2022-2023数据挖掘复习题整理

一、概论

1.1 名词解释

1. 学习相关
    1. **机器学习**：机器学习是用数据或以往的经验，以此优化计算机程序的性能标准；对于某类任务T和性能度量P，如果一个计算机程序在T上其性能P随着经验E而自我完善，那么我们称这个计算机程序从经验E中学习
    2. **强化学习**：强调如何基于环境而行动，以取得最大化的预期利益；其关注点在于寻找探索（对未知领域的）和利用（对已有知识的）的平衡
    3. **主动学习**：指通过自动的机器学习算法，从数据集中自动挑选出部分数据请求标签；有效的主动学习数据选择策略可以有效地降低训练的代价，并同时提高模型的识别能力
    4. **监督学习**：最常见的一种机器学习，可以由训练资料中学到或建立一个模式，它的训练数据是有标签的，训练目标是能够给新数据（测试数据）以正确的标签
    5. 无监督学习：用于在大量无标签数据中挖掘信息，它的训练数据是无标签的，训练目标是能对观察值进行分类或者区分等
    6. 半监督学习：用于在大量无标签数据中发现些什么。它的训练数据是无标签的，训练目标是能对观察值进行分类或者区分等
2. 风险相关
    1. **损失函数**：表示针对单个具体样本的模型预测值与真实样本值之间的差距
    2. **期望风险**：期望风险是全局概念，它是对所有样本，即对已知的训练样本加未知样本的预测能力
    3. **经验风险**：表示决策函数对训练数据集里的样本的预测能力，是模型关于训练样本集的平均损失
    4. **结构风险**：在经验风险的基础上加上表示模型复杂度的正则化项或罚项，防止发生过拟合
3. 过程相关
    2. **特征选择**：选取原始特征集合的一个有效子集，保留有用特征，移除冗余或无关的特征
    3. **特征提取**：通过属性间的关系，如组合不同的属性得到新的属性，这样就改变了原来的特征空间
    4. **交叉验证**：重复的使用数据，把得到的样本数据进行切分，组合为不同的训练集和测试集，得到多组不同的训练集和测试集，用训练模型、评估模型预测的好坏
    5. 验证集：是模型训练过程中单独留出的样本集，它可以用于调整模型的超参数和用于对模型的能力进行初步评估
    6. 训练误差：机器学习模型在训练数据集上表现出的误差
    7. 泛化误差：在任意一个测试数据样本上表现出的误差的期望值
4. 独立同分布

    在概率统计理论中，指随机过程中，任何时刻的取值都为随机变量，如果这些随机变量服从同一分布，并且互相独立，那么这些随机变量是独立同分布

5. 二分类

   混淆矩阵、查准率、查全率、ROC曲线横纵坐标、AUC

6. 机器学习的步骤

   收集数据、数据预处理、选择模型、训练、超参数调整、预测

1.2 简答分析

1. 有监督学习/无监督学习的学习机制、代表算法
2. 什么是过拟合，原因，以及避免的方法
3. 验证集有什么作用，怎样使模型学习到全部数据

一、贝叶斯决策论

2.1 概念简答

1. 基本概率概念
    1. 条件概率
    2. 全概率
    3. 先验概率：事情还没有发生，根据以往经验和分析得到的概率
    4. 类条件概率：已知一个条件下，结果发生的概率，条件概率实际上把一个完整的问题集合S通过特征进行了划分，类条件概率为其中之一
    5. **后验概率（贝叶斯公式）**：已知一个条件下，结果发生的概率。条件概率实际上把一个完整的问题集合S通过特征进行了划分
2. **（最小错误）贝叶斯决策理论**

   指在不完全信息（已知一些条件）下，对部分未知的状态用主观概率估计，然后用贝叶斯公式对发生概率进行修正，最后再利用期望值和修正概率做出最优决策

3. **最小风险贝叶斯决策理论**

   引入损失函数和期望损失

   损失函数：$\lambda_{ii}=\lambda(\alpha_i,\omega_i)$

   条件风险（条件期望损失）：
   $R(\alpha_i|x)=E[\lambda(\alpha_i,\omega)]=\sum_{i=1}^{M}\lambda(\alpha_i,\omega_j)P(\omega_j|x),i=1,2,\cdots,c(c\leqslant M)$

   规则：若$R(\alpha_k|x)=\min_{i=1,2,\ldots,M}R(\alpha_i|x)$，则判决$x\in\omega_k$

2.2 综合分析

2.3 证明计算

1. 贝叶斯决策论计算步骤
    1. 设定先验概率
    2. 通过给定的信息来设定条件概率
    3. 将先验概率转化为后验概率

4. 根据后验概率大小进行决策分类

设在某个局部地区细胞识别中正常 ω_1 和异常 ω_2 两类的先验概率分别为：
正常状态： $P(\omega_1)=0.9$
异常状态： $P(\omega_2)=0.1$
现有一待识别的细胞，其观察值为 x，从类条件概率密度分布曲线上查得

