机器学习大题（内部保密）

2.哪些机器学习算法不需要做归一化处理?
6.请简要说说一个完整机器学习项目的流程?
10.LR 和 SVM 的区别和联系?
24.请问（决策树.Random Forest.Boosting.Adaboot）GBDT 和 XGBoost 的区别是什么?
25.说说常见的损失函数?
31.线性分类器与非线性分类器的区别以及优劣?
32.L2.L1 的区别?
36.具体 Google 是怎么利用贝叶斯方法，实现"拼写检查"的功能?
39.请详细说说 EM 算法?
42.机器学习中，为何要经常对数据做归一化?
49.随机森林如何评估特征重要性?
50.请说说 Kmeans 的优化?

51.KMeans 初始类簇中心点的选取。
52.解释对偶的概念。
53.如何进行特征选择?
54.衡量分类器的好坏?
56.数据预处理。
58.什麽造成梯度消失问题?
59.到底什么是特征工程?
60.你知道有哪些数据处理和特征工程的处理?
62.数据不平衡问题
63.特征比数据量还大时，选择什么样的分类器?
64.常见的分类算法有哪些？他们各自的优缺点是什么?
65.常见的监督学习算法有哪些?
66.说说常见的优化算法及其优缺点?
67.特征向量的归一化方法有哪些?
68.RF 与 GBDT 之间的区别与联系?
69.证明样本空间任一点到超平面的距离公式
70.请比较下 EM 算法、HMM、CRF
71.带核的 SVM 为什么能分类非线性问题?
72.请说说常用核函数及核函数的条件
73.请具体说说 Boosting 和 Bagging 的区别
74.逻辑回归相关问题
75.什么是共线性，跟过拟合有什么关联?
77.用贝叶斯机率说明 Dropout 的原理
78.对于维度极低的特征，选择线性还是非线性分类器?
79.请问怎么处理特征向量的缺失值
80.SVM、LR、决策树的对比。
81.什么是 ill-condition 病态问题?
82.简述 KNN 最近邻分类算法的过程?
83.常用的聚类划分方式有哪些？列举代表算法。
84.什么是偏差与方差?
85.解决 bias 和 Variance 问题的方法是什么?
86.采用 EM 算法求解的模型有哪些，为什么不用牛顿法或梯度下降法?
87.xgboost 怎么给特征评分?
88.什么是 OOB?随机森林中 OOB 是如何计算的，它有什么优缺点?

89.推导朴素贝叶斯分类 P(c|d)，文档 d（由若干 word 组成），求该文档属于类别 c 的概率， 并说明公式中哪些概率可以利用训练集计算得到
91.请写出你对 VC 维的理解和认识
92.kmeans 聚类中，如何确定 k 的大小
94.怎么理解“机器学习的各种模型与他们各自的损失函数一一对应？”
95.给你一个有 1000 列和 1 百万行的训练数据集。这个数据集是基于分类问题的。
经理要求你来降低该数据集的维度以减少模型计算时间。你的机器内存有限。你会怎么做？
96.在 PCA 中有必要做旋转变换吗？如果有必要，为什么？如果你没有旋转变换那些成分，会发生什么情况？
97.给你一个数据集，这个数据集有缺失值，且这些缺失值分布在离中值有 1 个标准偏差的范围内。百分之多少的数据不会受到影响？为什么？
98.给你一个癌症检测的数据集。你已经建好了分类模型，取得了 96%的精度。为什么你还是不满意你的模型性能？你可以做些什么呢？
99.解释朴素贝叶斯算法里面的先验概率、似然估计和边际似然估计？
100.你正在一个时间序列数据集上工作。经理要求你建立一个高精度的模型。你开始用决策树算法，因为你知道它在所有类型数据上的表现都不错。后来，你尝试了时间序列回归模型，并得到了比决策树模型更高的精度。这种情况会发生吗？为什么？

101.给你分配了一个新的项目，是关于帮助食品配送公司节省更多的钱。问题是，公司的送餐队伍没办法准时送餐。结果就是他们的客户很不高兴。最后为了使客户高兴，他们只好以免餐费了事。哪个机器学习算法能拯救他们？
102.你意识到你的模型受到低偏差和高方差问题的困扰。应该使用哪种算法来解决问题呢？为什么？
103.给你一个数据集。该数据集包含很多变量，你知道其中一些是高度相关的。
107.KNN 和 KMEANS 聚类有什么不同？
112.是否有可能捕获连续变量和分类变量之间的相关性？如果可以的话，怎样做？
113.Gradient boosting 算法（GBM）和随机森林都是基于树的算法，它们有什么区别？
114.运行二元分类树算法很容易，但是你知道一个树是如何做分割的吗，即树如何决定把哪些变量分到哪个根节点和后续节点上？
115.你已经建了一个有 10000 棵树的随机森林模型。在得到 0.00 的训练误差后，你非常高兴。验证错误是 34.23。到底是怎么回事？你还没有训练好你的模型吗？
116.你有一个数据集，变量个数 p 大于观察值个数 n。为什么用最小二乘法 OLS 是一个不好的选择？用什么技术最好？为什么？
117.什么是凸包？（提示：想一想 SVM）。
118.我们知道，一位有效编码会增加数据集的维度。但是，标签编码不会。为什么？
119.你会在时间序列数据集上使用什么交叉验证技术？是用 k 倍或 LOOCV？
120.给你一个缺失值多于 30%的数据集。比方说，在 50 个变量中，有 8 个变量的缺失值都多于 30%。你对此如何处理？
121.“买了这个的客户，也买了……”亚马逊的建议是哪种算法的结果？
122.你怎么理解第一类和第二类错误？
123.当你在解决一个分类问题时，出于验证的目的，你已经将训练集随机抽样地分成训练集和验证集。你对你的模型能在未看见的数据上有好的表现非常有信心，因为你的验证精度高。但是，在得到很差的精度后，你大失所望。什么地方出了错？
124.请简单阐述下决策树.回归.SVM.神经网络等算法各自的优缺点？
133.机器学习中的 L0.L1 与 L2 范数到底是什么意思？
144.线性回归要求因变量服从正态分布？

2、哪些机器学习算法不需要做归一化处理?

解析:

在实际应用中，通过梯度下降法求解的模型一般都是需要归一化的，比如线性回归、logistic回归、KNN、SVM、神经网络等模型。

但树形模型不需要归一化，因为它们不关心变量的值，而是关心变量的分布和变量之间的条件概率，如决策树、随机森林(Random Forest)。

其他如管博士所说，我归一化和标准化主要是为了使计算更方便 比如两个变量的量纲不同 可能一个的数值远大于另一个那么他们同时作为变量的时候 可能会造成数值计算的问题，比如说求矩阵的逆可能很不精确 或者梯度下降法的收敛比较困难，还有如果需要计算欧式距离的话可能 量纲也需要调整 所以我估计lr 和 knn 标准化一下应该有好处。
至于其他的算法 我也觉得如果变量量纲差距很大的话 先标准化一下会有好处。

我们会经常提到标准化、归一化，那到底什么是标准化和归一化呢?

标准化：特征均值为0，方差为1
公式:

$$\frac{x - np.mean(x)}{np.std(x)}$$

归一化：把每个特征向量（特别是奇异样本数据）的值都缩放到相同数值范围，如[0,1]或[-1,1]。

最常用的归一化形式就是将特征向量调整为L1范数（就是绝对值相加），使特征向量的数值之和为1。
而L2范数就是欧几里得之和。
data_normalized = preprocessing.normalize(data , norm="L1")

公式:

$$\frac{x - \min(x)}{\max(x) - \min(x)}$$

这个方法经常用于确保数据点没有因为特征的基本性质而产生较大差异，即确保数据处于同一数量级（同一量纲），提高不同特征数据的可比性。

6、请简要说说一个完整机器学习项目的流程

解析：

1 抽象成数学问题

明确问题是进行机器学习的第一步。机器学习的训练过程通常都是一件非常耗时的事情，胡乱尝试时间成本是非常高的。

这里的抽象成数学问题，指的我们明确我们可以获得什么样的数据，目标是一个分类还是回归或者是聚类的问题，如果都不是的话，如果划归为其中的某类问题。

2 获取数据

数据决定了机器学习结果的上限，而算法只是尽可能逼近这个上限。

数据要有代表性，否则必然会过拟合。

而且对于分类问题，数据偏斜不能过于严重，不同类别的数据数量不要有数个数量级的差距。

而且还要对数据的量级有一个评估，多少个样本，多少个特征，可以估算出其对内存的消耗程度，判断训练过程中内存是否能够放得下。如果放不下就得考虑改进算法或者使用一些降维的技巧了。如果数据量实在太大，那就要考虑分布式了。

3 特征预处理与特征选择

良好的数据要能够提取出良好的特征才能真正发挥效力。

特征预处理、数据清洗是很关键的步骤，往往能够使得算法的效果和性能得到显著提高。归一化、离散化、因子化、缺失值处理、去除共线性等，数据挖掘过程中很多时间就花在它们上面。这些工作简单可复制，收益稳定可预期，是机器学习的基础必备步骤。

筛选出显著特征、摒弃非显著特征，需要机器学习工程师反复理解业务。这对很多结果有决定性的影响。特征选择好了，非常简单的算法也能得出良好、稳定的结果。这需要运用特征有效性分析的相关技术，如相关系数、卡方检验、平均互信息、条件熵、后验概率、逻辑回归权重等方法。

4 训练模型与调优

直到这一步才用到我们上面说的算法进行训练。现在很多算法都能够封装成黑盒供人使用。但是真正考验水平的是调整这些算法的（超）参数，使得结果变得更加优良。这需要我们对算法的原理有深入的理解。理解越深入，就越能发现问题的症结，提出良好的调优方案。

5 模型诊断

如何确定模型调优的方向与思路呢？这就需要对模型进行诊断的技术。

过拟合、欠拟合 判断是模型诊断中至关重要的一步。常见的方法如交叉验证，绘制学习曲线等。过拟合的基本调优思路是增加数据量，降低模型复杂度。欠拟合的基本调优思路是提高特征数量和质量，增加模型复杂度。

误差分析 也是机器学习至关重要的步骤。通过观察误差样本，全面分析误差产生误差的原因:是参数的问题还是算法选择的问题，是特征的问题还是数据本身的问题……

诊断后的模型需要进行调优，调优后的新模型需要重新进行诊断，这是一个反复迭代不断逼近的过程，需要不断地尝试， 进而达到最优状态。

6 模型融合

10、LR和SVM的联系与区别

隐藏解析 17条讨论 10 / 148 上一题 下一题

解析：

解析—
LR和SVM都可以处理分类问题，且一般都用于处理线性二分类问题（在改进的情况下可以处理多分类问题）

区别：
1、LR是参数模型，svm是非参数模型，linear和rbf则是针对数据线性可分和不可分的区别；
2、从目标函数来看，区别在于逻辑回归采用的是logistical loss，SVM采用的是hinge loss，这两个损失函数的目的都是增加对分类影响较大的数据点的权重，减少与分类关系较小的数据点的权重。

3、SVM的处理方法是只考虑support vectors，也就是和分类最相关的少数点，去学习分类器。而逻辑回归通过非线性映射，大大减小了离分类平面较远的点的权重，相对提升了与分类最相关的数据点的权重。
4、逻辑回归相对来说模型更简单，好理解，特别是大规模线性分类时比较方便。而SVM的理解和优化相对来说复杂一些，SVM转化为对偶问题后,分类只需要计算与少数几个支持向量的距离,这个在进行复杂核函数计算时优势很明显,能够大大简化模型和计算。

5、logic 能做的 svm能做，但可能在准确率上有问题，svm能做的logic有的做不了。
本解析—来源：@朝阳在望http://blog.csdn.net/timcompp/article/details/62237986

22、LR与线性回归的区别与联系

隐藏解析 6条讨论 22 / 148 上一题 下一题

解析：

LR工业上一般指Logistic Regression(逻辑回归)而不是Linear Regression(线性回归). LR在线性回归的实数范围输出值上施加sigmoid函数将值收敛到0~1范围, 其目标函数也因此从差平方和函数变为对数损失函数, 以提供最优化所需导数（sigmoid函数是softmax函数的二元特例, 其导数均为函数值的f*(1-f)形式）。请注意, LR往往是解决二元0/1分类问题的, 只是它和线性回归耦合太紧, 不自觉也冠了个回归的名字(马甲无处不在). 若要求多元分类,就要把sigmoid换成大名鼎鼎的softmax了。
引用自：@AntZ
个人感觉逻辑回归和线性回归首先都是广义的线性回归，
其次经典线性模型的优化目标函数是最小二乘，而逻辑回归则是似然函数，
另外线性回归在整个实数域范围内进行预测，敏感度一致，而分类范围，需要在[0,1]。逻辑回归就是一种减小预测范围，将预测值限定为[0,1]间的一种回归模型，因而对于这类问题来说，逻辑回归的鲁棒性比线性回归的要好。
引用自：@nishizhen
逻辑回归的模型本质上是一个线性回归模型，逻辑回归都是以线性回归为理论支持的。但线性回归模型无法做到sigmoid的非线性形式，sigmoid可以轻松处理0/1分类问题。

24、请问（决策树、Random Forest、Boosting、Adaboot）GBDT和XGBoost的区别是什么?

