题型：

1. 单词解释 2\*10
2. 解释概念 2\*10
3. 综合题
4. 某算法的实例分析过程。计算题
5. 写出某算法的输入、输出、伪代码。
6. 写出某类算法的主要思想。

了解手动开平方根怎么计算

一、

1. Data Mining 数据挖掘 从广义的观点，数据挖掘是从大型数据集（可能是不完全的、有噪声的、不确定性的、各种存储形式的）中，挖掘隐含在其中的、人们事先完全不知道的、对决策有用的知识的完整过程。 从狭义的观点，我们可以定义数据挖掘是从特定形式的数据集中提炼知识的过程。
2. Artificial Intelligence 人工智能
3. Machine Learning 机器学习
4. Knowledge Engineering 知识工程
5. Information Retrieval 信息检索
6. Data Visualization 数据可视化
7. OLTP（On-line Transaction Progressing） 联机事务处理
8. OLAP（On-line Analytic Progressing） 联机分析处理
9. Decision Support 决策支持
10. KDD（Knowledge Discovery in Database）数据库中的知识发现
11. Transaction Database 事务数据库
12. Distributed Database 分布式数据库
13. KDD的步骤
14. 问题定义阶段

数据挖掘人员必须和领域专家以及最终用户紧密协作，一方面，了解相关领域的有关情况，熟悉背景知识，弄清用户要求，确定挖掘的目标等；另一方面，通过对各种学习算法的对比进而确定可用的学习算法。

1. 数据抽取阶段

选取相应的源数据库，并根据要求从数据库中提取相关的数据。

1. 数据预处理阶段

主要对前一阶段抽取的数据进行再加共，检查数据的完整性及数据的一致性。

1. 数据挖掘阶段

运用选定的数据挖掘算法，从数据中提取出用户所需要的知识，这些知识可以用一种特定的方式表示。

两个考虑因素：1、不同数据有不同的特点，需要用与之相关的算法来挖掘。2、用户或实际运行系统的要求

1. 知识评估阶段

发现出来的模式，经过评估，可能存在冗余或无关的模式，这时需要将其剔除；或者模式不满足用户要求，退回前续阶段，甚至更换算法。

三、

1. 关联规则频繁项目集等概念、两步骤
2. 频繁项目集：对项目及 I 和事务数据库 D， T中所有满足用户指定的最小支持度的项目集，即大于等于最小支持度的 I 的非空子集，称为频繁项目集或者大项目集。
3. 最大频繁项目集：在频发项目集中挑选出所有不被其他元素包含的频繁项目集
4. 关联规则：一个定义在I 和 D 上的形如 I1 => I2 的关联规则通过满足一定的可信度、信任度或置信度来给出。所谓规则的可信度是指包含I1和I2的事务数与包含I1的事务数之比。

两步骤：1、发现频繁项目集。2、生成关联规则。

1. Apriori算法
2. 算法思想：Apriori算法的基本思想是通过对数据库的多次扫描来计算项集的支持度，发现所有的频繁项集从而生成关联规则。Apriori算法对数据集进行多次扫描，第一次扫描得到频繁1-项集L1，第k(k>1)次扫描首先利用第（k-1）次扫描的结果L(k-1)来产生候选k-项集的集合Ck，然后在扫描过程中确定Ck中元素的支持度，最后在每一次扫描结束时计算频繁k-项集的集合Lk，算法在当候选k-项集的集合Ck为空时结束。
3. 计算：

计算关联规则的可信度 可信度（I1=>I2） = 支持度（I1和I2）/ 支持度（I1）

计算频繁项目集 大于最小支持度

1. Close算法
2. 算法思想（原理）：一个频繁闭合项目集的所有闭合子集一定是频繁的，一个非频繁闭合项目集的所有闭合超集一定是非频繁的。因此，可以在闭合项目集格空间上讨论项目集的频繁问题。
3. 计算：

计算闭包：闭包是包含它的全部集合的交集。emm姑且认为所有子集都需要去掉

计算L1=>L2项目集：

1. FP-tree算法
2. 算法思想（原理）：进行2次数据库扫描：一次对所有1-项目的频度排序；一次将数据库信息转变成紧缩内存结构。不使用侯选集，直接压缩数据库成一个频繁模式树，通过频繁模式树可以直接得到频集。
3. 计算：

构造树：

模式挖掘：

[FPTree 理解\_zhangbcn的专栏-CSDN博客](https://blog.csdn.net/zhangbcn/article/details/82964742?ops_request_misc=&request_id=&biz_id=102&utm_term=FP-tree%E7%AE%97%E6%B3%95%E8%A7%A3%E6%9E%90&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-82964742.pc_search_mgc_flag&spm=1018.2226.3001.4187)