$$p(x|\omega_1)=0.2,\qquad p(x|\omega_2)=0.4$$

试使用贝叶斯决策对该细胞 x 进行分类（要求给出具体计算过程及计算结果）
解：

利用贝叶斯公式，分别计算出 ω_1 及 ω_2 的后验概率

$$P(\omega_1|x)=\frac{p(x|\omega_1)p(\omega_1)}{\sum_{j=1}^{2}p(x|\omega_j)p(\omega_j)}=\frac{0.2\times0.9}{0.2\times0.9+0.4\times0.1}=0.818$$

$$P(\omega_2|x)=1-P(\omega_1|x)=0.182$$

根据贝叶斯决策规则，有

$$P(\omega_1|x)=0.818>P(\omega_2|x)=0.182$$

所以合理的决策规则是把 x 归类于正常状态。

2. 增加条件λ11=0, λ12=6, λ21=1, λ22=0, 请判断该细胞是否正常

$$R(\alpha_1|\mathbf{x})=\sum_{i=1}^{2}\lambda_{1i}P(\omega_i|\mathbf{x})=1.092$$

$$R(\alpha_2|\mathbf{x})=\sum_{i=1}^{2}\lambda_{2i}P(\omega_i|\mathbf{x})=0.818$$

$\because R(\alpha_1|x)>R(\alpha_2|x)\therefore x\in$异常细胞(第2类)，因此决策$\omega_1$类风险大。
因λ_{12}=6较大，决策损失起决定作用。

三、最大似然估计和贝叶斯参数估计

考虑这样一个问题：总体X的概率密度函数为f(x|θ)，观测到一组样本(X1,X2,...,Xn)，需要估计参数θ，概率密度函数形式是已知

3.1 概念简答

1. **极大似然估计**
    1. 思想：待估计参数θ是客观存在的（本身常量），将样本的联合概率函数看成θ的函数，求其最大值
    2. 特点：

1. 比其他估计方法更加简单
2. 收敛性：无偏或者渐近无偏，当样本数目增加时，收敛性质会更好
3. 如果假设的类条件概率模型正确，则通常能获得较好的结果

2. **贝叶斯参数估计**

   据参数的先验分布P(θ)和一系列观察X，求出参数的后验分布P(θ|X)，然后求出的期望值，作为其最终值。简单来说，求已经发生结果，最大可能的条件是什么（后验）

3.2 证明计算

3. 极大似然估计过程
   1. 写出似然函数
   2. 对似然函数取对数，并整理
   3. 求导数
   4. 解似然方程
4. 贝叶斯参数估计过程

- 确定参数 θ 的先验概率密度函数 $p(\theta)$
- 由样本集$X=\{x_1,x_2\cdots x_N\}$ 求出样本联合概率密度函数 $p(X|\theta)$, 它是 θ 的函数

$$p(X|\theta)=\prod_{n=1}^{N}p(x_n|\theta)$$

- 利用贝叶斯定理, 求 θ 的后验概率密度函数

$$p(\theta|X)=\frac{p(X|\theta)p(\theta)}{\int p(X|\theta)p(\theta)d\theta}$$

- 求出贝叶斯估计值

$$\bar{\theta}=\int\theta p(\theta|X)d\theta$$

四、非参数估计

4.1 概念简答

1. 非参数估计

   非参数估计是指在不考虑原总体分布或者不作关于参数假定的前提下，直接用已知类别的学习样本的先验知识直接进行统计检验和判断分析的一系列方法的总称。非参数估计不假定数学模型，可避免对总体分布的假定不当导致重大错误所以常有较好的稳健性

直接用样本估计整个概率的分布，用频率逼近概率，划分小舱。这个估计当样本个数n非常大的时候将非常准确。如果我们假设px)是连续的，并且区域R足够小,以至于在这个区间中p几乎没有变化，那么有$\int_R p(x')dx' \approx p(x)V$

2. **KNN**

   （小舱大小可变，小舱内的样本数固定）

   所谓K最近邻，指每个样本都可以用它最接近的k个邻居来代表。kNN算法的核心思想是**如果一个样本在特征空间中的k个最相邻的样本中的大多数属于某一个类别，则该样本也属于这个类别，并具有这个类别上样本的特性**

   由于kNN方法主要靠周围有限的邻近的样本，而不是靠判别类域的方法来确定所属类别的，因此对于类域的交叉或重叠较多的待分样本集来说，kNN方法较其他方法更为适合

3. **Parzen 窗的原理**

   （小舱体积在全局处处保持不变）

   核密度（Parzen 窗）估计通过离散样本点来的线性加和来构建一个连续的概率密度函数，从而得到一个平滑的样本分布，可以看作是对直方图的一个自然拓展

4. 核函数应满足的性质
   1. 归一化：$\int_{+\infty}^{-\infty} K(u)du = 1$
   2. 对称性：对所有 uu 要求 $K(-u) = K(u)$

