算法面试题

1、基于每日用户搜索内容，假设只有少量已知商品的情况下，如何根据用户搜索内容获取平台内没有的新商品？





答案：这是一条类似于分词“新词获取问题”，答案是基于信息熵＋聚合度。

这边需要考虑排除，首先做stop词库，先去除形容词等。

信息熵：比如用户搜索“曲面显示屏 白色”，假设现在我们的商品库中没有显示屏这个商品，我们需要判断“显示屏”是否是潜在的商品，我们需要考虑“显示屏”左词、右词出现的可能。换句话说，如果大家都在搜索“显示屏”商品的话，会出现大量的“便宜显示屏”、“可旋转显示屏”、“显示屏 黑色”等搜索短语，根据信息熵计算公式-p∑logp，“显示屏”前后出现的词语类别越多，信息熵越大，代表用户搜索的需求越旺盛，“显示屏”越有可能是没有的商品。

聚合度：根据信息熵的理论也会出现“显示”等高频出现的干扰词，再用聚合度，比如先计算出p(“显示”)、p(“屏”)、或p(“显”)、p(“示屏”)的概率，如果“显示”是一个高频合理的搜索词的话，p(“显示”)\*p(“屏”)应该远远大于p(“显示屏”)，p(“显”)＊p(“示屏”)应该远远大于p(“显示屏”)的概率，而实际电商搜索中，用户连贯搜索“显示屏”的概率才是远超其它。

**2**

为什么logistic回归的要用sigmoid函数？优缺点？

答案：

优点：

1.数据压缩能力，将数据规约在［0，1］之间  
2.导数形式优秀，方便计算

缺点：

1.容易梯度消失，x稍大的情况下就趋近一条水平线  
2.非0中心化，在神经网络算法等情况下，造成反向传播时权重的全正全负的情况。

为什么要用？

答案1: logistic是基于Bernoulli分布的假设，也就是y|X~Bernoulli分布，而Bernoulli分布的指数族的形式就是1/(1+exp(-z))

其实还有一个答案二，我当时没想起来，如就是：  
对于logistic多分类而言，  
x1、x2、...、xn，属于k类的概率正比于：



我们回到2类：  
x1、x2、...xn属于1的概率是：



分子分母同除以分子极为1/(1+exp(-z))，z＝w11-w01，个人觉得这样的证明才有说服力

**3**

对比牛顿法、梯度下降法的关系

讲真，大学学完牛顿法就丢了，一时没回答出来，回来整理如下：

答案：牛顿法快于梯度下降法，且是梯度下降法的极限。

首先，我们有展开式：  
f′(x+Δx)=f′(x)+f″(x)∗Δx  
Δx=−μ∗f′(x)  
合并两个式子，有：  
f′(x+Δx)=f′(x)+f″(x)∗(−μ∗f′(x))  
令f′(x+Δx)＝0，  
μ＝1/f″(x)，极为牛顿法在随机梯度下降中的μ

**4**

两个盒子，50个红球，50个白球，问如何放球，抽到红球的概率最高？（每个盒子必须有球）

答案：一个盒子1个红球，另外一个盒子剩余的99个球

先假设第一个盒子放x个红球，y个白球，另外的一个盒子里面就有50-x红球，50-y个白球.  
求的目标函数：p＝1/2(x/(x+y))+1/2((50-x)/(100-x-y))  
subject to. x+y>0 & 100-x-y>0

常规解法如上，被坑了一手的是，面试的说没有常规解，我回来思考了半天，可能是盒子里面的排练顺序有差异，上层的抽取概率>下层的抽取概率，所以需要通过EM算法，先得到若干次抽取的结果下，每层的最大概率密度函数，再结合上述的结果去回答。

**5**

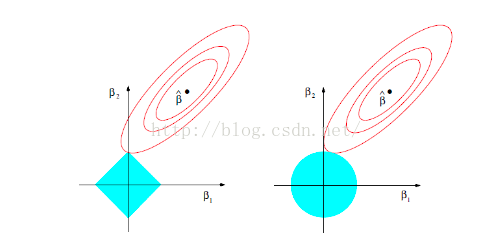
常见的正则化有是么，有什么作用，为什么l1是会把feature压缩到0而l2做不到？

答案：

(1)l1,l2正则化  
l1对应python里面numpy.linalg.norm(ord=1)  
形如|w1|+|w2|+|w3|+...  
l2对应python里面numpy.linalg.norm(ord=2)  
形如w12+w22+w3^2+...

