区域分开的稠密区域，由此得到以密度为度量标准的样本数据聚类方法。

用于对具有任意形状的聚簇进行聚类

密度聚类概念：

两个参数ε和$MinPts$：

ε领域半径；$MinPts$在以ε为半径的领域内最少包含点的个数（密度阈值）

核心对象: 一个对象的ε-邻域至少包含$MinPts$个对象

边界对象：不是核心点 ,但落在某个核心点的ε邻域内的对象

噪声对象：不属于任何簇的对象。

密度直达(Directly density reachable, DDR):如果q是一个核心对象，p1属于q的邻域，那么称q密度直达p1。密度可达的(density reachable):点 p 关于ε和$MinPts$ 是从 q密度可达的, 如果存在一个节点链 p1, ..., pn, p1 = q, pn = p，pi直接密度可达pi+1，则称p密度可达q。

密度相连的:点 p关于 ε和$MinPts$与点 q是密度相连的, 如果 存在点 o 使得, p 和 q 都是关于ε和$MinPts$是从 o 密度可达的(如果存在o，o密度可达q和p，则称p和q是密度连通的)。

算法：1.任意选取一个点 p。2.得到所有从p 关于 ε和$MinPts$密度可达的点。3.如果p 是一个核心点, 则找到一个聚类。4.如果 p 是一个边界点, 没有从p 密度可达的点, DBSCAN 将访问数据中的下一个点。5.继续这一过程, 直到所有点都被处理。

OPTICS算法并不显示的产生结果类簇，而是为聚类分析生成一个增广的簇排序，这个排序代表了各样本点基于密度的聚类结构。从这个排序中可以得到基于参数ε和$MinPts$在任意取值下的聚类结果。

DBSCAN和OPTICS对epsilon敏感

DENCLUE 算法通过样本点的分布估计密度函数，用与每个点相关联的影响函数之和对数据集进行总密度建模，最终得到一个在属性空间中的用来描述数据集总密度的密度函数。

总密度函数会有局部尖峰，即局部密度极大值和局部低谷，即局部密度极小值，每一个尖峰对应了一个簇质心，而簇与簇之间通过低谷来分离。

尖峰也被称为局部吸引点或密度吸引点。

。。。。玄学看不懂了

PCA

任意选择的一组k个线性无关m维向量$\{\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_k\}$作为基向量构成变换矩阵$\boldsymbol{W}$，将数据集$D$中样本数据降至$k$维。因此，如何选择一组适当的基向量$\{\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_k\}$是实现对样本数据进行有效降维的关键技术。

此外，对数据集D中样本数据的降维应尽可能保留原数据有效信息。数据点分布的分散度可用方差度量，方差越大的属性，其包含的信息量就越大，故要求所选基向量$\{\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_k\}$使得映射后数据方差尽可能地变大。

所以先要标准化！

将标准化后数据组成新的数据矩阵$\boldsymbol{Z}=(Z_1,Z_2,\dots,Z_n)$并构造其协方差矩阵$\boldsymbol{C}$：

$$\boldsymbol{C}=\frac{1}{m}\boldsymbol{Z}^T\boldsymbol{Z}=\begin{pmatrix}\frac{1}{m}\sum_{j=1}^{m}{z_{1j}}^2 & \frac{1}{m}\sum_{j=1}^{m}z_{1j}\,z_{2j} & \cdots & \frac{1}{m}\sum_{j=1}^{m}z_{1j}\,z_{nj}\\ \frac{1}{m}\sum_{j=1}^{m}z_{1j}\,z_{2j} & \frac{1}{m}\sum_{j=1}^{m}{z_{2j}}^2 & \cdots & \frac{1}{m}\sum_{j=1}^{m}z_{2i}\,z_{nj}\\ \cdots & \cdots & \cdots & \cdots\\ \frac{1}{m}\sum_{j=1}^{m}z_{1j}\,z_{nj} & \frac{1}{m}\sum_{j=1}^{m}z_{2j}\,z_{nj} & \cdots & \frac{1}{m}\sum_{j=1}^{m}{z_{nj}}^2\end{pmatrix}$$

$$J(\boldsymbol{w_1})=\frac{1}{n}\sum_{i=1}^{n}\left(Z_i^T\boldsymbol{w_1}\right)^2$$

$\max\limits_{w_1}\ \boldsymbol{w}_1^TZZ^T\boldsymbol{w}_1$；s.t. $\boldsymbol{w}_1^T\boldsymbol{w}_1=1$　(4-21)拉格朗日函数$\max\limits_{\boldsymbol{w}_1}\ \boldsymbol{w}_1^T\boldsymbol{C}\boldsymbol{w}_1-\alpha\left(\boldsymbol{w}_1^T\boldsymbol{w}_1-1\right)$　(4-22)

令上述优化问题的目标函数对$\boldsymbol{w}$_1的偏导数为0，则有2$\boldsymbol{C}\boldsymbol{w}$_1−2$\alpha\boldsymbol{w}$_1=0，$\boldsymbol{C}\boldsymbol{w}_1=$