4.2 综合分析

5. **KNN算法过程**
   1. 计算测试数据与各个训练数据之间的距离
   2. 按照升序（从小到大）对距离（欧氏距离）进行排序
   3. 选取距离最小的前k个点
   4. 确定前k个点所在类别出现的频率
   5. 返回前k个点中出现频率最高的类别作为测试数据的分类
6. **KNN算法k值的选取**
   1. 当k的取值过小时，一旦有噪声得成分存在们将会对预测产生比较大影响，整体模型变得复杂，容易发生过拟合
   2. 如果k的值取的过大时，学习的近似误差会增大，整体的模型变得简单。与输入目标点较远实例也会对预测起作用，使预测发生错误
   3. K的取值尽量要取奇数，以保证在计算结果最后会产生一个较多的类别，如果取偶数可能会产生相等的情况，不利于预测
   4. 常用的方法是从k=1开始，估计分类器的误差率。重复该过程，每次K增值1，允许增加一个近邻，直到产生最小误差率的k
      一般k的取值不超过20，上限是n的开方，随着数据集的增大，K的值也 要增大
7. **Parzen 窗的窗口大小**
   - h过大，由于假定小舱内p(x)为常数，则导致过于平均的估计结果
   - h过小，落入小舱的样本将会很少，或者没有样本落入，从而导致对p(x)的估计不连续（刺头）

4.3 证明计算

8. 简述 Parzen 窗的原理与过程，证明为什么可以用于高斯函数

五、线性判别函数

5.1 概念简答

1. **梯度**

   梯度的是一个向量（矢量），表示某一函数在该点处的方向导数沿着该方向取得最大值，即函数在该点处沿着该方向（此梯度的方向）变化最快，变化率最大（为该梯度的模）。

   设二元函数$z=f(x,y)$在平面区域D上具有一阶连续偏导数，则对于每一个点P（x，y）都可定出一个向量$\frac{\partial f}{\partial x},\frac{\partial f}{\partial y}=f_x(x,y)\overline{i}+f_y(x,y)\overline{j}$，该函数就称为函数z在点P(x，y)的梯度，记作$grad f(x，y)$或$\nabla f(x,y)$

2. **学习率**

   学习率是优化算法中的一个可调参数，它决定了每次迭代的步长，使得跌打向损失函数的最小值前进。它影响到新学习到的信息在多大程度上取代了旧信息，暗示了机器学习模型 "学习 "的速度。

3. **梯度下降法原理**

   （一阶优化算法）通过搜索方向和步长来对参数进行更新。其中搜索方向是目标函数在当前位置的负梯度方向。因为这个方向是最快的下降方向。步长确定了沿着这个搜索方向下降的大小。

4. **牛顿法原理**

   （二阶优化算法）牛顿法是求解函数值为0时的自变量取值的方法，利用牛顿法求解目标函数的最小值其实是转化成求使目标函数的一阶导为0的参数值（这一转换的理论依据是，函数的极值点处的一阶导数为0）。其迭代过程是在当前位置x0求该函数的切线，该切线和x轴的交点x1，作为新的x0,重复这个过程，直到交点和函数的零点重合。此时的参数值就是使得目标函数取得极值的参数值。

5.2 综合分析

3. **牛顿法和梯度下降法的比较**

梯度下降法	牛顿法	备注
最优值附近震荡		接近最优值时不断减少步长
	远离局部极小点可能不会收敛	先得到离最优点较近的点
	计算量、内存代价大	使用拟牛顿法
	计算速度快	二阶函数

4. **学习率过大或过小有什么后果？如何调整学习率**

   学习率设置太大，参数更新的幅度就非常大，在最优值附近徘徊，或者loss开始增加；学习率设置太小，网络收敛非常缓慢，会增大找到最优值的时间，且可能会进入局部极值点就收敛

   在训练过程中，一般根据训练轮数设置动态变化的学习率。刚开始训练时，学习率以 0.01 ~ 0.001 为宜；一定轮数过后，逐渐减缓；接近训练结束，学习速率的衰减应该在100倍以上

5.3 证明计算

4. **证明为什么梯度下降算法可以确保是下降的/为什么梯度下降选择负梯度优化目标函数/（为何负梯度是函数值减小的最快方向）**

   假设我们的object function为：$arg\min_w\ f(w)$，为了求得最优解，可以这样变形：
   $min = f(w + step_w) - f(w)$达到最小，利用泰勒展开公式有：
   $f(w + step_w) = f(w) + f'(w) * step_w$
   $f(w + step_w) - f(w) = f'(w) * step_w$
   为了得到min，我们只要使w的更新步长$step_w = -f'(w)$即可（这保证了函数一定不增，且梯度是函数增长最快的方向，取负为下降最快）
   这样就得到了梯度下降法的公式：
   $w = w - \alpha * f'(w)$

六、神经网络

6.1 概念简答

1. 基本名词
   1. 神经元（Neuron）：神经元形成神经网络的基本结构。在神经网络的情况下，神经元接收输入，处理它并产生输出，而这个输出被发送到其他神经元用于进一步处理，或者作为最终输出进行输出。
   2. 权重（Weights）：每个神经元具有分配给它的一个关联权重，反映其重要程度。算法中随机初始化权重，并在模型训练过程中更新这些权重
   3. 偏差（Bias）：改变权重与输入相乘所得结果的范围的线性分量
   4. **激活函数（Activation Function）**：一种添加到人工神经网络中的函数，旨在帮助网络学习数据中的复杂模式，将线性函数转变为非线性的