解析：

集成学习的集成对象是学习器. Bagging和Boosting属于集成学习的两类方法. Bagging方法有放回地采样同数量样本训练每个学习器, 然后再一起集成(简单投票); Boosting方法使用全部样本(可调权重)依次训练每个学习器, 迭代集成(平滑加权).

决策树属于最常用的学习器, 其学习过程是从根建立树, 也就是如何决策叶子节点分裂. ID3/C4.5决策树用信息熵计算最优分裂, CART决策树用基尼指数计算最优分裂, xgboost决策树使用二阶泰勒展开系数计算最优分裂.

下面所提到的学习器都是决策树:
Bagging方法:
学习器间不存在强依赖关系, 学习器可并行训练生成, 集成方式一般为投票;
Random Forest属于Bagging的代表, 放回抽样, 每个学习器随机选择部分特征去优化;
Boosting方法:
学习器之间存在强依赖关系、必须串行生成, 集成方式为加权和;
Adaboost属于Boosting, 采用指数损失函数替代原本分类任务的0/1损失函数;
GBDT属于Boosting的优秀代表, 对函数残差近似值进行梯度下降, 用CART回归树做学习器, 集成为回归模型;
xgboost属于Boosting的集大成者, 对函数残差近似值进行梯度下降, 迭代时利用了二阶梯度信息, 集成模型可分类也可回归. 由于它可在特征粒度上并行计算, 结构风险和工程实现都做了很多优化, 泛化, 性能和扩展性都比GBDT要好。

关于决策树，这里有篇《决策树算法》（链接：http://blog.csdn.net/v_july_v/article/details/7577684）。而随机森林Random Forest是一个包含多个决策树的分类器。至于AdaBoost，则是英文"Adaptive Boosting"（自适应增强）的缩写，关于AdaBoost可以看下这篇文章《Adaboost 算法的原理与推导》。GBDT（Gradient Boosting Decision Tree），即梯度上升决策树算法，相当于融合决策树和梯度上升boosting算法。
引用自：@AntZ

xgboost类似于gbdt的优化版，不论是精度还是效率上都有了提升。与gbdt相比，具体的优点有：
1.损失函数是用泰勒展式二项逼近，而不是像gbdt里的就是一阶导数
2.对树的结构进行了正则化约束，防止模型过度复杂，降低了过拟合的可能性
3.节点分裂的方式不同，gbdt是用的gini系数，xgboost是经过优化推导后的

接下来，我们来重点看下 下面几种损失函数。

一、log对数损失函数（逻辑回归）
有些人可能觉得逻辑回归的损失函数就是平方损失，其实并不是。平方损失函数可以通过线性回归在假设样本是高斯分布的条件下推导得到，而逻辑回归得到的并不是平方损失。在逻辑回归的推导中，它假设样本服从伯努利分布（0-1分布），然后求得满足该分布的似然函数，接着取对数求极值等等。

二、平方损失函数（最小二乘法, Ordinary Least Squares ）
最小二乘法是线性回归的一种，OLS将问题转化成了一个凸优化问题。在线性回归中，它假设样本和噪声都服从高斯分布（为什么假设成高斯分布呢？其实这里隐藏了一个小知识点，就是中心极限定理，可以参考：https://blog.csdn.net/v_july_v/article/details/8308762），最后通过极大似然估计（MLE）可以推导出最小二乘式子。最小二乘的基本原则是：最优拟合直线应该是使各点到回归直线的距离和最小的直线，即平方和最小。

三、指数损失函数（Adaboost）
学过Adaboost算法的人都知道，它是前向分步加法算法的特例，是一个加和模型，损失函数就是指数函数。在Adaboost中，经过m此迭代之后，可以得到fm(x)fm(x):

$$f_m(x) = f_{m-1}(x) + \alpha_m G_m(x)$$

五、其它损失函数
除了以上这几种损失函数，常用的还有：
0-1损失函数

$$L(Y, f(X)) = \begin{cases} 1, & Y \neq f(X) \\ 0, & y = f(X) \end{cases}$$

绝对值损失函数

$$L(Y, f(X)) = |Y - f(X)|$$

31、线性分类器与非线性分类器的区别以及优劣

解析：

线性和非线性是针对，模型参数和输入特征来讲的；比如输入x，模型y=ax+ax^2那么就是非线性模型，如果输入是x和X^2则模型是线性的。
线性分类器可解释性好，计算复杂度较低，不足之处是模型的拟合效果相对弱些。
非线性分类器效果拟合能力较强，不足之处是数据量不足容易过拟合、计算复杂度高、可解释性不好。
常见的线性分类器有：LR,贝叶斯分类，单层感知机、线性回归
常见的非线性分类器：决策树、RF、GBDT、多层感知机
SVM两种都有（看线性核还是高斯核）

32、L1和L2的区别

解析：

L1范数（L1 norm）是指向量中各个元素绝对值之和，也有个美称叫“稀疏规则算子”（Lasso regularization）。
比如 向量A=[1，-1，3]， 那么A的L1范数为 |1|+|-1|+|3|.

简单总结一下就是：
L1范数: 为x向量各个元素绝对值之和。
L2范数: 为x向量各个元素平方和的1/2次方，L2范数又称Euclidean范数或者Frobenius范数
Lp范数: 为x向量各个元素绝对值p次方和的1/p次方.

在支持向量机学习过程中，L1范数实际是一种对于成本函数求解最优的过程，因此，L1范数正则化通过向成本函数中添加L1范数，使得学习得到的结果满足稀疏化，从而方便人类提取特征，即L1范数可以使权值稀疏，方便特征提取。
L2范数可以防止过拟合，提升模型的泛化能力。

L1和L2的差别，为什么一个让绝对值最小，一个让平方最小，会有那么大的差别呢？看导数一个是1一个是w便知，在靠进零附近，L1以匀速下降到零，而L2则完全停下来了. 这说明L1是将不重要的特征(或者说，重要性不在一个数量级上)尽快剔除，L2则是把特征贡献尽量压缩最小但不至于为零. 两者一起作用，就是把重要性在一个数量级(重要性最高的)的那些特征一起平等共事(简言之，不养闲人也不要超人)。

找到约 733,000 条结果 （用时 0.53 秒）

显示的是以下查询字词的结果： July
仍然搜索： Julw

结构之法算法之道- 博客频道- CSDN.NET
blog.csdn.net/v_JULY_v
从头到尾彻底理解KMP 作者：July 时间：最初写于2011年12月，2014年7月21日晚10点全部删除重写成此文。 1. 引言本KMP原文最初写于2年多前的2011年12月，因 ...
程序员面试、算法研究、编程艺术 - 程序员如何快速准备面试中的算法 - 目录视图 - 尾页

这叫做拼写检查。

根据谷歌一员工写的文章显示，Google的拼写检查基于贝叶斯方法。请说说的你的理解，具体Google是怎么利用贝叶斯方法，实现"拼写检查"的功能。

隐藏解析 3条讨论 36 / 148 上一题 下一题

解析：

用户输入一个单词时，可能拼写正确，也可能拼写错误。如果把拼写正确的情况记做c（代表correct），拼写错误的情况记做w（代表wrong），那么"拼写检查"要做的事情就是：在发生w的情况下，试图推断出c。换言之：已知w，然后在若干个备选方案中，找出可能性最大的那个c，也就是求$P(c|w)$的最大值。

而根据贝叶斯定理，有：$P(c|w) = P(w|c) * P(c)/P(w)$

由于对于所有备选的c来说，对应的都是同一个w，所以它们的P(w)是相同的，因此我们只要最大化

$P(w|c) * P(c)$

即可。其中：

P(c)表示某个正确的词的出现"概率"，它可以用"频率"代替。如果我们有一个足够大的文本库，那么这个文本库中每个单词的出现频率，就相当于它的发生概率。某个词的出现频率越高，P(c)就越大。比如在你输入一个错误的词“Julw”时，系统更倾向于去猜测你可能想输入的词是“July”，而不是“Jult”，因为“July”更常见。

P(w|c)表示在试图拼写c的情况下，出现拼写错误w的概率。为了简化问题，假定两个单词在字形上越接近，就有越可能拼错，P(w|c)就越大。举例来说，相差一个字母的拼法，就比相差两个字母的拼法，发生概率更高。你想拼写单词July，那么错误拼成Julw（相差一个字母）的可能性，就比拼成Jullw高（相差两个字母）。值得一提的是，一般把这种问题称为“编辑距离”，参见博客中的这篇文章。

所以，我们比较所有拼写相近的词在文本库中的出现频率，再从中挑出出现频率最高的一个，即是用户最想输入的那个词。具体的计算过程及此方法的缺陷请参见这里。

到底什么是EM算法呢？Wikipedia给的解释是：

最大期望算法（Expectation-maximization algorithm，又译为期望最大化算法），是在概率模型中寻找参数最大似然估计或者最大后验估计的算法，其中概率模型依赖于无法观测的隐性变量。

最大期望算法经过两个步骤交替进行计算，

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；

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。

一般做机器学习应用的时候大部分时间是花费在特征处理上，其中很关键的一步就是对特征数据进行归一化。

为什么要归一化呢？很多同学并未搞清楚，维基百科给出的解释：1）归一化后加快了梯度下降求最优解的速度；2）归一化有可能提高精度。

下面再简单扩展解释下这两点。

1 归一化为什么能提高梯度下降法求解最优解的速度？

2 归一化有可能提高精度

一些分类器需要计算样本之间的距离（如欧氏距离），例如KNN。如果一个特征值域范围非常大，那么距离计算就主要取决于这个特征，从而与实际情况相悖（比如这时实际情况是值域范围小的特征更重要）。

3 归一化的类型

1）线性归一化

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

这种归一化方法比较适用在数值比较集中的情况。这种方法有个缺陷，如果max和min不稳定，很容易使得归一化结果不稳定，使得后续使用效果也不稳定。实际使用中可以用经验常量值来替代max和min。

49、随机森林如何评估特征重要性？

隐藏解析 4条讨论 49 / 148 上一题 下一题

解析：

衡量变量重要性的方法有两种，Decrease GINI 和 Decrease Accuracy：

1) Decrease GINI：对于回归问题，直接使用argmax(VarVarLeftVarRight)作为评判标准，即当前节点训练集的方差Var减去左节点的方差VarLeft和右节点的方差VarRight。

2) Decrease Accuracy：对于一棵树Tb(x)，我们用OOB样本可以得到测试误差1；然后随机改变OOB样本的第j列：保持其他列不变，对第j列进行随机的上下置换，得到误差2。至此，我们可以用误差1-误差2来刻画变量j的重要性。基本思想就是，如果一个变量j足够重要，那么改变它会极大的增加测试误差；反之，如果改变它测试误差没有增大，则说明该变量不是那么的重要。

50、请说说Kmeans的优化？

隐藏解析 4条讨论 50 / 148 上一题 下一题

解析：

解析一

k-means：在大数据的条件下，会耗费大量的时间和内存。

优化k-means的建议：

1、减少聚类的数目K。因为，每个样本都要跟类中心计算距离。

2、减少样本的特征维度。比如说，通过PCA等进行降维。

3、考察其他的聚类算法，通过选取toy数据，去测试不同聚类算法的性能。

4、hadoop集群，K-means算法是很容易进行并行计算的。

101.给你分配了一个新的项目，是关于帮助食品配送公司节省更多的钱。问题是，公司的送餐队伍没办法准时送餐。结果就是他们的客户很不高兴。最后为了使客户高兴，他们只好以免餐费了事。哪个机器学习算法能拯救他们？