$\alpha\boldsymbol{w}_1$　(4-23)此，$\boldsymbol{w}_1$是协方差矩阵$\boldsymbol{C}$的一个特征向量，α是与该特性向量对应的特征根。又因：

$\boldsymbol{w}_1^T\boldsymbol{C}\boldsymbol{w}_1=\alpha\boldsymbol{w}_1^T\boldsymbol{w}_1=\alpha$故要使得$\boldsymbol{w}_1^TZZ^T\boldsymbol{w}_1$取得最大化，即使得$\boldsymbol{w}_1^T\boldsymbol{C}\boldsymbol{w}_1$取得最大化，$\boldsymbol{w}_1$即为协方差矩阵$\boldsymbol{C}$的最大特征根$\lambda_1$所对应的特征向量，可由此获得样本数据$\boldsymbol{X}$或$\boldsymbol{Z}$的第一个主成分$\boldsymbol{z}_1'=\boldsymbol{w}_1^T\boldsymbol{Z}$。

算法：求出协方差矩阵$\boldsymbol{C}$全部特征根并将这些特征根按照从大到小次序排列，选择前k个特征值所对应特征向量按行排列构成变换矩阵$\boldsymbol{W}$；

累计贡献率$\Omega=\sum_{i=1}^{k}\lambda_i/\sum_{i=1}^{n}\lambda_i$ >0.99 or 0.97

核PCA

核主分量分析首先通过核映射技术对原始高维数据做进一步的升维变换

核主分量分析使用某个核函数$K\left(X_i,X_j\right)=\varphi^T(X_i)\varphi(X_j)$，所对应的核矩阵$\boldsymbol{K}$代替协方差矩阵$\boldsymbol{C}$。

稀疏编码

稀疏编码：将原始非稀疏数据转化为高维的稀疏数据进行处理，从数学上看，稀疏编码的目的是寻找一组适当的基向量将非稠密的原始样本数据映射成具有一定稀疏性数据。

对于任意m维样本数据$X_i=(x_{i1},x_{i2},\dots,x_{im})^T$，可将其表示为如下线性组合：

$$X_i=\sum_{j=1}^{k}\theta_{ij}\boldsymbol{w}_j\qquad(4-29)$$

其中θ_{ij}为元素x_{ij}所对应的组合系数，$\boldsymbol{w}_j$为元素x_{ij}所对应的基向量。

稀疏编码的目的是寻找到一组适当的基向量$\boldsymbol{w}_1,\boldsymbol{w}_2,\dots,\boldsymbol{w}_k$，使得样本数据在这组基向量的表示下大部分系数为0，使得这种数据表示具有一定的稀疏性。

对数据矩阵$\boldsymbol{X}$进行稀疏编码的结果是将其分解成字典矩阵W与组合系数矩阵θ的乘积，字典矩阵$\boldsymbol{W}$是由所求基向量$\boldsymbol{w}$_1,$\boldsymbol{w}$_2,...,$\boldsymbol{w}$_k组成的矩阵。

对样本数据集$D=\{X_1,X_2,\dots,X_n\}$进行稀疏编码，则需要实现如下两个目标：

1.寻找一组适当的基向量$\boldsymbol{w}_1,\boldsymbol{w}_2,\dots,\boldsymbol{w}_k$将D中所有样本数据表示成这组基向量的线性组合形式；

2.尽量使得大部分线性组合系数为0；

可根据上述两个目标构造相应的目标函数，并将对样本数据集$D=\{X_1,X_2,\dots,X_n\}$的稀疏编码转化为对如下优化问题的求解：

$$\arg_{\theta_{ij},\boldsymbol{w}_j}\min\left(\sum_{i=1}^{n}||X_i-\sum_{j=1}^{k}\theta_{ij}\boldsymbol{w}_j||_2^2+\lambda\sum_{j=1}^{k}J(\theta_{ij})\right)\quad(4-30)$$

在对数据进行稀疏编码的求解步骤如下：

1.设定初始字典矩阵$\boldsymbol{W}_0$，设定t=0及算法终止条件；

2.将字典矩阵$\boldsymbol{W}_t$作为已知量带入目标函数，并对目标函数进行优化求得相应参数矩阵$\boldsymbol{\theta}_t$；

3.若不满足算法终止条件，则由参数矩阵$\boldsymbol{\theta}_t$算出新一轮迭代的字典矩阵$\boldsymbol{W}_(t+1)$并令$t=t+1$，返回步骤（2）；否则结束迭代，返回字典矩阵$\boldsymbol{W}_t$和参数矩阵$\boldsymbol{\theta}_t$；

4.计算并输出样本数据集D所对应数据矩阵$\boldsymbol{X}$的稀疏表示$\boldsymbol{\theta}_t\,\boldsymbol{W}_t$

稀疏表示学习

在字典矩阵确定的情况下，如何求解满足一定稀疏条件的系数矩阵为啥求，因为表示不唯一呀！

L0范数 不为0的个数，严格来说不是范数　求解需要交换约数条件和目标函数

$\arg_{\boldsymbol{\theta}}\min||\boldsymbol{X}-\boldsymbol{\theta W}||_F^2+\lambda||\boldsymbol{\theta}||_0$