构造FP-tree和从树上挖掘频繁项目集和关联规则

四、

1. 数据分类的两个步骤
2. 建立一个模型，描述预定的数据类集或概念集。

通过分析由属性描述的数据库元组来构造模型。数据元组也称作样本、实例或对象。为建立模型而被分析的数据元组形成训练数据集。

1. 使用模型进行分类。
2. 基于距离的分类算法KNN
3. 算法思想（原理）：假定每个类包含多个训练数据，且每个训练数据都有一个唯一的类别标记，k-最临近分类的主要思想就是计算每个训练数据到待分类元组的距离，取和待分类元组距离最近的k个训练数据,k个数据中哪个类别的训练数据占多数，则待分类元组就属于哪个类别。
4. 计算：

计算距离：欧式距离

1. 伪代码：

4-1基于距离的类标识搜索算法

输入：每个类的中心C1，C2,C3....，Cm；待分类的元组t

输出：输出类别c。

dist=∞; //距离初始化

FOR i=1 to m DO

IF dis(ci,t) < dist THEN BEGIN

C = i;

dist = dist(ci,t);

END

flag t with c

4-2 K-最临近算法

输入：训练数据集T

最临近数目k

待分类的元组t

输出： 输出类别c。

N = Ø；

FOR each d ∈ T DO BEGIN

IF |N| ≤ k THEN

N = N ∪ {d};

ELSE

IF ə u ∈ N such that sim(t,u) < sim(t,d) THEN

BEGIN

N = N - {u};

N = N ∪{d};

END

END

c= class related to such u∈N which has the most number;

1. 决策树基本算法
2. 算法思想（原理）：

步骤：

1. 以代表训练样本的单个结点开始建树。
2. 如果样本都在同一个类，则该结点成为树叶，并用该类标记。
3. 否则，算法使用称为信息增益的基于熵的度量作为启发信息，选择能够最好地将样本分类的属性。该属性成为该结点的“测试”或“判定”属性。
4. 对测试属性的每个已知的值，创建一个分支，并据此划分样本。
5. 算法使用同样的过程，递归地形成每个划分上的样本决策树。一旦一个属性出现在结点上，就不必考虑该结点的任何后代。
6. 递归划分步骤，当下列条件之一成立时停止。：

①给定结点的所有样本属于同一类。

②没有剩余属性可以用来进一步划分样本。在此情况下，采用多数表决。

③分支test\_attribute=ai 没有样本。在这种情况下，以Samples中的多数类创建一个树叶

1. 概念解释：
2. 熵：熵是对事件对应的属性的不确定性的度量。一个属性的熵越大，它蕴含的不确定信息越大，越有利于数据的分类。
3. 信息增益：

[决策树算法：ID3\_未名湖畔的落叶-CSDN博客\_id3算法](https://blog.csdn.net/keyue123/article/details/82253538?ops_request_misc=%7B%22request%5Fid%22%3A%22163841301316780271568237%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=163841301316780271568237&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-82253538.pc_search_mgc_flag&utm_term=ID3%E7%AE%97%E6%B3%95&spm=1018.2226.3001.4187)