解析：你的大脑里可能已经开始闪现各种机器学习的算法。但是等等！这样的提问方式只是来测试你的机器学习基础。这不是一个机器学习的问题，而是一个路径优化问题。

机器学习问题由三样东西组成：1.模式已经存在。2.不能用数学方法解决（指数方程都不行）。3.有相关的数据。

102.你意识到你的模型受到低偏差和高方差问题的困扰。应该使用哪种算法来解决问题呢？为什么？

解析：低偏差意味着模型的预测值接近实际值。换句话说，该模型有足够的灵活性，以模仿训练数据的分布。貌似很好，但是别忘了，一个灵活的模型没有泛化能力。这意味着，当这个模型用在对一个未曾见过的数据集进行测试的时候，它会令人很失望。在这种情况下，我们可以使用 bagging 算法（如随机森林），以解决高方差问题。bagging 算法把数据集分成重复随机取样形成的子集。然后，这些样本利用单个学习算法生成一组模型。接着，利用投票（分类）或平均（回归）把模型预测结合在一起。另外，为了应对大方差，我们可以：1.使用正则化技术，惩罚更高的模型系数，从而降低了模型的复杂性。2.使用可变重要性图表中的前 n 个特征。可以用于当一个算法在数据集中的

所有变量里很难寻找到有意义信号的时候。

103.给你一个数据集。该数据集包含很多变量，你知道其中一些是高度相关的。经理要求你用 PCA。你会先去掉相关的变量吗？为什么？
解析：你可能会说不，但是这有可能是不对的。丢弃相关变量会对 PCA 有实质性的影响，因为有相关变量的存在，由特定成分解释的方差被放大。例如：在一个数据集有 3 个变量，其中有 2 个是相关的。如果在该数据集上用 PCA，第一主成分的方差会是与其不相关变量的差异的两倍。此外，加入相关的变量使 PCA 错误地提高那些变量的重要性，这是有误导性的。

105.KNN 和 KMEANS 聚类（kmeans clustering）有什么不同？
解析：这两种算法之间的根本区别是，KMEANS 本质上是无监督学习而 KNN 是监督学习。KMEANS 是聚类算法。KNN 是分类（或回归）算法。 KMEAN 算法把一个数据集分割成簇，使得形成的簇是同构的，每个簇里的点相互靠近。该算法试图维持这些簇之间有足够的可分离性。由于无监督的性质，这些簇没有任何标签。NN 算法尝试基于其 k（可以是任何数目）个周围邻居来对未标记的观察进行分类。它也被称为懒惰学习法，因为它涉及最小的模型训练。因此，它不用训练数据对未看见的数据集进行泛化。

106.真阳性率和召回有什么关系？写出方程式。
解析：真阳性率=召回。是的，它们有相同的公式（TP / TP + FN）。

107.你建了一个多元回归模型。你的模型 R2 为并不如你设想的好。为了改进，你去掉截距项，模型 R 的平方从 0.3 变为 0.8。这是否可能？怎样才能达到这个结果？
解析：是的，这有可能。我们需要了解截距项在回归模型里的意义。截距项显示模型预测没有任何自变量，比如平均预测。公式 $R^2 = 1 - \sum(y - y')^2 / \sum(y - ymean)^2$ 中的 y' 是预测值。当有截距项时，R^2 值评估的是你的模型基于均值模型的表现。在没有截距项（ymean）时，当分母很大时，该模型就没有这样的估值效果了，$\sum(y-y')^2/\sum(y-ymean)^2$ 式的值会变得比实际的小，而 R2 会比实际值大。

112.是否有可能捕获连续变量和分类变量之间的相关性？如果可以的话，怎样做？
解析：我们可以用 ANCOVA（协方差分析）技术来捕获连续型变量和分类变量之间的相关性。

113.Gradient boosting 算法（GBM）和随机森林都是基于树的算法，它们有什么区别？
解析：最根本的区别是，随机森林算法使用 bagging 技术做出预测。 GBM 采用 boosting 技术做预测。在 bagging 技术中，数据集用随机采样的方法被划分成使 n 个样本。然后，使用单一的学习算法，在所有样本上建模。接着利用投票或者求平均来组合所得到的预测。Bagging 是平行进行的。而 boosting 是在第一轮的预测之后，算法将分类出错的预测加高权重，使得它们可以在后续一轮中得到校正。这种给予分类出错的预测高权重的顺序过程持续进行，一直到达到停止标准为止。随机森林通过减少方差（主要方式）提高模型的精度。生成树之间是不相关的，以把方差的减少最大化。在另一方面，GBM 提高了精度，同时减少了模型的偏差和方差。

114.运行二元分类树算法很容易，但是你知道一个树是如何做分割的吗，即树如何决定把哪些变量分到哪个根节点和后续节点上？
解析：分类树利用基尼系数与节点熵来做决定。简而言之，树算法找到最好的可能特征，它可以将数据集分成最纯的可能子节点。树算法找到可以把数据集分成最纯净的可能的子节点的特征量。基尼系数是，如果总体是完全纯的，那么我们从总体中随机选择 2 个样本，而这 2 个样本肯定是同一类的而且它们是同类的概率也是 1。我们可以用以下方法计算基尼系数：1.利用成功和失败的概率的平方和(p^2+q^2)计算子节点的基尼系数 2.利用该分割的节点的加权基尼分数计算基尼系数以分割
熵是衡量信息不纯的一个标准（二分类）:这里的 p 和 q 是分别在该节点成功和失败的概率。当一个节点是均匀时熵为零。当 2 个类同时以 50%对 50%的概率出现在同一个节点上的时候，它是最大值。熵越低越好。

115.你已经建了一个有 10000 棵树的随机森林模型。在得到 0.00 的训练误差后，你非常高兴。但是，验证错误是 34.23。到底是怎么回事？你还没有训练好你的模型吗？

解析：该模型过度拟合。训练误差为 0.00 意味着分类器已在一定程度上模拟了训练数据，这样的分类器是不能用在未看见的数据上的。因此，当该分类器用于未看见的样本上时，由于找不到已有的模式，就会返回的预测有很高的错误率。在随机森林算法中，用了多于需求个数的树时，这种情况会发生。因此，为了避免这些情况，我们要用交叉验证来调整树的数量。

116.你有一个数据集，变量个数 p 大于观察值个数 n。为什么用最小二乘法 OLS 是一个不好的选择？用什么技术最好？为什么？
解析：在这样的高维数据集中，我们不能用传统的回归技术，因为它们的假设往往不成立。当 p>nN，我们不能计算唯一的最小二乘法系数估计，方差变成无穷大，因此 OLS 无法在此使用的。为了应对这种情况，我们可以使用惩罚回归方法，如 lasso.LARS.ridge，这些可以缩小系数以减少方差。准确地说，当最小二乘估计具有较高方差的时候，ridge 回归最有效。其他方法还包括子集回归.前向逐步回归。

117.什么是凸包？（提示：想一想 SVM）
解析：当数据是线性可分的，凸包就表示两个组数据点的外边界。一旦凸包建立，我们得到的最大间隔超平面（MMH）作为两个凸包之间的垂直平分线。 MMH 是能够最大限度地分开两个组的线。

118.我们知道，一位有效编码会增加数据集的维度。但是，标签编码不会。为什么？
解析：对于这个问题不要太纠结。这只是在问这两者之间的区别。用一位有效编码编码，数据集的维度（也即特征）增加是因为它为分类变量中存在的的每一级都创建了一个变量。例如：假设我们有一个变量“颜色”。这变量有 3 个层级，即红色.蓝色和绿色。对“颜色”变量进行一位有效编码会生成含 0 和 1 值的 Color.Red，Color.Blue 和 Color.Green 三个新变量。在标签编码中，分类变量的层级编码为 0 和 1，因此不生成新变量。标签编码主要是用于二进制变量。

119.你会在时间序列数据集上使用什么交叉验证技术？是用 k 倍或 LOOCV？
解析：都不是。对于时间序列问题，k 倍可能会很麻烦，因为第 4 年或第 5 年的一些模式有可能跟第 3 年的不同，而对数据集的重复采样会将分离这些趋势，我们可能最终是对过去几年的验证，这就不对了。相反，我们可以采用如下所示的 5 倍正向链接策略：
fold 1 : training [1], test [2]
fold 2 : training [1 2], test [3]
fold 3 : training [1 2 3], test [4]
fold 4 : training [1 2 3 4], test [5]
fold 5 : training [1 2 3 4 5], test [6]
1，2，3，4，5，6 代表的是年份。

120.给你一个缺失值多于 30%的数据集？比方说，在 50 个变量中，有 8 个变量的缺失值都多于 30%。你对此如何处理？
解析：我们可以用下面的方法来处理：1.把缺失值分成单独的一类，这些缺失值说不定会包含一些趋势信息。2.我们可以毫无顾忌地删除它们。3.或者，我们可以用目标变量来检查它们的分布，如果发现任何模式，我们将保留那些缺失值并给它们一个新的分类，同时删除其他缺失值。

121.“买了这个的客户，也买了……”亚马逊的建议是哪种算法的结果？
解析：这种推荐引擎的基本想法来自于协同过滤。协同过滤算法考虑用于推荐项目的“用户行为”。它们利用的是其他用户的购买行为和针对商品的交易历史记录.评分.选择和购买信息。针对商品的其他用户的行为和偏好用来推荐项目（商品）给新用户。在这种情况下，项目（商品）的特征是未知的。

122.你怎么理解第一类和第二类错误？
解析：第一类错误是当原假设为真时，我们却拒绝了它，也被称为“假阳性”。第二类错误是当原假设为是假时，我们接受了它，也被称为“假阴性”。在混淆矩阵里，我们可以说，当我们把一个值归为阳性（1）但其实它是阴性（0）时，发生第一类错误。而当我们把一个值归为阴性（0）但其实它是阳性（1）时，发生了第二类错误。

123.当你在解决一个分类问题时，出于验证的目的，你已经将训练集随机抽样地分成训练集和验证集。你对你的模型能在未看见的数据上有好的表现非常有信心，因为你的验证精度高。但是，在得到很差的精度后，你大失所望。什么地方出了错?
解析：在做分类问题时，我们应该使用分层抽样而不是随机抽样。随机抽样不考虑目标类别的比例。相反，分层抽样有助于保持目标变量在所得分布样本中的分布。

124.请简单阐述下决策树.回归.SVM.神经网络等算法各自的优缺点?
解析：一.正则化算法（Regularization Algorithms）
它是另一种方法（通常是回归方法）的拓展，这种方法会基于模型复杂性对其进行惩罚，它喜欢相对简单能够更好的泛化的模型。例子：岭回归（Ridge Regression）最小绝对收缩与选择算子（LASSO）GLASSO 弹性网络（Elastic Net）最小角回归（Least-Angle Regression）优点：其惩罚会减少过拟合总会有解决方法。缺点：惩罚会造成欠拟合很难校准
二.集成算法（Ensemble algorithms）
集成方法是由多个较弱的模型集成模型组，其中的模型可以单独进行训练，并且它们的预测能以某种方式结合起来去做出一个总体预测。该算法主要的问题是要找出哪些较弱的模型可以结合起来，以及结合的方法。这是一个非常强大的技术集，因此广受欢迎。Boosting、Bootstrapped Aggregation（Bagging）、AdaBoost、层叠泛化（Stacked Generalization）(blending)、梯度推进机（Gradient Boosting Machines，GBM)、梯度提升回归树（Gradient Boosted Regression Trees，GBRT)、随机森林（Random Forest)。优点：当先最先进的预测几乎都使用了算法集成。它比使用单个模型预测出来的结果要精确的多。缺点：需要大量的维护工作
三.决策树算法（Decision Tree Algorithm）
决策树学习使用一个决策树作为一个预测模型，它将对一个 item（表征在分支上）观察所得映射成关于该 item 的目标值的结论（表征在叶子中)。树模型中的目标是可变的，可以采一组有限值，被称为分类树；在这些树结构中，叶子表示类标签，分支表示表征这些类标签的连接的特征。例子：分类和回归树 (Classification and Regression Tree，CART)、Iterative Dichotomiser 3（ID3)、C4.5 和 C5.0（一种强大方法的两个不同版本)。优点：容易解释.非参数型缺点：趋向过拟合.可能或陷于局部最小值中.没有在线学习
四.回归算法（Regression）
回归是用于估计两种变量之间关系的统计过程。当用于分析因变量和一个 多个自变量之间的关系时，该算法能提供很多建模和分析多个变量的技巧。具体一点说，回归分析可以帮助我们理解当任意一个自变量变化，另一个自变量不变时，因变量变化的典型值。最常见的是，回归分析能在给定自变量的条件下估计出因变量的条件期望。
回归算法是统计学中的主要算法，它已被纳入统计机器学习。
例子：普通最小二乘回归（Ordinary Least Squares Regression，OLSR）线性回归（Linear Regression）
逻辑回归（Logistic Regression）逐步回归（Stepwise Regression）多元自适应回归样条（Multivariate Adaptive Regression Splines，MARS）本地散点平滑估计（Locally Estimated Scatterplot Smoothing，LOESS）优点：直接.快速，知名度高。缺点：要求严格的假设，需要处理异常值。
五.人工神经网络
人工神经网络是受生物神经网络启发而构建的算法模型。
它是一种模式匹配，常被用于回归和分类问题，但拥有庞大的子域，由数百种算法和各类问题的变体组成。
例子：感知器，反向传播、Hopfield 网络、径向基函数网络（Radial Basis Function Network，RBFN)。优点：在语音.语义.视觉.各类游戏（如围棋）的任务中表现极好，算法可以快速调整，适应新的问题。缺点：需要大量数据进行训练，训练要求很高的硬件配置，模型处于「黑箱状态」，难以理解内部机制元参数（Metaparameter）与网络拓扑选择困难。
六.深度学习（Deep Learning）
深度学习是人工神经网络的最新分支，它受益于当代硬件的快速发展。众多研究者目前的方向主要集中于构建更大.更复杂的神经网络，目前有许多方法正在聚焦半监督学习问题，其中用于训练的大数据集只包含很少的标记。例子：深玻耳兹曼机（Deep Boltzmann Machine，DBM)、Deep Belief Networks（DBN)、卷积神经网络（CNN)、Stacked Auto-Encoders。优点/缺点：见神经网络
七.支持向量机（Support Vector Machines）
给定一组训练事例，其中每个事例都属于两个类别中的一个，支持向量机（SVM）训练算法可以在被输入新的事例后将其分类到两个类别中的一个，使自身成为非概率二进制线性分类器。SVM 模型将训练事例表示为空间中的点，