      常用激活函数：$Sigmoid(z) = \frac{1}{1+e^{-z}}$、$\tanh(x) = \frac{e^x-e^{-x}}{e^x+e^{-x}}$
   5. **神经网络（Neural Network）**：神经网络由许多相互关联的概念化的人造神经元组成，它们之间传递相互数据，并且具有根据网络"经验"调整的相关权重。神经元具有激活阈值，如果通过其相关权重的组合和传递给他们的数据满足这个阈值的话，其将被解雇；发射神经元的组合导致"学习"。

      神经网络是一种人工智能方法，用于教计算机以受人脑启发的方式处理数据。是深度学习算法的核心，使用类似于人脑的分层结构中的互连节点或神经元。它可以创建自适应系统，计算机使用该系统来从错误中进行学习并不断改进。因此，人工神经网络可以尝试解决复杂的问题，例如更准确地总结文档或人脸识别。

由节点层组成，包含一个输入层、一个或多个隐藏层和一个输出层。 每个节点也称为一个人工神经元，它们连接到另一个节点，具有相关的权重和阈值

6. 输入/输出/隐藏层（Input / Output / Hidden Layer）：输入层是接收输入那一层，本质上是网络的第一层。而输出层是生成输出的那一层，也可以说是网络的最终层。处理层是网络中的隐藏层。这些隐藏层是对传入数据执行特定任务并将其生成的输出传递到下一层的那些层。输入和输出层是我们可见的，而中间层则是隐藏的
7. **多层感知器（MLP）**：是一种前向结构的人工神经网络，映射一组输入向量到一组输出向量。MLP可以被看作是一个有向图，由多个的节点层所组成，每一层都全连接到下一层。除了输入节点，每个节点都是一个带有非线性激活函数的神经元（或称处理单元）
8. 正向传播（Forward Propagation）：正向传播是指输入通过隐藏层到输出层的运动。在正向传播中，信息沿着一个单一方向前进。输入层将输入提供给隐藏层，然后生成输出。这过程中是没有反向运动的
9. 损失函数（Cost Function）：当我们建立一个网络时，网络试图将输出预测得尽可能靠近实际值。我们使用成本/损失函数来衡量网络的准确性
10. 反向传播（Backpropagation）：当我们定义神经网络时，我们为我们的节点分配随机权重和偏差值。一旦我们收到单次迭代的输出，我们就可以计算出网络的误差。然后将该误差与成本函数的梯度一起反馈给网络以更新网络的权重。 最后更新这些权重，以便减少后续迭代中的误差。使用成本函数的梯度的权重的更新被称为反向传播

2. **描述BP算法**

   反向传播算法主要由两个环节（激励传播、权重更新）反复循环迭代，直到网络的对输入的响应达到预定的目标范围为止。

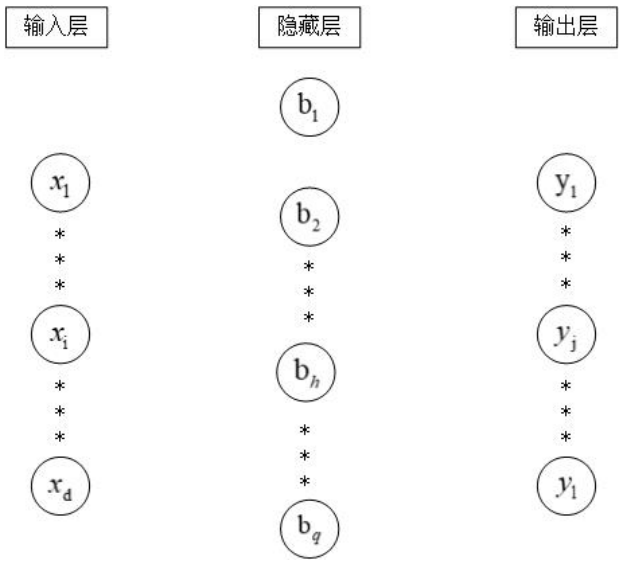
   BP算法的学习过程由正向传播过程和反向传播过程组成。在正向传播过程中，输入信息通过输入层经隐含层，逐层处理并传向输出层。如果在输出层得不到期望的输出值，则取输出与期望的误差的平方和作为目标函数，转入反向传播，逐层求出目标函数对各神经元权值的偏导数，构成目标函数对权值向量的梯量，作为修改权值的依据，网络的学习在权值修改过程中完成。误差达到所期望值时，网络学习结束

3. **奥卡姆剃刀原理**

   意为“简约法则”，如果关于同一个问题有许多种理论，每一种都能作出同样准确的预言，那么应该挑选其中使用假定最少的

6.2 证明计算

4. BP算法的相关推导



1. 前向传播

$\alpha_h = \sum_{i=1}^{d} v_{ih} x_i$

$b_h = sigmoid(\alpha_h)$

$\beta_j = \sum_{h=1}^{q} w_{hj} b_h$

$\hat{y}_i = sigmoid(\beta_i)$

$E = \frac{1}{2} \sum_{i=1}^{l} (\hat{y}_j - y_j)^2$

2. 方向传播

$\triangle w_{hj} = -\eta \frac{\partial E}{\partial w_{hj}}$其中$\frac{\partial E}{\partial w_{hj}} = \frac{\partial E}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial \beta_j} \frac{\partial \beta_j}{\partial w_{hj}}$