(2)防止过拟合  
其它防止过拟合的方法还有：  
1.增加数据量  
2.采取bagging算法，抽样训练数据  
\*\*

(3)画图解决



左边的l1，右边的l2，  
l1在作图只要不是特殊情况下与正方形的边相切，一定是与某个顶点优先相交，那必然存在横纵坐标轴中的一个系数为0，起到对变量的筛选的作用。

l2的时候，其实就可以看作是上面这个蓝色的圆，在这个圆的限制下，点可以是圆上的任意一点，所以q＝2的时候也叫做岭回归，岭回归是起不到压缩变量的作用的，在这个图里也是可以看出来的。

**6**

**分类模型如何选择？如何判断效果？如何计算AUC？你最熟悉的ensemble Classification model是什么？**

我这边参考了《Do we Need Hundreds of Classifiers to Solve Real World Classification Problems》里面的结论，有兴趣的自行去搜。

答案

整体上讲：数据量越大，神经网络越好；维度越多，bagging算法越优秀；数据量上不多不少的情况下，SVM效果最好；

常用判断：roc、auc、ks、f1值、recall等；

AUC计算方法：roc曲线下方的面积积分即可，或者大数定律的投点实验

最熟悉的集成分类模型，我说的是randomforest，详述了原理及实际应用的注意点，后来我问了面试管，主要在这块想了解的是实际解决的相关项目的真实性：

1、randomforest是由若干颗cart树构成的，每棵树尽情生长不枝剪，最后采取加权投票或者均值的方式确定输出值；

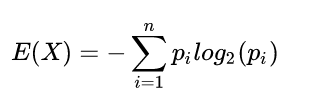
2、每棵树的数据是采取bagging式的随机抽取特征及数据样本，两颗树之间的数据有可能会重复；

3、一般流程会先以sqrt(feature\_number)作为每次输入的特征数，采取grid\_search的方法观察tree的数量由0-500，oob的变化  
这边被打断了，解释什么叫做oob，也就是out of bag，每次抽取的数据样本进行训练，没有被抽取到的数据作为检验样本，检验样本上的误差就叫做oob；

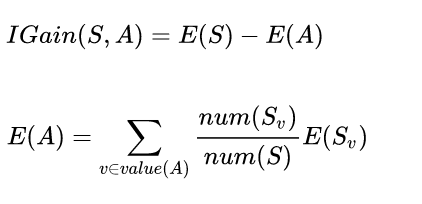
4、根据实际要求的精度上后期可以跟进调整：每次输入的特征个数、每棵树的最大深度、每个节点的分支方式（GINI还是信息增益率）、子节点最少数据量、父节点最少数据量等等。

这边又被打断了，问，什么叫做信息增益率？

首先熵的计算如下：



信息增益如下：



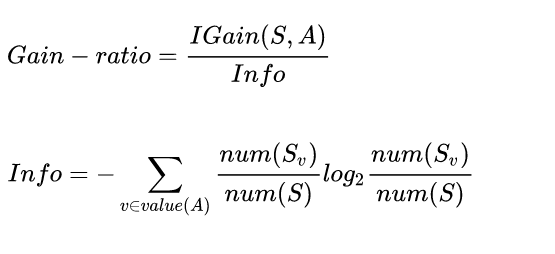
比如14个人，好人5个坏人9个。这14个人被通过性别划分开，10个男性中3个坏人，7个好人；4个女性中2个坏人，2个好人。