它们被映射到一幅图中，由一条明确的.尽可能宽的间隔分开以区分两个类别。随后，新的示例会被映射到同一空间中，并基于它们落在间隔的哪一侧来预测它属于的类别。优点：在非线性可分问题上表现优秀。缺点：非常难以训练很难解释。

七.降维算法（Dimensionality Reduction Algorithms）

和集簇方法类似，降维追求并利用数据的内在结构，目的在于使用较少的信息总结或描述数据。这一算法可用于可视化高维数据或简化接下来可用于监督学习中的数据。许多这样的方法可针对分类和回归的使用进行调整。

例子：

主成分分析（Principal Component Analysis (PCA)）

主成分回归（Principal Component Regression (PCR)）

偏最小二乘回归（Partial Least Squares Regression (PLSR)）

Sammon 映射（Sammon Mapping）

多维尺度变换（Multidimensional Scaling (MDS)）

投影寻踪（Projection Pursuit）

线性判别分析（Linear Discriminant Analysis (LDA)）

混合判别分析（Mixture Discriminant Analysis (MDA)）

二次判别分析（Quadratic Discriminant Analysis (QDA)）

灵活判别分析（Flexible Discriminant Analysis (FDA)）

优点：可处理大规模数据集、无需在数据上进行假设。缺点：难以搞定非线性数据、难以理解结果的意义。

八.聚类算法（Clustering Algorithms）

聚类算法是指对一组目标进行分类，属于同一组（亦即一个类，cluster）的目标被划分在一组中，与其他组目标相比，同一组目标更加彼此相似（在某种意义上）。例子：K-均值（k-Means）、k-Medians 算法、Expectation Maximi 封层 ation (EM)

九.最大期望算法（EM） 分层集群（Hierarchical Clstering）优点：让数据变得有意义。缺点：结果难以解读，针对不寻常的数据组，结果可能无用。

十.基于实例的算法（Instance-based Algorithms） 基于实例的算法（有时也称为基于记忆的学习）是这样学 习算法，不是明确归纳，而是将新的问题例子与训练过程中见过的例子进行对比，这些见过的例子就在存储器中。之所以叫基于实例的算法是因为它直接从训练实例中建构出假设。这意味这，假设的复杂度能随着数据的增长而变化：最糟的情况是，假设是一个训练项目列表，分类一个单独新实例计算复杂度为 O（n）例子：K 最近邻（k-Nearest Neighbor (kNN)）、学习向量量化（Learning Vector Quantization (LVQ)）、自组织映射（Self-Organizing Map (SOM)）局部加权学习（Locally Weighted Learning (LWL)）优点：算法简单.结果易于解读。缺点：内存使用非常高、计算成本高、不可能用于高维特征空间。

十一.贝叶斯算法（Bayesian Algorithms） 贝叶斯方法是指明确应用了贝叶斯定理来解决如分类和回归等问题的方法。例子：朴素贝叶斯（Naive Bayes）、贝叶斯网络（Bayesian Network (BN)）优点：快速.易于训练.给出了它们所需的资源能带来良好的表现。缺点：如果输入变量是相关的，则会出现问题。

十二.关联规则学习算法（Association Rule Learning Algorithms） 关联规则学习方法能够提取出对数据中的变量之间的关系的最佳解释。比如说一家超市的销售数据中存在规则 {洋葱，土豆}=> {汉堡}，那说明当一位客户同时购买了洋葱和土豆的时候，他很有可能还会购买汉堡肉。例子：Apriori 算法（Apriori algorithm）、Eclat 算法（Eclat algorithm）、FP-growth

125.在应用机器学习算法之前纠正和清理数据的步骤是什么？

解析：1.将数据导入 2.看数据：重点看元数据，即对字段解释.数据来源等信息；导入数据后，提取部分数据进行查看 3.缺失值清洗- 根据需要对缺失值进行处理，可以删除数据或填充数据- 重新取数：如果某些非常重要的字段缺失，需要和负责采集数据的人沟通，是否可以再获得 4.数据格式清洗：统一数据的时间.日期.全半角等显示格式 5.逻辑错误的数据- 重复的数据- 不合理的值 6.不一致错误的处理：指对矛盾内容的修正，最常见的如身份证号和出生年月日不对应。不同业务中数据清洗的任务略有不同，比如数据有不同来源的话，数据格式清洗和不一致错误的处理就尤为突出。数据预处理是数据类岗位工作内容中重要的部分。

133.机器学习中的 L0.L1 与 L2 范数到底是什么意思？

解析：一.L0 范数与 L1 范数：L0 范数是指向量中非 0 的元素的个数。如果我们用 L0 范数来规则化一个参数矩阵 W

的话，就是希望 W 的大部分元素都是 0。换句话说，让参数 W 是稀疏的。
L1 范数是指向量中各个元素绝对值之和，也有个美称叫“稀疏规则算子”（Lasso regularization）。为什么 L1 范数会使权值稀疏？任何的规则化算子，如果他在 Wi=0 的地方不可微，并且可以分解为一个“求和”的形式，那么这个规则化算子就可以实现稀疏。这说是这么说，W 的 L1 范数是绝对值，|w|在 w=0 处是不可微，但这还是不够直观。这里因为我们需要和 L2 范数进行对比分析。既然 L0 可以实现稀疏，为什么不用 L0，而要用 L1 呢？
一句话总结：L1 范数和 L0 范数可以实现稀疏，L1 因具有比 L0 更好的优化求解特性而被广泛应用。

144.线性回归要求因变量服从正态分布?
解析：对于线性回归模型,当因变量服从正态分布,误差项满足高斯–马尔科夫条件（零均值.等方差.不相关）时,回归参数的最小二乘估计是一致最小方差无偏估计
解释一:
我们假设线性回归的噪声服从均值为 0 的正态分布。当噪声符合正态分布 N(0,delta^2)时，因变量则符合正态分布 N(ax(i)+b,delta^2)，其中预测函数 y=ax(i)+b。这个结论可以由正态分布的概率密度函数得到。也就是说当噪声符合正态分布时，其因变量必然也符合正态分布。 在用线性回归模型拟合数据之前，首先要求数据应符合或近似符合正态分布，否则得到的拟合函数不正确。若本身样本不符合正态分布或不近似服从正态分布，则要采用其他的拟合方法，比如对于服从二项式分布的样本数据，可以采用 logistics 线性回归。

53、如何进行特征选择?

隐藏解析 2条讨论 53 / 148 上一题 下一题

解析:

特征选择是一个重要的数据预处理过程，主要有两个原因：一是减少特征数量、降维，使模型泛化能力更强，减少过拟合;二是增强对特征和特征值之间的理解

常见的特征选择方式：

1. 去除方差较小的特征

2. 正则化。1正则化能够生成稀疏的模型。L2正则化的表现更加稳定，由于有用的特征往往对应系数非零。

3. 随机森林，对于分类问题，通常采用基尼不纯度或者信息增益，对于回归问题，通常采用的是方差或者最小二乘拟合。一般不需要feature engineering、调参等繁琐的步骤。它的两个主要问题，1是重要的特征有可能得分很低（关联特征问题），2是这种方法对特征变量类别多的特征越有利（偏向问题）。

4. 稳定性选择。是一种基于二次抽样和选择算法相结合较新的方法，选择算法可以是回归、SVM或其他类似的方法。它的主要思想是在不同的数据子集和特征子集上运行特征选择算法，不断的重复，最终汇总特征选择结果，比如可以统计某个特征被认为是重要特征的频率（被选为重要特征的次数除以它所在的子集被测试的次数）。理想情况下，重要特征的得分会接近100%。稍微弱一点的特征得分会是非0的数，而最无用的特征得分将会接近于0。

54、衡量分类器的好坏?

隐藏解析 1条讨论 54 / 148 上一题 下一题

解析:

这里首先要知道TP、FN（真的判成假的）、FP（假的判成真）、TN四种（可以画一个表格）。

几种常用的指标：

精度precision = TP/(TP+FP) = TP/~P （~p为预测为真的数量）

召回率 recall = TP/(TP+FN) = TP/ P

F1值： 2/F1 = 1/recall + 1/precision

ROC曲线：ROC空间是一个以伪阳性率（FPR，false positive rate）为X轴，真阳性率（TPR, true positive rate）为Y轴的二维坐标系所代表的平面。其中真阳率TPR = TP / P = recall， 伪阳率FPR = FP / N

56、数据预处理。

隐藏解析 1条讨论 56 / 148

解析:

1. 缺失值，填充缺失值fillna：

i. 离散：None,

ii. 连续：均值。

iii. 缺失值太多，则直接去除该列

2. 连续值：离散化。有的模型（如决策树）需要离散值

3. 对定量特征二值化。核心在于设定一个阈值，大于阈值的赋值为1，小于等于阈值的赋值为0。如图像操作

4. 皮尔逊相关系数，去除高度相关的列

58、什麽造成梯度消失问题?