L1、Lp范数其实也可以用类似的方式求

$\arg_{\boldsymbol{\theta}}\min||\boldsymbol{\theta}||_p,\quad \text{s.t. } \boldsymbol{X}=\boldsymbol{\theta W}$

$\arg_{\boldsymbol{\theta}}\min||\boldsymbol{X}-\boldsymbol{\theta W}||_F^2+\lambda||\boldsymbol{\theta}||_p$

加权范数稀疏表示模型$||\boldsymbol{\lambda A}||_1=\sum_i\sum_j w_{ij}|a_{ij}|\quad(4-39)$

其中权重计算公式为：$w_{ij}=\begin{cases}\frac{1}{|a_{ij}|}, & a_{ij}\neq 0\\ \infty, & a_{ij}=0\end{cases}$

基于L1模型$\arg_{\boldsymbol{\theta}}\min||\boldsymbol{X}-\boldsymbol{\theta W}||_F^2+||\boldsymbol{\lambda\theta}||_1$

匹配追踪算法的基本思想：通过减小残差的方式逐步逼近原数据。对于确定的字典矩阵，对样本数据进行线性映射的像空间结构是已知的，此时为求得对应的组合系数矩阵，通过不断减小原始样本向量与其像之间的残差实现。

稀疏表示学习解决了在已知字典矩阵W条件下计算系数矩阵θ的问题。如何通过已知的系数矩阵θ解决字典矩阵$\boldsymbol{W}$的计算问题，即字典矩阵$\boldsymbol{W}$的自动构造方法。

MOD方法。在字典学习过程中，由于数据矩阵X和系数矩阵θ均已知，只需确定字典矩阵W使得X与θW之间的差别达到最小即可，故MOD方法直接使用最小二乘法求得W。

字典学习优化计算问题的目标函数为：$J(\boldsymbol{W})=||\boldsymbol{X}-\boldsymbol{\theta W}||_F^2$

K-SVD方法。K-SVD方法的基本思想是通过依次更新字典矩阵$\boldsymbol{W}$中的原子实现对整个字典矩阵的更新，并且在更新原子$\boldsymbol{w}_j$时，其它原子均保持不变。其字典学习的目标函数表示为如下形式：

1）输入数据矩阵$\boldsymbol{X}$，从中随机选择k个样本，并根据这些样本生成初始字典矩阵$\boldsymbol{W}^0$，令$t=$

0，$s=1$，设定阈值ε；

（2）根据数据矩阵$\boldsymbol{X}$和字典矩阵$\boldsymbol{W}^0$进行稀疏表示学习，得到系数矩阵$\boldsymbol{\theta}^0$；

（3）选择字典矩阵$\boldsymbol{W}^t$中的原子$\boldsymbol{w}_s$作为待更新原子并固定其它原子的参数，结合数据矩阵$\boldsymbol{X}$和系数矩阵$\boldsymbol{\theta}^t$计算残差矩阵：

$$\boldsymbol{E}_{\boldsymbol{w}_j}=\left(\boldsymbol{X}-\sum_{i\neq s}\boldsymbol{w}_i\boldsymbol{\vartheta}_i\right)$$

（4）选择$\boldsymbol{E}_{\boldsymbol{w}_j}$中只与$\boldsymbol{\vartheta}_s$的非0元素相关的列组成矩阵$\boldsymbol{E}'_{\boldsymbol{w}_j}$；

（5）对矩阵$\boldsymbol{E}'_{\boldsymbol{w}_j}$进行奇异值分解，求得$\widehat{\boldsymbol{w}}_s$并将$\widehat{\boldsymbol{w}}_s$代替$\boldsymbol{w}_s$写入字典矩阵$\boldsymbol{W}^t$中，若字典矩阵$\boldsymbol{W}^t$中存在未更新的原子，则令$s=s+1$并返回步骤（3），否则执行步骤（6）；

（6）若$\left\|\boldsymbol{X}-\boldsymbol{\theta}^t\boldsymbol{W}^t\right\|_F^2\geq\varepsilon$则令$t=t+1$并返回步骤（2），否则输出$\boldsymbol{W}^t$并结束算法。

图像降噪

初始化字典学习(划分小像素块，KSVD)稀疏表示学习图像重建

第五章

2021年1月4日　　2:07

集成学习将多个性能一般的普通模型进行有效集成，形成一个性能优良的集成模型，通常将这种性能一般的普通模型称为个体学习器。如果所有个体学习器都属于同类模型，则称由这些个体学习器产生的集成模型为同质集成模型，并称这些属于同类模型的个体学习器为基学习器。反之，将属于不同类型的个体学习器进行组合产生的集成模型称为异质集成模型。

上述情况是由于弱学习器泛化性能均太弱造成的。在集成学习的实际应用当中，应尽可能选择泛化性能较强的弱学习器进行组合，如图5-2所示，当每个弱分类器分类错误的样本各不相同时，则能得到一个效果优异的集成模型。