信息增益的公式、熵的计算等

4.贝叶斯分类描述过程思想计算 离散百分比 先验概率后验概率

1.概念：

设X是类标号未知的数据样本。设H为某种假定，如数据样本X属于某特定类C。对于分类问题，我们希望确定P（H|X），即给定观测数据样本X，假定H成立的概率。

P（H|X）=P（X|H）\* P（H） / P（X）

先验概率： P（H） 或称H的先验概率

后验概率： P（H|X）是后验概率，或称条件X下H的后验概率。

离散百分比：？

1. 朴素贝叶斯分类过程、计算：

概念：对于给出的待分类项，求解在此项出现的条件下各个类别出现的概率，哪个最大，就认为此待分类项属于哪个类别。

过程：[贝叶斯分类（这个讲的比较清晰，一看就明白）\_a925907195的专栏-CSDN博客](https://blog.csdn.net/a925907195/article/details/53822830?ops_request_misc=%7B%22request%5Fid%22%3A%22163842404016780366598692%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=163842404016780366598692&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-53822830.pc_search_mgc_flag&utm_term=%E8%B4%9D%E5%8F%B6%E6%96%AF%E5%88%86%E7%B1%BB&spm=1018.2226.3001.4187)

1. EM算法的思想（原理）、步骤：

[如何通俗理解EM算法\_结构之法 算法之道-CSDN博客\_em算法](https://blog.csdn.net/v_JULY_v/article/details/81708386?ops_request_misc=%7B%22request%5Fid%22%3A%22163842488716780261981846%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=163842488716780261981846&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-81708386.pc_search_mgc_flag&utm_term=EM%E7%AE%97%E6%B3%95&spm=1018.2226.3001.4187)

步骤：

１.第一步是计算期望（E），利用对隐藏变量的现有估计值，计算其最大似然估计值；

２.第二步是最大化（M），最大化在E步上求得的最大似然值来计算参数的值。M步上找到的参数估计值被用于下一个E步计算中，这个过程不断交替进行。

1. 分类器性能的表示和评估
2. 保持法：把给定的数据随机地划分成两个独立的集合：训练集和测试集。通常，三分之一的数据分配到训练集，其余三分之二分配到测试集。使用训练集得到分类器，其准确率用测试集评估。
3. 交叉验证：先把数据随机分成不相交的n份，每份大小基本相等，训练和测试都进行n次。比如，如果把数据分成10份,先把第一份拿出来放在一边用作模型测试，把其他9份合在一起来建立模型，然后把这个用90%的数据建立起来的模型用上面放在一边的第一份数据做测试。这个过程对每一份数据都重复进行一次，得到10个不同的错误率。然后把所有数据放在起建立一个模型，模型的错误率为上面10个错误率的平均。使用这些技术评估分类法的准确率增加了总体的计算时间，但是对于分类方法的选择是有意义的。

五、

1. 距离与相似性的度量：
2. 本章我们使用ｓ（ｘ，ｙ）表示样本ｘ和样本ｙ的相似度。当ｘ和ｙ相似时，ｓ（ｘ，ｙ）的取值是很大的；当ｘ，ｙ不相似时，ｓ（ｘ，ｙ）的取值是很小的。相似性的度量具有自反性ｓ（ｘ，ｙ）＝ｓ（ｙ，ｘ）。对于大多数据类方法来说，相似性度量标准被标准化为　０≤ｓ（ｘ，ｙ）≤１.
3. 相异度的度量用ｄ（ｘ，ｙ）来表示，通常成相异度为距离。当ｘ和ｙ相似时，距离ｄ（ｘ，ｙ）的取值很小；当ｘ，ｙ不相似时，ｄ（ｘ，ｙ）取值很大。
4. 明可夫斯基距离　Ｐ１７４

当ｒ取值为１时，为绝对值距离

当ｒ取值为２时，为欧氏距离

1. 类间距离
2. 最短距离法：定义两个类中最靠近的两个元素间的距离为类间距离。
3. 最长距离法：定义两个类中最远的两个元素间的距离为类间距离。
4. 中心法：定义两类的两个中心间的距离为类间距离。
5. 类平均法：将两个类中任意两个元素间的距离的平均值定义为类间距离
6. 离差平方和：类的直径反映了类中各元素间的差异，可以定义为类中各元素至类中心的欧式距离之和。
7. K－Means算法
8. 概念：ｋ－平均算法以ｋ为参数，把ｎ个对象分为ｋ个簇，以使簇内具有较高的相似度。相似度的计算根据一个簇中对象的平均值来进行。