信息增益就是:

IGain＝(-5/14)log(5/14)+(-9/14)log(9/14)-(10/14(-3/10log(3/10)-7/10log(7/10))+4/14(-1/2log(1/2)-1/2log(1/2)))

看到这样的计算方式，必然会存在问题，假设我们身份证为区分类别的化，每个身份证号码都是独一无二的，势必存在存在1/n\*log(1)=0这样的最佳划分，但是这样的结果就是将所有的情况分别作为子节点，很明显没有意义，所以引出下面的信息增益率。

信息增益率就是:



比如上面分人的例子，Info＝-10/14log(10/14)-4/14log(4/14)  
很明显也可以看出，当你划分的子类别越多，你的info会越大，Gain\_ratio就越小，信息增益率就越低，惩罚了刚才身份证分类这种行为。

这也是id3和c4.5之间最大的差异，c4.5以信息增益率代替率id3里面的信息增益，除此之外，id3只能对分类变量处理而c4.5既可以分类变量也可以连续变量，还是很强的，同时他们都可以做多分类，而后续的cart等做多分类的成本会增加（叠加的方式）。

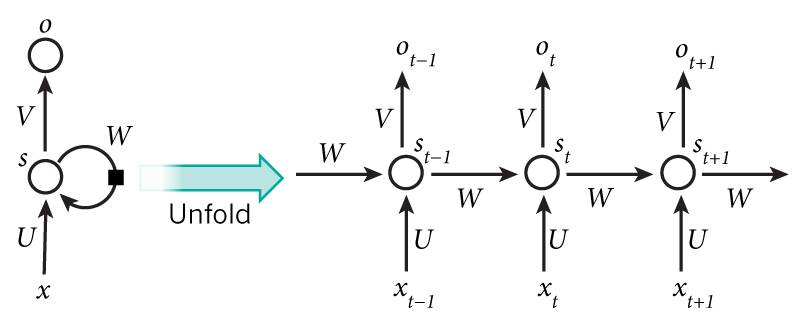
其实，这些都很基础但是时间长了，真的很绕人，我也是先自己默默的在纸上画了挺久才和面试管聊，有点出乎我的意料。

**7**

**循环神经网络中介绍一个你熟悉的？**

我说的是LSTM。

首先，先跑出了循环的机制，同时点明了RNN潜在隐藏节点对output的影响，做了下图：



当前的预测结果，与input及上次的layer1节点下的结果相关。

正向循环：  
节点1的值 = sigmoid(np.dot(输入参数,神经元1) + np.dot(上次节点1的值,潜在神经元))  
输出值＝sigmoid(np.dot(节点1的值,神经元2))

误差计算：  
真实y－输出值

delta：  
节点2处的deltas=误差计算\*sigmoid(np.dot(节点1的值,神经元2))／(1-sigmoid(np.dot(节点1的值,神经元2)))

反向修正神经元：  
神经元2 += (节点1的值).T.dot(节点2处的delta)  
潜藏神经元 += (上次的节点1的值).T.dot(节点1处的delta)  
神经元1 += 输入值.T.dot(节点1处的delta)

核心强调了：sigmoid(np.dot(输入参数,神经元1) + np.dot(上次节点1的值,潜在神经元))，输出值与输出值及上次节点1处的输入值有关。

然后讲了简单的在语义识别的实际作用。

**8**

**kmeans的原理及如何选择k？如何选择初始点？**

原理是送分题。

原理：在给定K值和K个初始类簇中心点的情况下,把每个点(亦即数据记录)分到离其最近的类簇中心点所代表的类簇中，优点在于易于理解和计算，缺点也是很明显，数据一多的情况计算量极大，且标签feature定义距离的难度大。