图 5-1 过弱泛化性能个体学习器集成效果

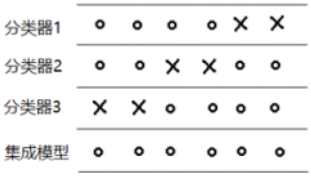


图 5-2 个体学习器的差异对集成结果的影响

使用单个学习器会带来过多的模型偏好，从而产生模型泛化能力不强的现象，结合多个弱学习器则可以有效降低此类风险。

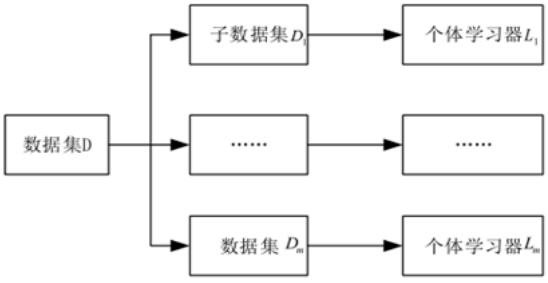


图 5-3 弱学习器并行构造方式

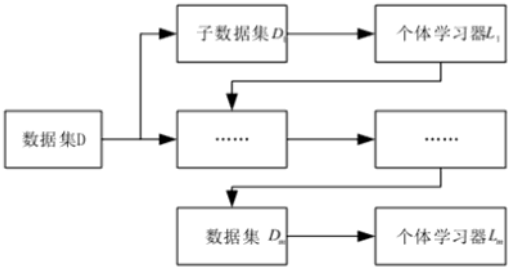


图 5-4 弱学习器串行构造方式

组合弱学习器方法

回归问题：1.简单平均 会过分依赖不重要的弱学习器，泛化差

2. 加权平均

分类问题：相对、绝对多数投票法（相对投票最多票少，票数比较分散），

加权计算投票

集成学习泛化

$aveD = \frac{1}{m}\sum_{i=1}^{m}\left(L_i(X) - L(X)\right)^2$ 一组弱回归器的差异度或多样性可表示为该组所有弱回归器关于集成模型输出偏差的平均值

$Q\left(L_i, X\right) = \frac{1}{m}\sum_{i=1}^{m}\left(f(X) - L_i(X)\right)^2$ 令Q为所有若回归其对于输入

则有：$Q(\mathrm{L}, X) = \overline{Q\left(L_i, X\right)} - aveD$

降低弱学习器的泛化误差: 样本扩充、范数惩罚等机器学习正则化策略。

提高个体学习器的多样性：改变训练样本和改变模型训练参数。

对于给定的样本数据集D,Bagging集成学习主要通过自助采样法生成训练样本数据子集。假设D中包含有n个样本数据，自助采样对D进行n次有放回的随机采样并将采样样本纳入训练集。可将这些未被抽到的样本构成测试集，用于测试弱学习器的泛化性能。
对样本数据集D进行多次自助采样就可以分别生成多个具有一定差异的训练样本子集$D_1, D_2, \cdots, D_K$，可分别通过对这些子集的训练构造出所需的弱学习器。一般通过简单平均法集成多个弱回归器，通过相对多数投票法集成多个弱分类器。Bagging集成学习的基本流程图如图5-6所示。

随机森林模型在Bagging集成策略基础上进一步增加了弱学习器之间的差异性，这使得随机森林模型能有效解决许多实际问题。

假设在确定决策树 T_i中某个结点的划分属性时，该结点所对应样本特征属性集合为:

$A_i = \{a_{i1}, a_{i2}, \cdots, a_{im}\}$

（1）通过自助法确定的训练样本集D_i;

（2）从当前样本数据集的特征集合$A_i = \{a_{i1}, a_{i2}, \cdots, a_{im}\}$中随机选择$s$个特征组成新的特征集合$A_i' = \{a_{i1}', a_{i2}', \cdots, a_{is}'\}$;

（3）分别计算A_i'中所有属性关于该样本数据集的基尼指数，并根据基尼指数确定最优特征切分点，并依据最优特征切分点将D_i 的中样本分配到子结点所对应样本子集中。

(4）分别对两个子结点递归地调用（2）-（3），直至满足算法停止条件。不难发现，上述训练过程没有包含剪枝操作。这是因为决策树剪枝操作可能会造成单棵决策树模型的预测偏差增大，不利于提升随机森林模型的泛化性能。

Boosting 集成学习方法，又名提升式集成学习方法。该方法主要通过集成各个弱学习器的成功经验和失败教训实现对模型性的提升。

该方法使用迭代方式完成对各个弱学习器的训练构造，每次迭代对训练样本集的选择都与前面各轮的学习结果有关，使用前面各轮学习结果更新当前各训练样本的权重，对前面被错误预测的赋予较大的权重，实现对当前训练样本集合数据分布的优化。

Boosting 集成学习通常使用两种方式调整训练样本集的数据分布：

一是仅调整样本数据的权重，而不改变当前训练样本集合；提高当前训练样本集合中被错误预测样本的权重，

二是改变当前训练样本集合，复制被前面弱学习器错误预测样本到样本训练集当中重新进行训练。

Schapire’s Boosting算法 我觉得这个玩意不重要

AdaBoost（ Adaptive Boosting ）自适应改权重

（1）令$i = 1$并设定弱学习器的数目m。使用均匀分布初始化训练样本集的权重分布，令n维向量w^i表示第i次需更新的样本权重，则有：$w^1 = (w_{i1}, w_{i2}, \cdots, w_{in})^T = \left(\frac{1}{n}, \frac{1}{n}, \cdots, \frac{1}{n}\right)^T$