1. $\frac{\partial E}{\partial \hat{y}_j} = -(\hat{y}_j - y_j)$
2. $\frac{\partial \hat{y}_j}{\partial \beta_j} = \hat{y}_j (1 - \hat{y}_j)$
3. $\frac{\partial \beta_j}{\partial w_{hj}} = b_h$

$w_{hj} = w_{hj} + \Delta w_{hj} = w_{hj} - \eta(-(\hat{y}_j - y_j)\hat{y}_j(1 - \hat{y}_j) b_h)$

同理:

$$\frac{\partial E}{\partial v_{ih}} = \frac{\partial E}{\partial b_h} \frac{\partial b_h}{\partial \alpha_h} \frac{\partial \alpha_h}{\partial v_{ih}} = \sum_{j=1}^{l} \left(\frac{\partial E}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial \beta_j} \frac{\partial \beta_j}{\partial b_h} \right) \frac{\partial b_h}{\partial \alpha_h} \frac{\partial \alpha_h}{\partial v_{ih}}$$

$$= \sum_{j=1}^{l} [-(-(\hat{y}_j - y_j)\hat{y}_j(1 - \hat{y}_j) w_{hj}][b_h(1 - b_h)][x_i])$$

七、决策树

7.1 概念简答

1. **决策树分类原理**

   决策树是通过 一系列规则 对数据进行分类的过程。它提供一种在什么条件下会得到什么值的类似规则的方法。决策树分为分类树和回归树两种，分类树对离散变量做决策树，回归树对连续变量做决策树。一棵决策树的生成过程主要分为以下3个部分：特征选择、决策树生成、剪枝

2. 信息论基础

   1. 假定当前样本集合D中第k 类样本所占的比例为p_k(k = 1,2,... ，n)（假设共n类），则D的信息熵定义为：$Ent(D) = -\sum_{k=1}^{n} p_k \log_2 p_k$，Ent(D)的值越小，则D的纯度越高
   2. **信息增益**：假定离散属性a有V个可能的取值$a^1, a^2, \cdots, a^V$，若使用a来对样本集D 进行划分，则会产生V个分支结点，其中第v个分支结点包含了 D 中所有在属性a 上取值为淤的样本，记为D^v，则用属性a 对样本集D 进行划分所获得的“信息增益”为：$Gain(D,a) = Ent(D) - \sum_{v=1}^{V} \frac{|Dv|}{|D|} Ent(D^v)$，一般而言，信息增益**越大**，则划分所获得的“**纯度**提升”**越大**，因此，我们选择属性$a_* = \arg\max_{a \in A} Gain(D,a)$划分（ID3）
   3. 增益率：$Gain\_ratio(D,a) = \frac{Gain(D,a)}{IV(a)}$，其中$IV(a) = -\sum_{i=1}^{V} \frac{|Dv|}{|D|} \log_2 \frac{|Dv|}{|D|}$，称为属性a 的“固有值"（C4.5：先从候选划分属性中找出信息增益高于平均水平的属性，再从中选择增益率最高的）
   4. 基尼系数：$Gini(D) = \sum_{k=1}^{n} \sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{i=1}^{n} p_k^2$，Gini(D)反映了从数据集D中随机抽取两个样本，其类别标记不一致的概率，因此，Gini(D)越小，则数据集D的纯度越高。单一属性a 的基尼指数定义为：$Gini\_index(D,a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} Gini(D^v)$，选择$a_* = \arg\min_{a \in A} Gain(D,a)$为划分属性（CART，既可分类，又可回归）

7.2 综合分析

3. **ID3、C4.5、CART比较**

   1. ID3的不足

      1. ID3没有考虑连续特征
      2. ID3采用信息增益大的特征优先建立决策树的节点（在相同条件下，取值比较多的特征比取值少的特征信息增益大）
      3. ID3算法对于缺失值的情况没有做考虑
      4. 没有考虑过拟合的问题

   2. **C4.5对ID3的改进**

      1. 将连续的特征离散化：对m个样本排序，取相邻两样本值的平均数，计算m-1个作为二元分类点的信息增益，选择最大的点作为连续特征的分类点
      2. 引入信息增益比，可以校正信息增益容易偏向于取值较多的特征的问题
      3. 解决样本某些特征缺失的情况下选择划分的属性，以及对于在该属性上缺失特征的样本的处理的问题：

        1. 将特征划分，对于没有缺失特征的数据集来和对应的特征的各个特征值一起计算加权重后的信息增益比，最后乘上无缺失样本加权后所占加权总样本的比例
        2. 可以将缺失特征的样本同时划分入所有的子节点，并将该样本的权重按各个子节点样本的数量比例来分配
    4. 引入了正则化系数进行初步的剪枝
3. C4.5的不足
    1. C4.5的剪枝方法是PEP，使用的从上而下的剪枝策略，会造成剪枝过度，并且会出现剪枝失败的情况
    2. C4.5生成的是多叉树
    3. C4.5只能用于分类
    4. C4.5由于使用了熵模型，里面有大量的耗时的对数运算
4. CART对C4.5的改进
    1. 使用CCP代价复杂度剪枝算法
    2. 采用的是不停的二分，找到基尼系数最小的组合，离散特征会参与多次节点建立
    3. 两种树的不同在于连续值的处理方法和树建立后做预测方式
    4. 使用基尼系数作为度量方式
5. CART的不足
    1. 大多数，分类决策不应该是由某一个特征决定的，而是应该由一组特征决定的
    2. 样本发生一点改动，就会导致树结构的剧烈改变
4. **如何解决决策树构造过程中会出现过拟合现象**
    1. 有效地抽样，用相对能够反映业务逻辑的训练集去产生决策树
    2. 提前停止树的增长或者对已经生成的树按照一定的规则进行后剪枝