隐藏解析 3条讨论 58 / 148 上一题 下一题

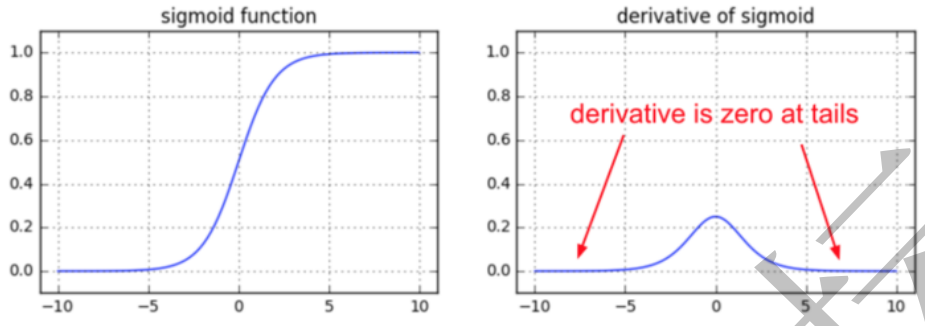
解析:

Yes you should understand backdrop - Andrej Karpathy

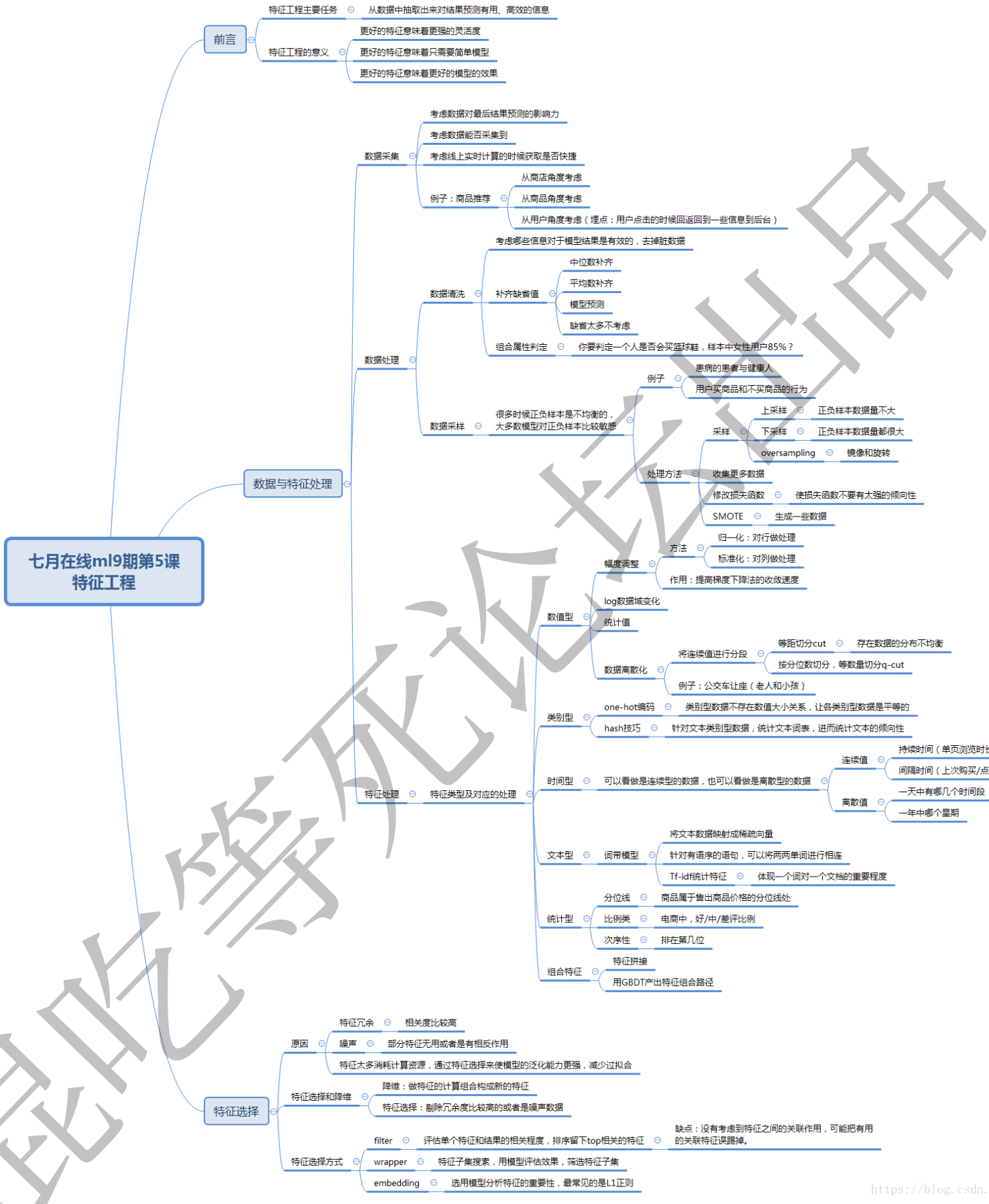
How does the ReLu solve the vanishing gradient problem?

神经网络的训练中，通过改变神经元的权重，使网络的输出值尽可能逼近标签以降低误差值，训练普遍使用BP算法，核心思想是，计算出输出与标签间的损失函数值，然后计算其相对于每个神经元的梯度，进行权值的迭代。

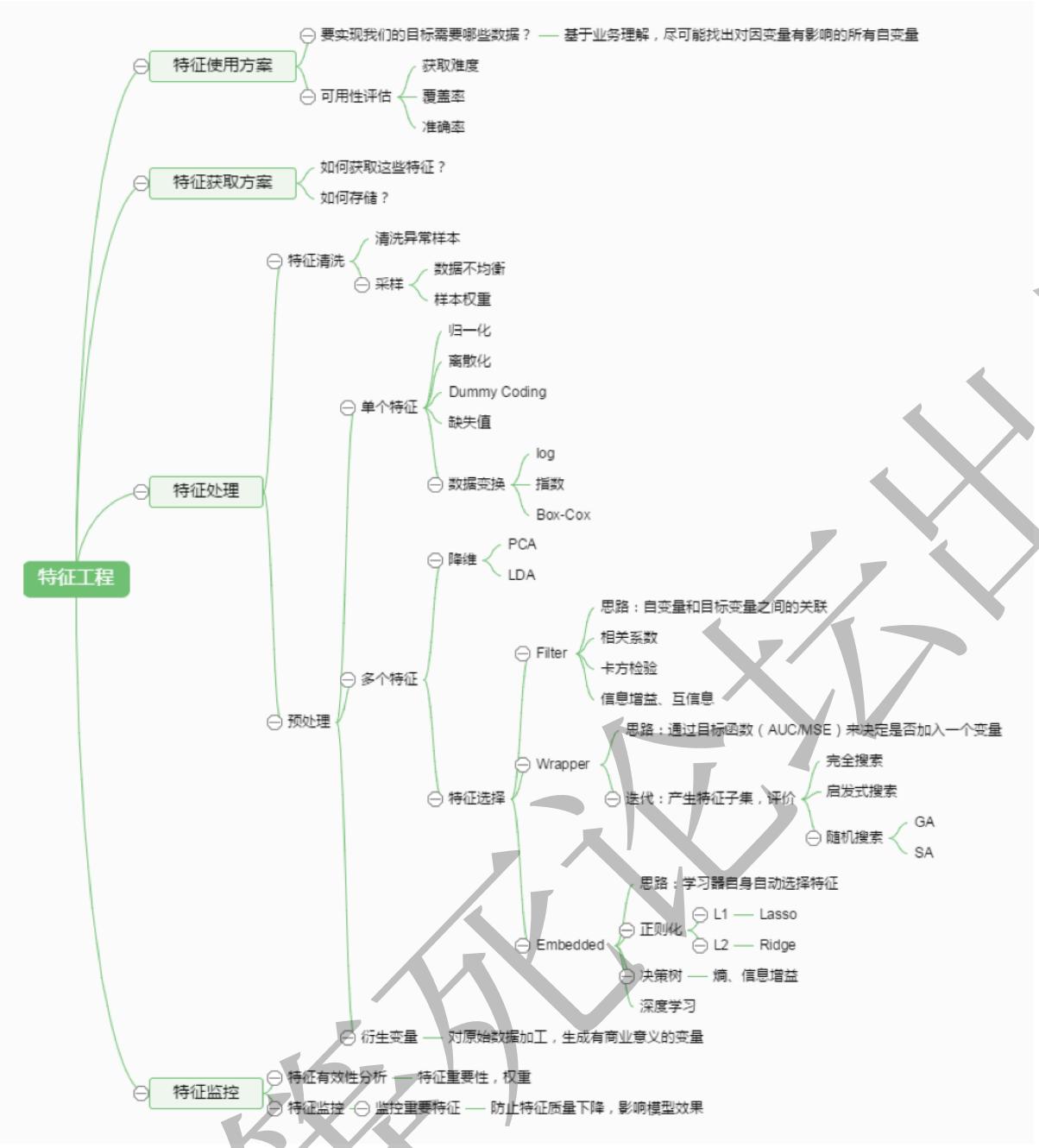
梯度消失会造成权值更新缓慢，模型训练难度增加。造成梯度消失的一个原因是，许多激活函数将输出值挤压在很小的区间内，在激活函数两端较大范围的定义域内梯度为0，造成学习停止。



59、到底什么是特征工程?



60、你知道有哪些数据处理和特征工程的处理？



62、数据不平衡问题

隐藏解析　4条讨论

解析：

这主要是由于数据分布不平衡造成的。解决方法如下：

采样，对小样本加噪声采样，对大样本进行下采样
数据生成，利用已知样本生成新的样本
进行特殊的加权，如在Adaboost中或者SVM中
采用对不平衡数据集不敏感的算法
改变评价标准：用AUC/ROC来进行评价
采用Bagging/Boosting/ensemble等方法
在设计模型的时候考虑数据的先验分布

63、特征比数据量还大时，选择什么样的分类器？

隐藏解析　1条讨论　63 / 148

解析：

线性分类器，因为维度高的时候，数据一般在维度空间里面会比较稀疏，很有可能线性可分。

64、常见的分类算法有哪些？他们各自的优缺点是什么？

贝叶斯分类法

优点：

1）所需估计的参数少，对于缺失数据不敏感。

2）有着坚实的数学基础，以及稳定的分类效率。

缺点：

1）假设属性之间相互独立，这往往并不成立。（喜欢吃番茄、鸡蛋，却不喜欢吃番茄炒蛋）。

2）需要知道先验概率。

3）分类决策存在错误率。

决策树

优点：

1）不需要任何领域知识或参数假设。

2）适合高维数据。

3）简单易于理解。

4）短时间内处理大量数据，得到可行且效果较好的结果。

5）能够同时处理数据型和常规性属性。

缺点：

1）对于各类别样本数量不一致数据，信息增益偏向于那些具有更多数值的特征。

2）易于过拟合。

3）忽略属性之间的相关性。

4）不支持在线学习。

支持向量机

优点：

1）可以解决小样本下机器学习的问题。

2）提高泛化性能。

3）可以解决高维、非线性问题。超高维文本分类仍受欢迎。

4）避免神经网络结构选择和局部极小的问题。

缺点：

1）对缺失数据敏感。

2）内存消耗大，难以解释。

3）运行和调参略烦人。

Logistic回归

优点：

1）速度快。

2）简单易于理解，直接看到各个特征的权重。

3）能容易地更新模型吸收新的数据。

4）如果想要一个概率框架，动态调整分类阈值。

缺点：

特征处理复杂。需要归一化和较多的特征工程。

K近邻

优点：

1）思想简单，理论成熟，既可以用来做分类也可以用来做回归；

2）可用于非线性分类；

3）训练时间复杂度为O(n)；

4）准确度高，对数据没有假设，对outlier不敏感；

缺点：

1）计算量太大

2）对于样本分类不均衡的问题，会产生误判。

3）需要大量的内存。

4）输出的可解释性不强。

神经网络

优点：

1）分类准确率高。

2）并行处理能力强。

3）分布式存储和学习能力强。

4）鲁棒性较强，不易受噪声影响。

缺点：

1）需要大量参数（网络拓扑、阀值、阈值）。

2）结果难以解释。

3）训练时间过长。

Adaboost

优点：

1）adaboost是一种有很高精度的分类器。

2）可以使用各种方法构建子分类器，Adaboost算法提供的是框架。

3）当使用简单分类器时，计算出的结果是可以理解的。而且弱分类器构造极其简单。

4）简单，不用做特征筛选。

5）不用担心overfitting。

缺点：

对outlier比较敏感

1）随机梯度下降

优点：可以一定程度上解决局部最优解的问题

缺点：收敛速度较慢

2）批量梯度下降

优点：容易陷入局部最优解

缺点：收敛速度较快

3）mini_batch 梯度下降

综合随机梯度下降和批量梯度下降的优缺点，提取的一个中和的方法。

4）牛顿法

牛顿法在迭代的时候，需要计算 Hessian 矩阵，当维度较高的时候，计算 Hessian 矩阵比较困难。

5）拟牛顿法

拟牛顿法是为了改进牛顿法在迭代过程中，计算 Hessian 矩阵而提取的算法，它采用的方式是通过逼近 Hessian 的方式来进行求解。

从每个batch的数据来区分

梯度下降：每次使用全部数据集进行训练

优点：得到的是最优解

缺点：运行速度慢，内存可能不够

随机梯度下降：每次使用一个数据进行训练

优点：训练速度快，无内存问题

缺点：容易震荡，可能达不到最优解

Mini-batch梯度下降

优点：训练速度快，无内存问题，震荡较少

缺点：可能达不到最优解

从优化方法上来分：

随机梯度下降（SGD）

缺点

选择合适的learning rate比较难

对于所有的参数使用同样的learning rate

容易收敛到局部最优

可能困在saddle point

SGD+Momentum

优点：

积累动量，加速训练

局部极值附近震荡时，由于动量，跳出陷阱

梯度方向发生变化时，动量缓解动荡。

Nesterov Mementum

与Mementum类似，优点：

避免前进太快

提高灵敏度

AdaGrad

优点：

控制学习率，每一个分量有各自不同的学习率

适合稀疏数据

缺点

依赖一个全局学习率

学习率设置太大，其影响过于敏感

后期，调整学习率的分母积累的太大，导致学习率很低，提前结束训练。

RMSProp

优点：

解决了后期提前结束的问题。

缺点：

依然依赖全局学习率

Adam

Adagrad和RMSProp的合体

优点：

结合了Adagrad善于处理稀疏梯度和RMSprop善于处理非平稳目标的优点

为不同的参数计算不同的自适应学习率

也适用于大多非凸优化 - 适用于大数据集和高维空间

牛顿法

牛顿法在迭代的时候，需要计算Hessian矩阵，当维度较高的时候，计算 Hessian矩阵比较困难

拟牛顿法

拟牛顿法是为了改进牛顿法在迭代过程中，计算Hessian矩阵而提取的算法，它采用的方式是通过逼近Hessian的方式来进行求解。

67、特征向量的归一化方法有哪些?