（2）使用权重分布为w^i的训练样本集D_i学习得到第i个弱学习器L_i；

3）计算L_i在训练样本集D_i上的分类错误率e_i ：$e_i = \sum_{k=1}^{n} w_{ik} I(L_i(X_k) \neq y_k)$

（4）确定弱学习器L_i的组合权重α_i 。由于弱学习器L_i的权重取值应与其分类性能相关，对于分类

错误率e_i越小的L_i ，则其权重α_i 应该越大，故有：$\alpha_i = \frac{1}{2}\ln\frac{1-e_i}{e_i}$

（5）依据弱学习器L_i对训练样本集D_i的分类错误率e_i 更新样本权重，更新公式为：

$$w_{i+1,j} = \frac{w_{ij}\exp(-\alpha_i y_k L_i(X_k))}{Z_i}$$

其中：$Z_i = \sum_{k=1}^{n} w_{ij}\exp(-\alpha_i y_k L_i(X_k))$为归一化因子，保证更新后权重向量为概率分布；

AdaBoost 集成学习算法关键要点是如何更新样本权重，即步骤（5）中的权重更新公式。则可将该公式改写为如下形式：

$$w_{i+1,j} = \begin{cases} \frac{w_{ij}}{Z_i}\exp(-\alpha_i), L_i(X_k) = y_k \\ \frac{w_{ij}}{Z_i}\exp(\alpha_i), L_i(X_k) \neq y_k \end{cases}$$

即当某个样本被前一个弱学习器错误预测时，该样本的权重会被放大$e_i/(1-e_i)$倍以便在后续弱学习器构造过程得到应有的重视。

GBDT的含义为梯度提升决策树（Gradient Boosting Decision Tree），是一种以回归决策树为弱学习器的集成学习模型，主要用于完成机器学习的回归任务。GBDT集成学习模型通常使用CART决策树（回归树）模型作为弱学习器。

$$e(L_0(X)) = \frac{1}{2}[L_0(X) - y]^2 \qquad \frac{\partial e(L_0(X))}{\partial L_0(X)} = L_0(X) - y$$

则梯度的反方向为$y - L_0(X)$，应对模型$L_0(X)$往该方向进行调整。然而，由于单个回归决策树模型的结点数目较少难以有效拟合所有训练样本的梯度方向，故通常无法直接根据上述方向对模型$L_0(X)$进行更新。

可构造一个新的模型$L_1(X)$对模型$L_0(X)$的预测误差$y - L_0(X)$进行拟合。
由于L_1对于X的输出是对L_0输出的某个校正量，且校正方向一定是误差e减小的方向，故这两个模型的输出之和$L_0\ (X)+L_1\ (X)$一定比$L_0\ (X)$更加接近样本真实值y。

1）构造初始学习器$L^0(X)$。令$t = 0$,根据下式构建初始回归树$L^0(X) = L_0(X)$：$L_0(X) = \arg_c \min \sum_{(X_i,y_i)\in D} J(y_i, c)$　　(5-18)

其中$L_0(X)$为只有一个根节点的初始回归决策树，c为使得目标函数最小化的模型参数，$J(y_i, c)$为损失函数。

这里采用平方误差损失函数，即有：

$$J(y, g(X)) = \frac{1}{2}(y - g(X))^2 \qquad (5-19)$$

其中y为样本真实值或标注值，$g(X)$为单个回归决策树模型的预测。

（2）令$t = t + 1$,并计算数据集D中每个训练样本的负梯度∇_i　：　　$\nabla_i =$

$$-\left[\frac{\partial J(y,L(X_i))}{\partial L(X_i)}\right]_{L(X)=L^t(X)} \quad (5-20)$$

（3）构建新的训练样本集T_t：

$$T_t=\{(X_1,\nabla_1),(X_2,\nabla_2),\dots,(X_n,\nabla_n)\} \quad (5-21)$$

使用T_t作为训练样本集构造一棵回归树，并使用该回归树作为第$t+1$个弱学习器$L_t(X)$，该决策树中第j个叶子的输出值为：

$$\mathrm{C}_{t,j}=\arg_c\min\sum_{(X_i,\nabla_i)\in T_t^j}J(y_i,L^t(X_i)+c) \quad (5-22)$$

其中T_t^j表示第$t+1$个弱学习器的第j个叶子节点所对应的数据集合。

上式表明弱学习器$L_t(X)$中每个叶节点的输出均使得上轮迭代所得集成模型$L^{t-1}(X)$的预测误差达到最小

可将回归决策树$L_t(X)$表示为：$L_t(X)=\sum_j \mathrm{C}_{t,j}I[(X_i,\nabla_i)\in T_t^j]$ (5-23)

其中：$I\left[(X_i,\nabla_i)\in T_t^j\right]=\begin{cases}1(X_i,\nabla_i)\in T_t^j\\0(X_i,\nabla_i)\notin T_t^j\end{cases}$