7.3 证明计算

5. 采用信息增益的思想选择属性

八、集成学习

8.1 概念简答

1. 集成学习

   集成学习（Ensemble Learning）是解决有监督机器学习任务的一类方法，通过有策略地生成和组合多个弱学习器，合成强学习器来完成学习任务，提升预测结果。分为同质集成和异质集成、并行集成和串行集成。结合策略主要有平均法、投票法和学习法等

2. **Bagging**

    0. 也称为袋装法，是最为经典的并行集成算法，对原始训练数据集进行有放回的重复抽样达到目的。少数服从多数的原则集成多个分类器结果，或对多个分类器的预测值求平均作为最终的预测结果
    1. 输入：包含n个样本的训练数据集D、基分类器算法f、设定基分类器构建个数N
    2. for i in 1:N do
    3. 从初始数据集D中有放回的抽取n个样本，生成自助训练集Di

4. 基于训练集Di，训练获得基分类器fi
5. end for
6. 输出

3. **Boosting(AdaBoost)**

0. 将较弱的分类器逐步提升为强的分类器，下一个分类器将根据上一个分类器的预测结果对样本的权重进行调整，对于错判样本将给予更大的权重，从而使得新的分类器更加关注错误样本的预测，在最终的集成模型中，性能更好的模型具有更高的权重
1. 输入：包含n个样本的训练数据集D、基分类器算法f、设定基分类器构建个数（训练轮次）N
2. 初始化样本权重w1={w1k=1/n, k=1,2,...n}
3. for i in 1:N do
4. 基于训练集D，训练获得基分类器fi
5. 计算基分类器fi的加权误差：$\varepsilon_t = \sum_{k-1}^{n} \omega_{ik} \varphi(f_i(x_k) \neq y_k)$
6. if εi >0.5 then
7. 重新初始化样本权重wi
8. 返回步骤4
9. end if
10. 计算分类器权重：$\alpha_i = \frac{1}{2} \ln\left(\frac{1-\varepsilon_i}{\varepsilon_i}\right)$
11. 更新下一轮的抽样权重：$\omega_{(it1)k} = \frac{\omega_{ik}}{Z_i} \times \begin{cases} e^{-\alpha_i}, \text{if } f_i(x_k) = y_k \\ e^{\alpha_i}, \text{ if } f_i(x_k) \neq y_k \end{cases}$，其中$Z_i$为规范化因为，保证和等于1，$Z_i = \sum_{i=1}^{n} \omega_{ik} \times \begin{cases} e^{-\alpha_i}, \text{if } f_i(x_k) = y_k \\ e^{\alpha_i}, \text{ if } f_i(x_k) \neq y_k \end{cases}$
12. end for
13. 组合各个弱分类器得到强分类器：$F(x) = sign(\sum_{k=1}^{n} \alpha_k f_k(x))$

8.2 综合分析

4. **Bagging和Boosting的区别**

	Bagging	Boosting
样本选择	有放回，各轮训练集之间是独立	训练集不变，权重改变
样例权重	均匀取样，权重相等	错误率越大则权重越大
预测函数	所有预测函数的权重相等	分类误差小的分类器会有更大的权重
计算方式	并行	串行
侧重方向	降低方差，有利于不稳定分类器	减小偏差，强化弱分类器

5. 从期望损失的角度分析 adaboost 的合理性，可从分布和分类器权重更新方面阐述

8.3 证明计算

6. **Adaboost算法α_i推导**

基分类器h_i的权重 α_t 应使得 $\alpha_t h_t$ 最小化指数损失函数：(D：权重w，h：分类器f)

$$\begin{aligned}\ell_{\exp}\left(\alpha_t h_t \mid \mathcal{D}_t\right) &= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}\left[e^{-f(\boldsymbol{x})\alpha_t h_t(\boldsymbol{x})}\right] \\ &= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}\left[e^{-\alpha_t}\mathbb{I}\left(f(\boldsymbol{x})=h_t(\boldsymbol{x})\right)+e^{\alpha_t}\left[f(\boldsymbol{x})\neq h_t(\boldsymbol{x})\right]\right] \\ &= e^{-\alpha_t}P_{x\sim D_t}\left(f(\boldsymbol{x})=h_t(\boldsymbol{x})\right)+e^{\alpha_t}P_{x\sim D_t}\left(f(\boldsymbol{x})\neq h_t(\boldsymbol{x})\right) \\ &= e^{-\alpha_t}\left(1-\epsilon_t\right)+e^{\alpha_t}\epsilon_t\end{aligned} \tag{1.8}$$