隐藏解析 2条讨论

解析:

线性函数转换，表达式如下:
y=(x-MinValue)/(MaxValue-MinValue)
对数函数转换，表达式如下:
y=log10 (x)
反余切函数转换 ，表达式如下:
y=arctan(x)*2/PI
减去均值，除以方差:
y=(x-means)/ variance

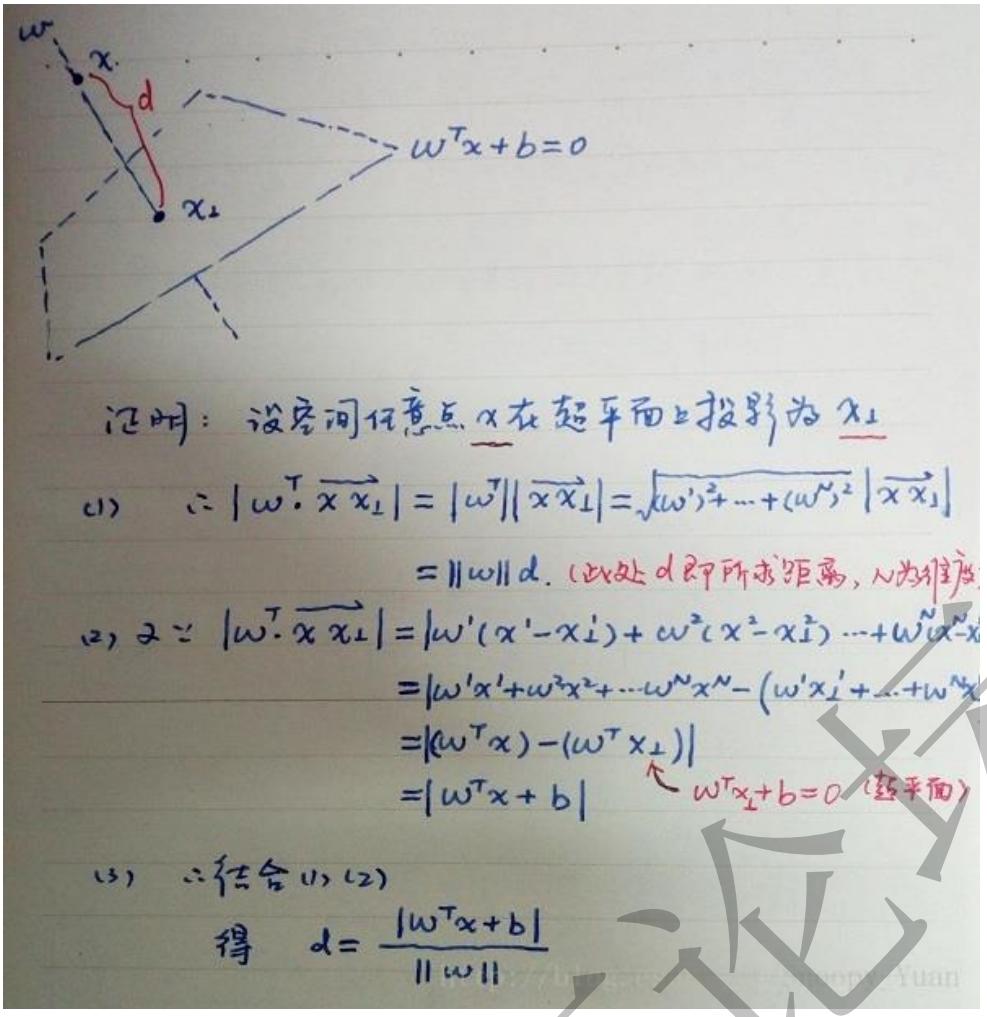
68、RF与GBDT之间的区别与联系?

隐藏解析 4条讨论 68

解析:

1）相同点：都是由多棵树组成，最终的结果都是由多棵树一起决定。
2）不同点:
a 组成随机森林的树可以分类树也可以是回归树，而GBDT只由回归树组成
b 组成随机森林的树可以并行生成，而GBDT是串行生成
c 随机森林的结果是多数表决表决的，而GBDT则是多棵树累加之和
d 随机森林对异常值不敏感，而GBDT对异常值比较敏感
e 随机森林是减少模型的方差，而GBDT是减少模型的偏差
f 随机森林不需要进行特征归一化。而GBDT则需要进行特征归一化

69、**6.1** 试证明样本空间中任意点 $\boldsymbol{x}$ 到超平面 $(\boldsymbol{w}, b)$ 的距离为式(6.2).



70、请比较下EM算法、HMM、CRF

隐藏解析　3条讨论　70 / 148　上一题　下一题

解析：

这三个放在一起不是很恰当，但是有互相有关联，所以就放在这里一起说了。注意重点关注算法的思想。

（1）EM算法

EM算法是用于含有隐变量模型的极大似然估计或者极大后验估计，有两步组成：E步，求期望（expectation）；M步，求极大（maxmization）。本质上EM算法还是一个迭代算法，通过不断用上一代参数对隐变量的估计来对当前变量进行计算，直到收敛。

注意：EM算法是对初值敏感的，而且EM是不断求解下界的极大化逼近求解对数似然函数的极大化的算法，也就是说EM算法不能保证找到全局最优值。对于EM的导出方法也应该掌握。

（2）HMM算法

隐马尔可夫模型是用于标注问题的生成模型。有几个参数（π，A，B）：初始状态概率向量π，状态转移矩阵A，观测概率矩阵B。称为马尔科夫模型的三要素。

马尔科夫三个基本问题：

概率计算问题：给定模型和观测序列，计算模型下观测序列输出的概率。–》前向后向算法

学习问题：已知观测序列，估计模型参数，即用极大似然估计来估计参数。–》Baum-Welch(也就是EM算法)和极大似然估计。

预测问题：已知模型和观测序列，求解对应的状态序列。–》近似算法（贪心算法）和维比特算法（动态规划求最优路径）

（3）条件随机场CRF

给定一组输入随机变量的条件下另一组输出随机变量的条件概率分布密度。条件随机场假设输出变量构成马尔科夫随机场，而我们平时看到的大多是线性链条随机场，也就是由输入对输出进行预测的判别模型。求解方法为极大似然估计或正则化的极大似然估计。

之所以总把HMM和CRF进行比较，主要是因为CRF和HMM都利用了图的知识，但是CRF利用的是马尔科夫随机场（无向图），而HMM的基础是贝叶斯网络（有向图）。而且CRF也有：概率计算问题、学习问题和预测问题。大致计算方法和HMM类似，只不过不需要EM算法进行学习问题。

（4）HMM和CRF对比

其根本还是在于基本的理念不同，一个是生成模型，一个是判别模型，这也就导致了求解方式的不同。

71、带核的SVM为什么能分类非线性问题?

隐藏解析 4条讨论 71 / 148 上一题 下一题

解析:

核函数的本质是两个函数的内积，通过核函数将其隐射到高维空间，在高维空间非线性问题转化为线性问题, SVM得到超平面是高维空间的线性分类平面, 如图:

72、请说说常用核函数及核函数的条件

隐藏解析 参与讨论 72 / 148 上一题 下一题

解析:

我们通常说的核函数指的是正定和函数，其充要条件是对于任意的x属于X，要求K对应的Gram矩阵要是半正定矩阵。RBF核径向基，这类函数取值依赖于特定点间的距离，所以拉普拉斯核其实也是径向基核。SVM关键是选取核函数的类型，常用核函数主要有线性内核，多项式内核，径向基内核（RBF），sigmoid核。

线性核函数

$$\kappa(x, x_i) = x \cdot x_i$$

线性核，主要用于线性可分的情况，我们可以看到特征空间到输入空间的维度是一样的，其参数少速度快，对于线性可分数据，其分类效果很理想，因此我们通常首先尝试用线性核函数来做分类，看看效果如何，如果不行再换别的

多项式核函数

$$\kappa(x, x_i) = ((x \cdot x_i) + 1)^d$$

多项式核函数可以实现将低维的输入空间映射到高纬的特征空间，但是多项式核函数的参数多，当多项式的阶数比较高的时候，核矩阵的元素值将趋于无穷大或者无穷小，计算复杂度会大到无法计算。

高斯（RBF）核函数

$$\kappa(x, x_i) = exp(-\frac{||x - x_i||^2}{\delta^2})$$

高斯径向基函数是一种局部性强的核函数，其可以将一个样本映射到一个更高维的空间内，该核函数是应用最广的一个，无论大样本还是小样本都有比较好的性能，而且其相对于多项式核函数参数要少，因此大多数情况下在不知道用什么核函数的时候，优先使用高斯核函数。

sigmoid核函数

$$\kappa(x, x_i) = tanh(\eta < x, x_i > + \theta)$$

采用sigmoid核函数，支持向量机实现的就是一种多层神经网络。

因此，在选用核函数的时候，如果我们对我们的数据有一定的先验知识，就利用先验来选择符合数据分布的核函数；如果不知道的话，通常使用交叉验证的方法，来试用不同的核函数，误差最下的即为效果最好的核函数，或者也可以将多个核函数结合起来，形成混合核函数。

在吴恩达的课上，也曾经给出过一系列的选择核函数的方法：
如果特征的数量大到和样本数量差不多，则选用LR或者线性核的SVM；
如果特征的数量小，样本的数量正常，则选用SVM+高斯核函数；
如果特征的数量小，而样本的数量很大，则需要手工添加一些特征从而变成第一种情况。

73、请具体说说Boosting和Bagging的区别

隐藏解析 参与讨论 73 / 148 上一题 下一题

解析：

（1）Bagging之随机森林

随机森林改变了决策树容易过拟合的问题，这主要是由两个操作所优化的：

1）Boostrap从袋内有放回的抽取样本值

2）每次随机抽取一定数量的特征（通常为sqr(n)）。

分类问题：采用Bagging投票的方式选择类别频次最高的

回归问题：直接取每颗树结果的平均值。

常见参数 误差分析 优点 缺点

1、树最大深度

2、树的个数

3、节点上的最小样本数

4、特征数(sqr(n)) oob(out-of-bag)

将各个树的未采样样本作为预测样本统计误差作为误分率 可以并行计算

不需要特征选择

可以总结出特征重要性

可以处理缺失数据

不需要额外设计测试集 在回归上不能输出连续结果

（2）Boosting之AdaBoost

Boosting的本质实际上是一个加法模型，通过改变训练样本权重学习多个分类器并进行一些线性组合。而Adaboost就是加法模型+指数损失函数+前项分布算法。Adaboost就是从弱分类器出发反复训练，在其中不断调整数据权重或者是概率分布，同时提高前一轮被弱分类器误分的样本的权值。最后用分类器进行投票表决（但是分类器的重要性不同）。

（3）Boosting之GBDT

将基分类器变成二叉树，回归用二叉回归树，分类用二叉分类树。和上面的Adaboost相比，回归树的损失函数为平方损失，同样可以用指数损失函数定义分类问题。但是对于一般损失函数怎么计算呢？GBDT（梯度提升决策树）是为了解决一般损失函数的优化问题，方法是用损失函数的负梯度在当前模型的值来模拟回归问题中残差的近似值。