（4）更新集成模型为：$L^t(X)=L^{t-1}(X)+L_t(X)$ (5-25)

（5）若未满足算法终止条件，则返回步骤（2），否则算法结束。

第六章

2021年1月4日 2:41

序贯决策过程是在游戏博弈或对弈等应用场合完成任务时需要连续进行多步决策的过程；
序贯决策问题是如何让计算机像人类一样能够自动进行合理的序贯决策。
强化学习的目标是通过机器学习方式有效解决序贯决策问题，或者说通过机器学习方式实现对连续多步自动决策问题的优化求解。

智能体是行为的执行者，在实际应用中可能是一个游戏玩家、一个棋手或一辆自动驾驶的汽车等；
动作是智能体发出的行为，例如在自动驾驶任务中汽车向右转弯便是一个动作；
系统环境是智能体所处的外部环境，也是智能体的交互对象，例如在自动驾驶任务中系统环境便是实际的交通环境
状态是智能体当前所处的可观察状态，如自动驾驶任务中的汽车速度、汽车与路边的距离等。
奖励或反馈是系统环境能够对智能体的行为做出的某种合理评价。例如可将汽车自动驾驶的安全行驶里程数作为反馈信息。
强化学习的目标是使得智能体的动作满足某一任务需求，例如希望自动驾驶汽车能够通过一系列自动操作安全驾驶到目的地。

有模型强化学习：强化学习通过建立环境模型来对智能体和系统环境进行模拟，并且系统环境满足已知且有限
系统环境**有限**指的是动作集合，奖励集合，状态集合为有限集。
系统环境**已知**指的是在智能体选择某一动作时环境给予的奖励值为已知，并且在动作执行后环境的状态改变为已知。
不能或难以建立环境模型的强化学习称为**无模型强化学习**

值函数描述了从当前动作开始到将来的某一个动作执行完毕为止所获累计奖励值，故值函数是对多次连续动作满意度的度量。

> 由于强化学习的目的是使得智能体一系列的动作满足任务需求，**故通常将值函数作为强化学习优化计算的目标函数**

收敛速度慢多数强化学习算法收敛到最优解的理论保障都是建立在任意状态都能被无限次访问到这个前提条件之上。
强化学习会经常面临利用已经学到知识还是对未知知识进行探索的**平衡难题**。
产生这个问题的根源在于难以权衡长期利益和短期利益。
一方面为了获得较高的奖赏，智能体需要利用学到的经验在已经探索过的动作中贪心地选择一个获益最大的动作；
另一方面，为了发现更好的策略，智能体需要扩大探索范围，尝试以前没有或较少试过的动作。若不能权衡好两者的关系，智能体就处于进退两难境地。
由于强化学习具有**回报延迟**的特点，即环境反馈给智能体的信息比较稀疏且有一定延时，故当智能体收到一个奖赏信号时，决定先前的哪些行为应分配到多大权重有时比较困难。

通常使用一个小于1的折扣因子γ表示权重随时序向后推移而逐步衰减的效果。

$w^{(t+j)} = \gamma^{j-1}$

有模型的可以动态规划

无模型：分层、启发式

ε–贪心策略中的ε–判别函数作为无模型强化学习的启发函数

启发函数的选择对强化学习的效果具有很大影响。目前主要通过两种方式确定启发函数。

第一种方式是直接基于领域先验知识构造启发函数。

第二种方式是通过在学习过程中获得的信息构造启发函数。

启发函数的构造过程可大致分为两个基本阶段：

第一阶段是结构提取阶段，完成的任务是根据值函数实现领域结构的提取；

第二阶段是启发式构造阶段，完成的任务是根据提取到的领域结构构造启发式函数。下图表示启发函数构造的基本流程。

基本强化学习

值迭代 每一次贪婪选择值最大的动作

值迭代通常从已知的当前状态s开始对当前策略进行评估，通过迭代方式计算新策略状态值函数$V_h(s)$取值。

这种计算方式在动作空间和状态空间均为离散空间且规模较小时较为有效，但对于连续或规模较大的动作空间或状态空间，值迭代优化方法的计算成本通常较高且容易陷入局部最优。

可用基于函数逼近思想的冗余值迭代算法解决这个问题

时序差分学习的基本思想是首先通过模拟一段时序中的状态变化方式估计动作值函数的取值，然后，在每执行一次或几次状态转移之后根据所得新状态的价值对估计值进行迭代更新。

Q学习则是一种**异策略算法**，在动作选择时所遵循的策略与更新动作值函数时的策略不同。
为使得动作值函数的更新过程更快收敛，Q学习算法直接选择状态$s^{'}$所对应的最大值函数参与更新过程

示范强化学习
模仿 、逆向
使用上述方法确定权重向量的学习方式通常称之为**学徒学习。**
由以上分析可知，学徒学习从某个初始策略开始求解回报函数的参数值，并利用所求回报函数和现有强化学习方法更新策略，不断重复上述过程直至算法收敛，实现对最优策略的求解。

啊这都什么玩意

第七章

2021年1月4日 3:02

Sigmoid函数是一种最常用的激活函数，可将人工神经元的输出限制在区间(0,1)内

tanh函数将Sigmoid函数图像在竖直方向上拉伸了两倍并向下平移了一个单位，有时会更便于进行模型参数求解
感知机模型是一种只有一层神经元参与数据处理的人工神经网络模型