其中，$\epsilon_t = P_{\boldsymbol{x}\sim\mathcal{D}_t}\left(h_t(\boldsymbol{x})\neq f(\boldsymbol{x})\right)$，也即是第 t 轮迭代时，真实函数与该学习器学习到的数据不同的概率。为了使得损失函数最小化，同样的，我们对其求偏导：

$$\frac{\partial\ell_{\exp}\left(\alpha_t h_t \mid \mathcal{D}_t\right)}{\partial\alpha_t} = -e^{-\alpha_t}\left(1-\epsilon_t\right)+e^{\alpha_t}\epsilon_t = 0 \tag{1.9}$$

由此，我们不难得到：

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \tag{1.10}$$

这既是图 2.1 中的分类器的权重更新公式。

7. 权重推导

九、支持向量机SVM

9.1 概念简答

1. **线性可分**

线性可分指的是可以用一个线性函数将两类样本分开（无误差），比如在二维空间中的直线、三位空间中的平面以及高维空间中的线性函数

2. **支持向量机**

支持向量机在高维或无限维空间中构造超平面或超平面集合，其可以用于分类、回归或其他任务。SVM是一种二类分类模型，他的基本模型是的定义在特征空间上的**间隔最大**的线性分类器，SVM的学习策略就是间隔最大化

给定一组训练实例，每个训练实例被标记为属于两个类别中的一个或另一个，SVM训练算法建立一个将新的实例分配给两个类别之一的模型，使其成为非概率二元线性分类器。SVM模型是将实例表示为空间中的点，这样映射就使得单独类别的实例被尽可能宽的明显的间隔分开。然后，将新的实例映射到同一空间，并基于它们落在间隔的哪一侧来预测所属类别

除了进行线性分类之外，SVM还可以使用所谓的核技巧有效地进行非线性分类，将其输入隐式映射到高维特征空间中

3. SVM模型由简至繁的分类

线性可分支持向量机：训练数据线性可分，通过硬间隔最大化，学习一个线性的分类器

线性支持向量机：训练数据近似线性可分，通过软间隔最大化，学习一个线性的分类器

非线性支持向量机：训练数据线性不可分，通过核技巧和软间隔最大化，学习一个非线性的分类器

4. VC维

   在VC理论中，VC维是对一个可学习分类函数空间的能力（复杂度，表示能力等）的衡量。它定义为算法能“打散”的点集的势的最大值

5. **间隔**

   边界在到达数据点之前可以增加的宽度

   硬间隔：完全分类准确，其损失函数不存在；其损失值为0；只要找出两个异类正中间的那个平面

   软间隔：允许一定量的样本分类错误；优化函数包括两个部分，一部分是点到平面的间隔距离，一部分是误分类的损失个数；C是惩罚系数，误分类个数在优化函数中的权重值，权重值越大，误分类的损失惩罚的越厉害

6. **核函数**

   将样本从原始的特征空间映射到一个高维的特征空间，以使得样本在高维特征空间中仍然线性可分，可以用低维空间的核函数计算代替高维空间复杂的的内积运算

   设x是输入空间(欧式空间R^n的子集或离散集合)，又设H是特征空间(希尔伯特空间)，如果存在一个X到H的映射$\phi(x)：x \to H$使得对所有$x, z \in X$，函数$K(x, z)$满足条件$K(x, z) = \phi(x) \cdot \phi(z)$则称$K(x, z)$为核函数，$\phi(x)$为映射函数，式中$\phi(x) \cdot \phi(z)$为$\phi(x)$和$\phi(z)$的内积。

9.2 综合分析

7. **针对线性不可分问题，SVM 有哪些方法，简要描述**

   1. 加入松弛变量和惩罚因子，找到相对“最好”超平面，尽可能地将数据正确分类
   2. 使用核函数，将低维的数据映射到更高维的空间，使得在高维空间中数据是线性可分的，在高维空间使用线性分类模型

9.3 证明计算

8. **线性可分SVM计算步骤**

   1. **间隔最大化**

      求解最大间隔超平面（两直线距离公式），即求：

      $$\max_{w,b} \frac{2}{||w||} = min \frac{1}{2}||w||$$

      $$s.t. \qquad y_i(w^T x + b) \geq 1, \forall i$$

   2. **推导目标函数的对偶形式**

      1. 建拉格朗日函数，引进拉格朗日乘子：

      $$L(W, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i [y_i(w^T x_i + b) - 1]$$