注：由于GBDT很容易出现过拟合的问题，所以推荐的GBDT深度不要超过6，而随机森林可以在15以上。

（4）Boosting之Xgboost

这个工具主要有以下几个特点：

支持线性分类器

可以自定义损失函数，并且可以用二阶偏导

加入了正则化项：叶节点数、每个叶节点输出score的L2-norm

支持特征抽样

在一定情况下支持并行，只有在建树的阶段才会用到，每个节点可以并行的寻找分裂特征。

74、逻辑回归相关问题

（4）LR和SVM对比

首先，LR和SVM最大的区别在于损失函数的选择，LR的损失函数为Log损失（或者说是逻辑损失都可以）、而SVM的损失函数为hinge loss。

其次，两者都是线性模型。

最后，SVM只考虑支持向量（也就是和分类相关的少数点）

（5）LR和随机森林区别

随机森林等树算法都是非线性的，而LR是线性的。LR更侧重全局优化，而树模型主要是局部的优化。

（6）常用的优化方法

逻辑回归本身是可以用公式求解的，但是因为需要求逆的复杂度太高，所以才引入了梯度下降算法。

一阶方法：梯度下降、随机梯度下降、mini 随机梯度下降降法。随机梯度下降不但速度上比原始梯度下降要快，局部最优化问题时可以一定程度上抑制局部最优解的发生。

二阶方法：牛顿法、拟牛顿法：

这里详细说一下牛顿法的基本原理和牛顿法的应用方式。牛顿法其实就是通过切线与x轴的交点不断更新切线的位置，直到达到曲线与x轴的交点得到方程解。在实际应用中我们因为常常要求解凸优化问题，也就是要求解函数一阶导数为0的位置，而牛顿法恰好可以给这种问题提供解决方法。实际应用中牛顿法首先选择一个点作为起始点，并进行一次二阶泰勒展开得到导数为0的点进行一个更新，直到达到要求，这时牛顿法也就成了二阶求解问题，比一阶方法更快。我们常常看到的x通常为一个多维向量，这也就引出了Hessian矩阵的概念（就是x的二阶导数矩阵）。

缺点：牛顿法是定长迭代，没有步长因子，所以不能保证函数值稳定的下降，严重时甚至会失败。还有就是牛顿法要求函数一定是二阶可导的。而且计算Hessian矩阵的逆复杂度很大。

拟牛顿法： 不用二阶偏导而是构造出Hessian矩阵的近似正定对称矩阵的方法称为拟牛顿法。拟牛顿法的思路就是用一个特别的表达形式来模拟Hessian矩阵或者是他的逆使得表达式满足拟牛顿条件。主要有DFP法（逼近Hession的逆）、BFGS（直接逼近Hession矩阵）、L-BFGS（可以减少BFGS所需的存储空间）。

75、什么是共线性，跟过拟合有什么关联？

隐藏解析　1条讨论　75 / 148

解析：

共线性：多变量线性回归中，变量之间由于存在高度相关关系而使回归估计不准确。
共线性会造成冗余，导致过拟合。
解决方法：排除变量的相关性 / 加入权重正则。

77、用贝叶斯机率说明Dropout的原理

隐藏解析　2条讨论　77 / 148　上一题　下一题

解析：

回想一下使用Bagging学习,我们定义 k 个不同的模型,从训练集有替换采样 构造 k 个不同的数据集,然后在训练集上训练模型 i。

Dropout的目标是在指数 级数量的神经网络上近似这个过程。Dropout训练与Bagging训练不太一样。在Bagging的情况下,所有模型是独立 的。

在Dropout的情况下,模型是共享参数的,其中每个模型继承的父神经网络参 数的不同子集。参数共享使得在有限可用的内存下代表指数数量的模型变得可能。 在Bagging的情况下,每一个模型在其相应训练集上训练到收敛。
在Dropout的情况下,通常大部分模型都没有显式地被训练,通常该模型很大,以致到宇宙毁灭都不 能采样所有可能的子网络。取而代之的是,可能的子网络的一小部分训练单个步骤,参数共享导致剩余的子网络能有好的参数设定。

78、对于维度极低的特征，选择线性还是非线性分类器？

隐藏解析　参与讨论　78 / 148

解析：

非线性分类器，低维空间可能很多特征都跑到一起了，导致线性不可分。
1. 如果Feature的数量很大，跟样本数量差不多，这时候选用LR或者是Linear Kernel的SVM
2. 如果Feature的数量比较小，样本数量一般，不算大也不算小，选用SVM+Gaussian Kernel
3. 如果Feature的数量比较小，而样本数量很多，需要手工添加一些feature变成第一种情况。

79、请问怎么处理特征向量的缺失值

隐藏解析 参与讨论 79 / 148

解析：

一方面，缺失值较多.直接将该特征舍弃掉，否则可能反倒会带入较大的noise，对结果造成不良影响。

另一方面缺失值较少,其余的特征缺失值都在10%以内，我们可以采取很多的方式来处理:

1) 把NaN直接作为一个特征，假设用0表示；

2) 用均值填充；

3) 用随机森林等算法预测填充。

80、SVM、LR、决策树的对比。

隐藏解析 5条讨论 80 / 148 上一题

解析：

模型复杂度：SVM支持核函数，可处理线性非线性问题;LR模型简单，训练速度快，适合处理线性问题;决策树容易过拟合，需要进行剪枝

损失函数：SVM hinge loss; LR L2正则化; adaboost 指数损失

数据敏感度：SVM添加容忍度对outlier不敏感，只关心支持向量，且需要先做归一化; LR对远点敏感

数据量：数据量大就用LR，数据量小且特征少就用SVM非线性核

81、什么是ill-condition病态问题?

隐藏解析 2条讨论 81 / 148 上一题

解析：

训练完的模型，测试样本稍作修改就会得到差别很大的结果，就是病态问题，模型对未知数据的预测能力很差，即泛化误差大。

82、简述KNN最近邻分类算法的过程?

隐藏解析 4条讨论 82 / 148

解析：

1. 计算测试样本和训练样本中每个样本点的距离（常见的距离度量有欧式距离，马氏距离等）；
2. 对上面所有的距离值进行排序；
3. 选前 k 个最小距离的样本；
4. 根据这 k 个样本的标签进行投票，得到最后的分类类别；

83、常用的聚类划分方式有哪些？列举代表算法。

隐藏解析　参与讨论　83 / 148

解析：

1. 基于划分的聚类:K-means，k-medoids，CLARANS。
2. 基于层次的聚类：AGNES（自底向上），DIANA（自上向下）。
3. 基于密度的聚类：DBSACN，OPTICS，BIRCH(CF-Tree)，CURE。
4. 基于网格的方法：STING，WaveCluster。
5. 基于模型的聚类：EM,SOM，COBWEB。

84、什么是偏差与方差?

隐藏解析　1条讨论　84 / 148　上一题　下一题

解析：

泛化误差可以分解成偏差的平方加上方差加上噪声。偏差度量了学习算法的期望预测和真实结果的偏离程度，刻画了学习算法本身的拟合能力，方差度量了同样大小的训练集的变动所导致的学习性能的变化，刻画了数据扰动所造成的影响，噪声表达了当前任务上任何学习算法所能达到的期望泛化误差下界，刻画了问题本身的难度。偏差和方差一般称为bias和variance，一般训练程度越强，偏差越小，方差越大，泛化误差一般在中间有一个最小值，如果偏差较大，方差较小，此时一般称为欠拟合，而偏差较小，方差较大称为过拟合。

偏差:

$$E_D[(\overline{f(x)}-y)^2]$$

方差:

$$E_D[(f(x,D)-\overline{f(x)})^2]$$

85、解决bias和Variance问题的方法是什么?

隐藏解析　2条讨论　85 / 148　上一题　下一题

解析：

High bias解决方案:Boosting、复杂模型（非线性模型、增加神经网络中的层）、更多特征
High Variance解决方案：bagging、简化模型、降维

具体而言
高偏差, 可以用boosting模型, 对预测残差进行优化, 直接降低了偏差. 也可以用高模型容量的复杂模型(比如非线性模型, 深度神经网络), 更多的特征, 来增加对样本的拟合度.
高方差, 一般使用平均值法, 比如bagging, 或者模型简化/降维方法, 来降低方差.

高偏差和高方差都是不好的, 我们应该加以避免. 但是它们又是此消彼长的关系, 所以必须权衡考虑. 一般情况下, 交叉验证训练可以取得比较好的平衡:
将原始样本均分成K组, 将每组样本分别做一次验证集,其余的K-1组子集数据作为训练集,这样会得到K个模型, 这K个模型可以并发训练以加速. 用这K个模型最终的验证集的分类准确率的平均数作为此K-CV下分类器的性能指标. K一般大于等于3, 而K-CV 的实验共需要建立 k 个models，并计算 k 次 test sets 的平均预测正确率。

在实作上，k 要够大才能使各回合中的 训练样本数够多，一般而言 k=10 (作为一个经验参数)算是相当足够了。

86、采用 EM 算法求解的模型有哪些，为什么不用牛顿法或梯度下降法？

隐藏解析 参与讨论 86 / 148 上一题 下一题

解析：

用EM算法求解的模型一般有GMM或者协同过滤，k-means其实也属于EM。EM算法一定会收敛，但是可能收敛到局部最优。由于求和的项数将随着隐变量的数目指数上升，会给梯度计算带来麻烦。

88、什么是OOB？随机森林中OOB是如何计算的，它有什么优缺点？

隐藏解析 2条讨论 88 / 148 上一题 下一题

解析：

bagging方法中Bootstrap每次约有1/3的样本不会出现在Bootstrap所采集的样本集合中，当然也就没有参加决策树的建立，把这1/3的数据称为袋外数据oob（out of bag），它可以用于取代测试集误差估计方法。

袋外数据(oob)误差的计算方法如下：
对于已经生成的随机森林,用袋外数据测试其性能,假设袋外数据总数为O,用这O个袋外数据作为输入,带进之前已经生成的随机森林分类器,分类器会给出O个数据相应的分类,因为这O条数据的类型是已知的,则用正确的分类与随机森林分类器的结果进行比较,统计随机森林分类器分类错误的数目,设为X,则袋外数据误差大小=X/O;这已经经过证明是无偏估计的,所以在随机森林算法中不需要再进行交叉验证或者单独的测试集来获取测试集误差的无偏估计。

89、推导朴素贝叶斯分类 P(c|d)，文档 d（由若干 word 组成），求该文档属于类别 c 的概率， 并说明公式中哪些概率可以利用训练集计算得到

隐藏解析 1条讨论 89 / 148 上一题 下一题

解析：

根据贝叶斯公式P(c|d)=（P(c)P(d|c)/P(d)）

这里，分母P(d)不必计算，因为对于每个类都是相等的。 分子中，P(c)是每个类别的先验概率，可以从训练集直接统计，P(d|c)根据独立性假设，可以写成如下 P(d|c)=￥P(wi|c)（￥符号表示对d中每个词i在c类下概率的连乘），P(wi|c)也可以从训练集直接统计得到。 至此，对未知类别的d进行分类时，类别为c=argmaxP(c)￥P(wi|c)。

91、请写出你对VC维的理解和认识

隐藏解析 参与讨论 91 / 148

解析：

VC维是模型的复杂程度，模型假设空间越大，VC维越高。某种程度上说，VC维给机器学习可学性提供了理论支撑。

1. 测试集合的loss是否和训练集合的loss接近？VC维越小，理论越接近，越不容易overfitting。
2. 训练集合的loss是否足够小？VC维越大，loss理论越小，越不容易underfitting。

我们对模型添加的正则项可以对模型复杂度(VC维)进行控制，平衡这两个部分。

92、kmeans聚类中，如何确定k的大小

1 按需选择
2 观察法
3 手肘法
4 Gap Statistics方法

一、按需选择

简单地说就是按照建模的需求和目的来选择聚类的个数。比如说，一个游戏公司想把所有玩家做聚类分析，分成顶级、高级、中级、菜鸟四类，那么K=4；如果房地产公司想把当地的商品房分成高中低三档，那么K=3。按需选择虽然合理，但是未必能保证在做K-Means时能够得到清晰的分界线。

二、观察法

就是用肉眼看，看这些点大概聚成几堆。这个方法虽然简单，但是同时也模棱两可。

第一张是原始点，第二张分成了两类，第三张是三类，第四张是四类。至于K到底是选3还是选4，可能每个人都有不同的选择。

观察法的另一个缺陷就是：原始数据维数要低，一般是两维（平面散点）或者三维（立体散点），否则人类肉眼则无法观察。对于高维数据，我们通常利用PCA降维，然后再进行肉眼观察。