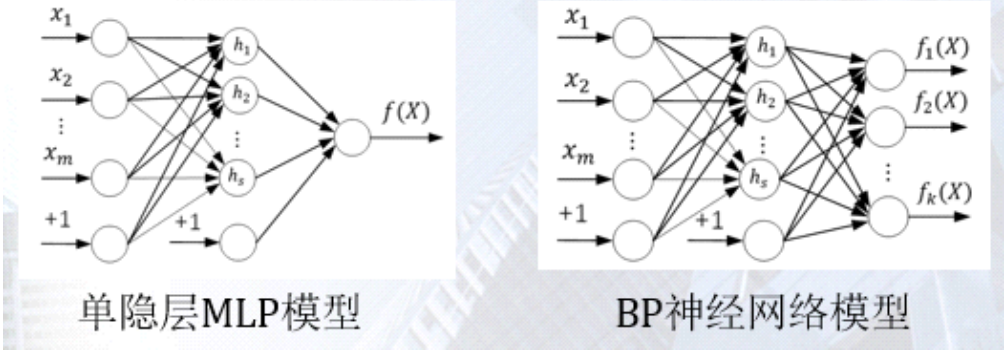
通常将此类没有环路或回路的人工神经网络称为前馈网络模型。感知机是一种最简单的前馈网络模型

MLP模型的网络结构没有环路或回路，故是一类前馈网络模型。MLP模型中隐含层的层数可为一层也可为多层。

MLP模型中信息处理神经元的激活函数通常为Sigmoid函数。故MLP模型的隐含层可将数据通过非线性映射表示在另一个空间当中，并将模型输出限制在区间(0,1)当中。

MLP模型可通过隐含层将原始数据分布映射，可见原线性不可分的数据映射后线性可分。

由于MLP模型输出层只有一个神经元，故只适用于二分类任务。对于多分类任务，可考虑增加输出层神经元个数，使得模型具备处理多分类任务的能力，由此得到一种如下图所示的反向传播神经网络模型或BP(Back Propagation)神经网络模型。



神经网络中最常见的编码方式有二进制编码方式和独热式编码方式两种。

对于采用独热式编码的BP神经网络，其输出层神经元的激活函数通常采用softmax激活函数，该激活函数可将神经元输出转化为样本属于某一类别的伪概率。

$$\sigma_i(\boldsymbol{t}) = e^{t_i} / \sum_j e^{t_j}$$

其中直接影响网络模型性能因素包括训练样本集的大小及样本质量、网络模型结构、优化目标函数形式和模型优化算法。在模型的训练构造过程中通常综合考虑这些因素；构建一个神经网络模型大

致可分为数据准备与预处理、模型初始化、确定优化目标、模型优化求解和验证模型性能这五个基本步骤。

一、数据准备与预处理

1. 首先需要针对任务需求收集样本并对其进行标注
2. 样本增强方式实现对训练样本集扩充
3. 将带标注样本划分为两部分，其中一部分样本作为训练集用于模型训练，其余部分作为测试集用于验证模型性能
4. 对数据进行特征提取等预处理
5. 采用合适方式对标签数据进行编码

二、模型初始化

模型初始化参数一般有：连接权重、偏置项、超参数。

这种建立初始网络模型结构和模型参数进行初始赋值的过程通常称之为模型初始化过程。

模型初始化过程确定了模型优化过程从何处开始，从一组较好的模型参数开始的训练过程通常能够避免参数陷入局部最优并获得性能较好的优化模型。

hinge损失函数的具体形式如下$L(X_i, y_i) = \max(0, 1 - f(X_i) \cdot y_i)$

在神经网络模型优化过程中，可采用一种特殊的正则化手段以缓解模型过拟合现象。这种正则化方法通常称之为随机失活(**Dropout**)方法。

Dropout正则化方法的基本思想通过随机去除网络模型中的非输出结点的方式实现减小模型容量效果。具体做法是在模型训练的每次迭代过程中，对于除输出层之外的任意一层神经元，以一定概率p设置每一个神经元的输出为0，即使其失活。

采用Dropout正则化方法相当于构造了众多不同子网络模型，并将这些子网络集成起来获得训练模型。这个过程与Bagging集成方法比较类似，故能有效提高训练模型的泛化性能。

由于单次模型性能验证实验存在很强的随机性，故通常采用交叉验证法进行模型性能测试和验证

径向基函数(Radial Basis Function，RBF)神经网络则是一种局部逼近型网络模型。所谓局部逼近型网络，是指网络模型输出仅与少数几个连接权重相关，对于每个参与模型训练的样本，通常仅有少数与其相关的权重需要更新。这种局部性的参数更新方式有利于加快模型训练过程。

RBF网络对于此类问题的求解思路则是通过对已知离散数据进行插值的方式确定网络模型参数

监督学习中心方法和自组织学习方法。监督学习中心方法直接将径向基函数中心、扩展常数和连接权重作为可学习参数使用监督学习方法对其进行更新，自组织学习方法的基本思想是通过无监督学习的$k-$均值聚类算法自动确定径向基函数的中心，并根据聚类中心之间的距离确定扩展常数。