优化问题为：

$$\min_{w,b}\max_{\alpha} L(w,b,\alpha)$$

对偶问题为：

$$\max_{\alpha}\min_{w,b} L(w,b,\alpha)$$

$$s.t. \quad \nabla_{w,b}L(w,b,\alpha)=0$$

$$\alpha \geq 0$$

2. 求$\min\limits_{w,b} L(w,b,\alpha)$，令导数等于0：

$$\nabla_w L(w,b,\alpha)=w-\sum_{i=1}^{n}\alpha_i y_i x_i=0 \Rightarrow w^*=\sum_{i=1}^{n}\alpha_i y_i x_i$$

$$\nabla_b L(w,b,\alpha)=\sum_{i=1}^{n}\alpha_i y_i=0$$

3. 代入拉格朗日函数：

$$\begin{aligned}L(w,b,\alpha)&=\frac{1}{2}|w|^2-\sum_{i=1}^{n}\alpha_i[y_i(w^Tx_i+b)-1]\\&=\frac{1}{2}\sum_{i=1}^{n}\alpha_i y_i x_i^T\sum_{j=1}^{n}\alpha_j y_j x_j-\sum_{i=1}^{n}\alpha_i y_i x_i^T\sum_{j=1}^{n}\alpha_j, y_j x_j-b\sum_{i=1}^{n}\alpha_i y_i+\sum_{i=1}^{n}\alpha_i\\&=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\alpha_i y_i x_i^T\sum_{j=1}^{n}\alpha_j, y_j x_j\\&=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}y_i y_j\alpha_i\alpha_j x_i^T x_i\end{aligned}$$

即：$L(w,b,\alpha)=-\frac{1}{2}\sum_{i,j=1}^{n}y_i y_j\alpha_i\alpha_j x_i^T x_i+\sum_{i=1}^{n}\alpha_i$

4. 求$\min_{w,b} L(w,b,\alpha)$对α的极大，等价于求：

$$\min_{\alpha}\frac{1}{2}\sum_{i,j=1}^{n}y_i y_j\alpha_i\alpha_j x_i^T x_j-\sum_{i=1}^{n}\alpha_i$$

$$s.t. \quad \sum_{i=1}^{n}\alpha_i y_i=0$$

$$\alpha_i\geq 0,\forall i$$

至此，我们得到了原始最优化问题和对偶最优化问题

3. **决策函数的解**

假设得到了对偶i问题的最优解α^*，则：$w^*=\sum_{i=1}^{n}\alpha_i^* y_i x_i$

由题设，至少存在一个$\alpha_j^*>0$，根据KKT条件，至少存在一个j，使得$y_j(w^{*T}x_j+b^*)-1=0$，即可求得最优b^*：

$b^*=y_j-w^{*T}x_j$

代回原式可得决策函数为：

$$f(X) = sgn(\sum_{i=1}^{n} \alpha_i^* y_i x^T x_i + b^*)$$

9. 线性SVM计算步骤

1. 间隔最大化，引入惩罚项和松弛变量

$$\max_{w,b} \frac{2}{||w||} = min \frac{1}{2}||w|| + C\sum_{i=1}^{n}\xi_i$$

$$s.t. \qquad y_i(w^T x + b) \geq 1 - \xi_i$$

$$\xi_i \geqslant 0, \forall i$$

2. 对偶形式

$$L(W, b, \xi, \alpha, \beta) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i[y_i(w^T x_i + b) - 1 + \xi_i] - \sum_{i=1}^{n}\beta_i\xi_i$$

$$\max_{\alpha,\beta} \min_{w,b,\xi} L(w, b, \xi, \alpha, \beta)$$

$$s.t. \quad \nabla_{w,b,\xi} L(w, b, \xi, \alpha, \beta) = 0$$

$$\alpha \geq 0, \beta \geq 0$$

$$\nabla_w L(w, b, \xi, \alpha, \beta) = w - \sum_{i=1}^{n}\alpha_i y_i x_i = 0 \Rightarrow w^* = \sum_{i=1}^{n}\alpha_i y_i x_i$$

$$\nabla_b L(w, b, \xi, \alpha, \beta) = \sum_{i=1}^{n}\alpha_i y_i = 0$$

$$\nabla_{\xi_i} L(w, b, \xi, \alpha, \beta) = C - \alpha_i - \beta_i = 0 \Rightarrow \beta_i = C - \alpha_i$$

$$\min_{\alpha} \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j - \sum_{i=1}^{n}\alpha_i$$

$$s.t. \qquad \sum_{i=1}^{n}\alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C, \forall i$$

3. 决策函数的解

保持一致

10. 非线性SVM计算步骤

对偶问题：

$$\min_{\alpha} \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \sum_{i=1}^{n}\alpha_i$$

$$s.t. \qquad \sum_{i=1}^{n}\alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C, \forall i$$

十、线性回归

10.1 概念简答

1. 线性回归

   利用数理统计中回归分析，来确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法

2. 逻辑回归

   一种广义线性回归，决策边界是一条直线或在高维是超平面

3. 线性模型

   基本形式：$f(x_i)=w_1x_{i1}+w_2x_{i2}+\cdots+w_dx_{id}+b$

   向量形式：$f(x_i)=w^Tx_i=X\widehat{\omega}$

4. 最小二乘法

   损失函数：$E_w=(y-X\widehat{w})^T(y-X\omega)$

   解析解：$\widehat{w}=(X^TX)^{-1}X^Ty$

5. 岭回归

   损失函数：$E_w=(y-X\widehat{w})^T(y-X\omega)+\lambda||\omega||_2^2\ \lambda>0$

   解析解：$\widehat{w}=(X^TX+\lambda I)^{-1}X^Ty$

6. 套索回归

   损失函数：$E_w=(y-X\widehat{w})^T(y-X\omega)+\lambda||\omega||_1\ \lambda>0$

十一、聚类

见数据挖掘