三、手肘法Elbow Method

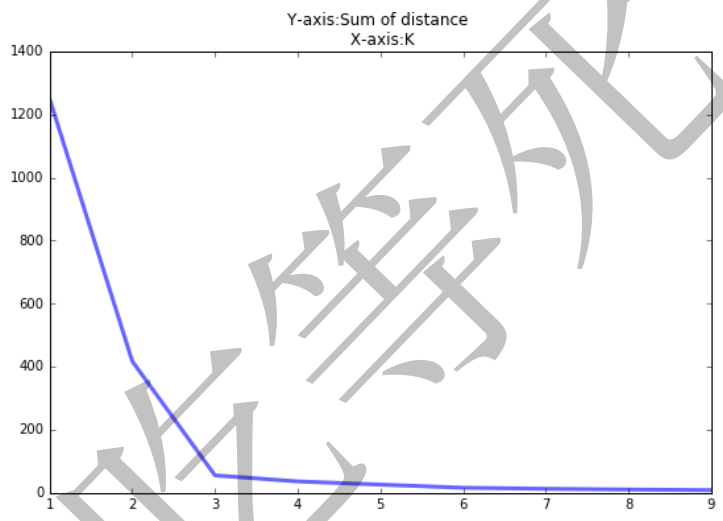
手肘法本质上也是一种间接的观察法。这里需要一点K-Means的背景知识。当K-Means算法完成后，我们将得到K个聚类的中心点Mi,i=1,2,⋯,Ki=1,2,⋯,K。以及每个原始点所对应的聚类Ci，i=1,2,⋯,K。我们通常采用所有样本点到它所在的聚类的中心点的距离的和作为模型的度量，记为DK。

$$D_K = \sum_{i=1}^{K} \sum_{X \in C_i} \|X - M_i\|$$

这里距离可以采用欧式距离。

对于不同的K，最后我们会得到不同的中心点和聚类，所有会有不同的度量。

我们把K作为横坐标，DK作为纵坐标，可以得到下面的折线。



很显然K越大，距离和越小。但是我们注意到K=3是一个拐点，就像是我们的肘部一样，K=1到3下降很快，K=3之后趋于平稳。手肘法认为这个拐点就是最佳的K。

手肘法是一个经验方法，而且肉眼观察也因人而异，特别是遇到模棱两可的时候。相比于直接观察法，手肘法的一个优点是，适用于高维的样本数据。有时候人们也会把手肘法用于不同的度量上，如组内方差组间方差比。

四、Gap Statistic法

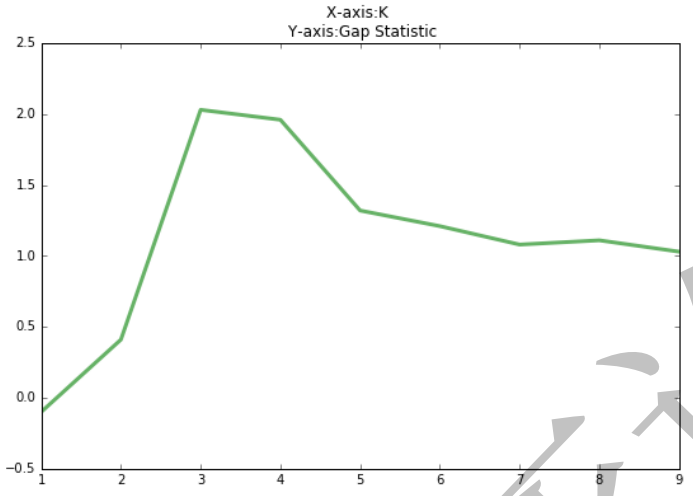
这个方法是源自斯坦福几个machine learning大牛的paper Estimating the number of clusters in a data set via the gap statistic 。

这里我们要继续使用上面的DK。Gap Statistic的定义为

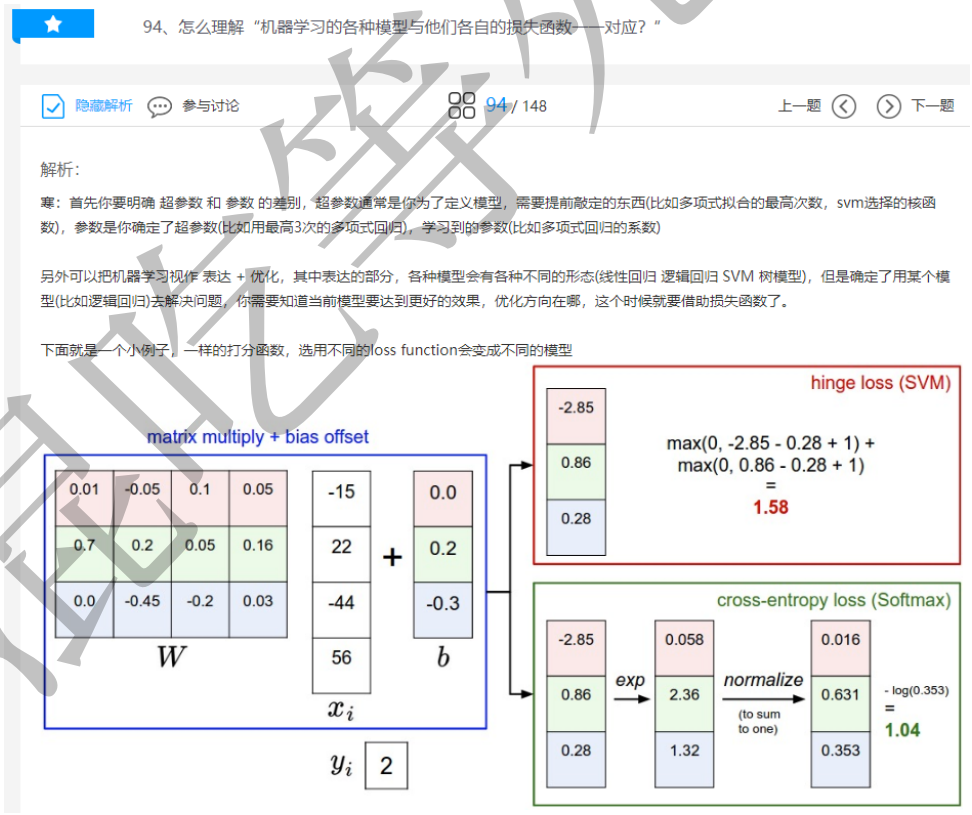
$$Gap(K) = \mathrm{E}(\log D_k) - \log D_k$$

这里E(logDk)指的是logDk的期望。这个数值通常通过蒙特卡洛模拟产生，我们在样本里所在的矩形区域中（高维的话就是立方体区域）按照均匀分布随机地产生和原始样本数一样多的随机样本，并对这个随机样本做K-Means，从而得到一个DK。如此往复多次，通常20次，我们可以得到20个logDK。对这20个数值求平均值，就得到了E(logDK)的近似值。最终可以计算Gap Statisitc。而Gap statistic取得最大值所对应的K就是最佳的K。

用上图的例子，我们计算了K=1,2,..9对应的Gap Statisitc.



Gap Statistic的优点是，我们不再需要肉眼了。我们只需要找到最大gap statistic所对应的K即可。所以这种方法也适用于“批量化作业”。如果我们要做1000次聚类分析，不需要肉眼去看1000次了。



95、给你一个有1000列和1百万行的训练数据集。这个数据集是基于分类问题的。

经理要求你来降低该数据集的维度以减少模型计算时间。你的机器内存有限。你会怎么做？（你可以自由做各种实际操作假设）

隐藏解析 参与讨论 95 / 148 上一题 下一题

解析：

答：你的面试官应该非常了解很难在有限的内存上处理高维的数据。以下是你可以使用的处理方法：

1.由于我们的RAM很小，首先要关闭机器上正在运行的其他程序，包括网页浏览器，以确保大部分内存可以使用。
2.我们可以随机采样数据集。这意味着，我们可以创建一个较小的数据集，比如有1000个变量和30万行，然后做计算。
3.为了降低维度，我们可以把数值变量和分类变量分开，同时删掉相关联的变量。对于数值变量，我们将使用相关性分析。对于分类变量，我们可以用卡方检验。
4.另外，我们还可以使用PCA（主成分分析），并挑选可以解释在数据集中有最大偏差的成分。
5.利用在线学习算法，如VowpalWabbit（在Python中可用）是一个可能的选择。
6.利用Stochastic GradientDescent（随机梯度下降）法建立线性模型也很有帮助。
7.我们也可以用我们对业务的理解来估计各预测变量对响应变量的影响大小。但是，这是一个主观的方法，如果没有找出有用的预测变量可能会导致信息的显著丢失。

96、问2：在PCA中有必要做旋转变换吗？

如果有必要，为什么？如果你没有旋转变换那些成分，会发生什么情况？

隐藏解析 1条讨论 96 / 148 上一题 下一题

解析：

答：是的，旋转（正交）是必要的，因为它把由主成分捕获的方差之间的差异最大化。这使得主成分更容易解释。但是不要忘记我们做PCA的目的是选择更少的主成分（与特征变量个数相较而言），那些选上的主成分能够解释数据集中最大方差。

通过做旋转，各主成分的相对位置不发生变化，它只能改变点的实际坐标。如果我们没有旋转主成分，PCA的效果会减弱，那样我们会不得不选择更多个主成分来解释数据集里的方差。

97、给你一个数据集，这个数据集有缺失值，且这些缺失值分布在离中值有1个标准偏差的范围内。百分之多少的数据不会受到影响？为什么？

隐藏解析 1条讨论 97 / 148 上一题 下一题

解析：

答：这个问题给了你足够的提示来开始思考！由于数据分布在中位数附近，让我们先假设这是一个正态分布。

我们知道，在一个正态分布中，约有68%的数据位于跟平均数（或众数、中位数）1个标准差范围内的，那样剩下的约32%的数据是不受影响的。

因此，约有32%的数据将不受到缺失值的影响。

98、给你一个癌症检测的数据集。你已经建好了分类模型，取得了96%的精度。为什么你还是不满意你的模型性能？你可以做些什么呢？

隐藏解析 3条讨论 98 / 148 上一题 下一题

解析：

答：如果你分析过足够多的数据集，你应该可以判断出来癌症检测结果是不平衡数据。在不平衡数据集中，精度不应该被用来作为衡量模型的标准，因为96%（按给定的）可能只有正确预测多数分类，但我们感兴趣是那些少数分类（4%），是那些被诊断出癌症的人。

因此，为了评价模型的性能，应该用灵敏度（真阳性率），特异性（真阴性率），F值用来确定这个分类器的“聪明”程度。如果在那4%的数据上表现不好，我们可以采取以下步骤：

1.我们可以使用欠采样、过采样或SMOTE让数据平衡。
2.我们可以通过概率验证和利用AUC-ROC曲线找到最佳阀值来调整预测阀值。
3.我们可以给分类分配权重，那样较少的分类获得较大的权重。
4.我们还可以使用异常检测。

99、解释朴素贝叶斯算法里面的先验概率、似然估计和边际似然估计？

隐藏解析 1条讨论 99 / 148 上一题 下一题

解析：

先验概率就是因变量（二分法）在数据集中的比例。这是在你没有任何进一步的信息的时候，是对分类能做出的最接近的猜测。
例如，在一个数据集中，因变量是二进制的（1和0）。例如，1（垃圾邮件）的比例为70%和0（非垃圾邮件）的为30%。因此，我们可以估算出任何新的电子邮件有70%的概率被归类为垃圾邮件。

似然估计是在其他一些变量的给定的情况下，一个观测值被分类为1的概率。例如，“FREE”这个词在以前的垃圾邮件使用的概率就是似然估计。
边际似然估计就是，“FREE”这个词在任何消息中使用的概率

100、你正在一个时间序列数据集上工作。经理要求你建立一个高精度的模型。你开始用决策树算法，因为你知道它在所有类型数据上的表现都不错。
后来，你尝试了时间序列回归模型，并得到了比决策树模型更高的精度。

这种情况会发生吗？为什么？

隐藏解析 7条讨论 100 / 148 上一题 下一题

解析：

众所周知，时间序列数据有线性关系。另一方面，决策树算法是已知的检测非线性交互最好的算法。

为什么决策树没能提供好的预测的原因是它不能像回归模型一样做到对线性关系的那么好的映射。

因此，我们知道了如果我们有一个满足线性假设的数据集，一个线性回归模型能提供强大的预测。