自编码器

在机器学习任务中经常需要采用某种方式对数据进行有效编码，例如对原始数据进行特征提取便是几乎所有机器学习问题均需解决的编码任务。除此之外，对数据进行降维处理或稀疏编码也是常见的编码任务。

通常对数据进行编码时需要按照编码要求将原始数据转化为特定形式的编码数据，并要求编码数据尽可能多地保留原始数据信息，对这样的编码数据进行分析处理不仅会更加方便，而且可保证分析处理的结果较为准确。

由于自编码器隐含层输出向量$\boldsymbol{y}=\left(f_1(X),f_2(X),\ldots,f_s(X)\right)^T$即为编码数据，故只需限制$\boldsymbol{y}$中取值为0的分量较多便可实现对原始数据的稀疏编码。

如果某种自编码器能对被破坏的数据X^b进行编码并将其解码为真实原始数据X，则该自编码器的编码方式显然更为有效。通常称这种自编码器为降噪自编码器。，降噪自编码器比普通的自编码器具有更强的鲁棒性。

构建降噪自编码的思路较为简单，只需将原始自编码器模型训练过程中输入数据X替换为被破坏的数据X^b即可

玻尔兹曼机

事实上，还可从系统稳定性角度出发设计目标函数。由于系统越稳定则其能量越低，故为得到一个稳定的模型输出，可设计与网络模型相关的能量函数作为网络模型优化的目标函数，由此实现对神经网络模型的优化求解。玻尔兹曼机便是此类神经网络的代表模型。

该模型包含可视层与隐含层两层神经元，通过可视层神经元完成与外部的信息交互且可视层与隐含层的所有神经元均参与信息处理过程。

玻尔兹曼机中所有神经元的两两之间均存在信息传递且任意两个神经元之间的连接权重均相等

玻尔兹曼机中每个神经元的输出信号均限制为0或1，并且每个神经元的状态取值均具有一定的随机性，即以一定概率输出0或输出1，这个概率与该神经元的输入相关

玻尔兹曼机在运行过程中更倾向于选择使得模型能量函数更低的神经元输出，即使得能量函数取值更小的神经元输出发生的概率较大。

事实上，这一概率还与温度T相关。

当温度T的取值很大时，神经元处于各状态的概率相近，此时网络以近乎随机的方式运行，网络处于各状态的概率几乎相等。

由于能量函数全局最小值所对应收敛域通常大于极小值的收敛域。因此，在温度T的取值很大时，网络状态值有较大概率运行到能量函数全局最小值所对应收敛域中。

当温度T较小时，网络处于各状态的概率差异较大，在网络状态值很难跳出当前收敛域，但并非没有可能；

当温度T趋于0时，网络状态值几乎再无跳出当前收敛域的可能，此时能量函数收敛于该收敛域所对应极小值点
由以上分析可知，若直接从某一较低温度开始运行模型，则模型达到稳态时所对应的能量函数取值很有可能是局部最小值。为避免这种情况发生，可从某个较高的温度开始运行模型并逐步降低温度，由此以较大概率获得全局最优模型。

模拟退火算法正是基于这种思想一种启发式优化搜索算法。

在当前温度T下，若有$E\left(w_{ij},O_i,0\right)>E\left(w_{ij},O_i,1\right)$，则取$O_j=1$，否则计算

由于在使用模拟退火算法优化能量函数之前，已通过最大似然估计方法确定了模型的最优权重值，这组权重能够使得玻尔兹曼机有效拟合训练样本集D的分布，故在模型取该组权重的基础上再通过模拟退火算法优化能量函数，就可使得所求模型既能较好地拟合训练样本集数据分布，又能达到能量最低的稳态。

深度堆栈网络
模型太深：1.梯度消失2.模型复杂导致出现鞍点容易陷入局部最小

从输入数据开始逐步训练浅层学习模型，再将前一个浅层学习模型中某一层的输出作为下一个浅层学习模型的输入并对该模型进行训练，重复上述过程直至构建了多个浅层学习模型，

通过自编码器进行堆叠所获得的深度网络模型称之为深度堆栈网络。与上述方法类似，也可通过对受限玻尔兹曼机进行堆叠获得参数较优的深度网络模型，通常称此类网络模型为深度置信网络。下图（左，右）分别表示深度堆栈网络和深度置信网络的基本结构。

堆叠过程完成之后所得的深度网络模型仅能获得关于原始数据的高层特征表示，还需在其后添加一层输出层，由此获得深度堆栈网络的完整结构。

例如对于k个类别的多分类任务，可在堆叠所得网络的最后添加包含k个使用softmax激活函数的神经元作为输出结点，获得完整的深度堆栈网络。

深度堆栈网络与普通的深层前馈网络模型具有相同的拓扑结构，但普通的深层前馈网络的所有参数通常均由随机初始化方式得到，而深度堆栈网络从输入层开始到最后一个隐含层之间的连接权重均通过逐层训练方式获得，只有隐含层到输出层之间的连接权重需要通过随机初始化等方式得到。