算法首先随机地选择ｋ个对象，每个对象初始地代表了一个簇的平均值或中心。对剩余的每个对象根据其与各个簇中心的距离，将它赋给最近的簇。然后重新计算每个簇的平均值。这个过程不断重复，直到准则函数收敛。

1. 算法： P178页，符号打不出来

Ｋ－means算法

输入：簇的数目ｋ和包含ｎ个对象的数据库

输出：ｋ个簇，使平方误差准则最小

1. 计算：大概率，计算点与点的欧式距离，记欧式距离公式
2. PAM算法
3. 算法思想（原理）：选用簇中位置最中心的对象，试图对n个对象给出k个划分；代表对象也被称为是中心点，其他对象则被称为非代表对象；最初随机选择k个对象作为中心点，该算法反复地用非代表对象来代替代表对象，试图找出更好的中心点，以改进聚类的质量；在每次迭代中，所有可能的对象对被分析，每个对中的一个对象是中心点，而另一个是非代表对象。对可能的各种组合，估算聚类结果的质量；一个对象Oi可以被使最大平方-误差值减少的对象代替；在一次迭代中产生的最佳对象集合成为下次迭代的中心点。
4. 步骤：

1.在总体n个样本点中任意选取k个点作为medoids

　 2.按照与medoids最近的原则，将剩余的n-k个点分配到当前最佳的medoids代表的类中（实现了初始的聚类）

　 3.对于第 i 个类中除对应medoids点外的所有其他点，按顺序计算当其为新的medoids时，准则函数的值，遍历所有可能，选取准则函数最小时对应的点作为新的medoids

　 4.重复2-3的过程，直到所有的medoids点不再发生变化或已达到设定的最大迭代次数

　 5.产出最终确定的k个类

1. AGNES算法实例分析、伪码、描述和过程 (DIANA算法同)
2. 思想：是凝聚的层次据类算法。AGNES算法最初将每个对象作为一个簇，然后这些簇根据某些准则被一步步合并。
3. 伪代码（描述）：

AGENS（自底向上凝聚算法）

输入：包含n个对象的数据库，终止条件簇的数目k。

输出：k个簇，达到终止条件规定簇数目

将每个对象当成一个簇

REPEAT

根据两个簇中最近的数据点找到最近的两个簇；

合并两个簇，生成新的簇的集合；

UNTIL　达到定义的簇的数目；

1. 计算：案例P186
2. DIANA算法思想：属于分裂的层次聚类。与凝聚的层次据类相反，它采用一种自顶向下的策略，它首先将所有对象置于一个簇中，然后逐渐细分为越来越小的簇，直到每个对象自成一簇，或者达到某个终结条件。
3. 伪代码（描述）：

输入：包含ｎ个对象的数据库，终止条件簇的数目ｋ。

输出：ｋ个簇，达到终结条件规定的簇数

将所有对象整个当成一个初始簇;

FOR（i＝1; i≠ｋ; i++）DO BEGIN

在所有簇中挑出具有最大直径的簇;

找出所挑出簇里与其他平均相异度最大的一个点放入splinter group,剩余的放入old party中;

REPEAT

在old party里找出到splinter group中点的最近距离不大于到old party中点的最近距离的点，并将该点加入splinter group;

UNTIL 没有新的old party的点分配给splinter group;

Splinter group和old party为被选中的簇分裂成的两个簇，与其他簇一起组成新的簇集合。

END;

1. 计算：样例P187
2. DBSCAN算法描述、实例分析
3. 描述：是一个比较有代表性的基于密度的据类算法。与划分和层次据类方法不同，它将簇定义为密度相连的点的最大集合，能够把具有足够高密度的区域划分为簇，并可在有“噪声”的空间数据库中发现任意形状的聚类。
4. 案例：P191

六、

1. 时间序列：所谓时间序列就是将某一指标在不同时间上的不同数值，按照时间的先后顺序排列而成的数列。
2. 序列挖掘：或称序列模式挖掘，是指从序列数据库中发现相对时间或者其他顺序所出现的高频率子序列。

七、

1.7.7.2 原理，计算，伪码

1）PageRank原理：

2）伪代码：P266

3）公式：

4）步骤：

1. 补充：