K的选择，我答的一般，欢迎大家补充，

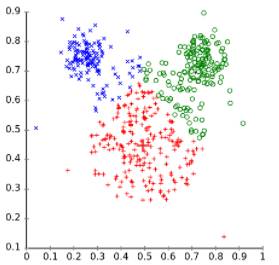
1、根据具体的业务需求，实际需求确定最后聚成的类的个数

2、grid\_search去试，看那种距离下，损失函数最小（其实这样回答不好，数据量大的情况下，机会不可能）  
这边的损失函数类别较多，可能包括组内间距和／组外间距和等

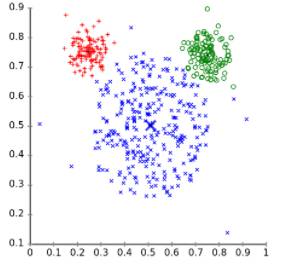
3、随机抽样下的层次聚类作为预参考  
理论上，随机采样的数据分布满足原来的数据集的分布，尤其是大量采样次数下的情况，针对每一个较小的数据集合采取层次聚类确定最后的聚类个数，再针对原始的数据集合进行kmeans聚类

如何选取初始点？

这个问题我被问过好多次，其实，不管是r或者python里面，或者大家日常使用中都是默认的随机选取，然后通过多次k-折等方法不断的去迭代，其实这样存在的问题就是如果初始点随机选取的有误，导致无论这么迭代都得不到最优的点，如：



随机初始点

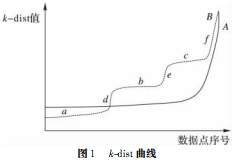


修正初始点

在随机初始点的情况下，红色区域的部分点被蓝色和绿色侵占为己点，修正初始点，也就是将随机初始点的聚类中心全部上移的情况下，蓝色点区收回了原属于自己的点区。

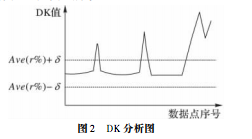
之前我恶补过一片论文：《K-means 初始聚类中心的选择算法》，里面提出了两个指标来衡量：

1.k-dist



某个点 p 到它的第k 个最近点的距离为点 p 的 k-dist 值。点的 k-dist 半径范围内至少包含k + 1 个点，理论上同一个聚类中改变k值不会引起k-dist值明显变化。将 k-dist 值由小到大排序，a、b、c表示平缓点，d，e，f为跃迁点。

2.DK图



k-dist 图中相邻两点的 k-dist 值之差记为 DK。k-dist 图中相邻两点pm和pm－1的 k-dist的差为DKm=k-distm －k-distm－1 ( m ＞ 1) 。由于 k-distm 非递减，显然 DKm ＞ 0。DK 值接近的连续邻近点处于 k-dist 图的同一条平缓曲线上，即处于  
同一个密度层次; DK 值大幅跳动的点处于密度转折曲线或噪声曲线上。

3.选择  
对 DK 值从小到大排序，得到 DK 标准范围δ。依据 DK 标准范围内对应的数据点的分布情况，在 k-dist 图中找出 k' 个平缓曲线，代表 k' 个主要密度水平。选择每个密度水平的第一个点作为初始聚类中心。

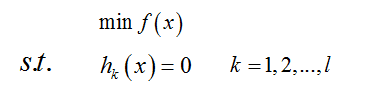
重复若干次，得到若干组的优化聚类中心，在根据优化聚类中心组下的组内间距和／组外间距和判断那个点组为最优点组。

其实这样的开销也挺大的，目前也没有看到其它比较易理解的kmeans的初始点计算的方式。

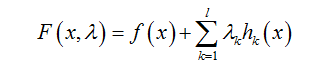
**9**

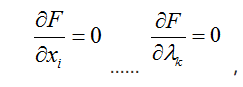
**大致讲解一下最优化中拉格朗日乘子法的思路？KKT是什么？**

当我们求解一个函数的最小值，且这个函数也被某些确定的限制条件限制的时候：

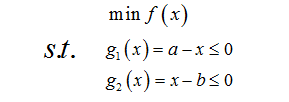


我们可以将限制条件加入f(x)中一同进行后续的偏导计算：

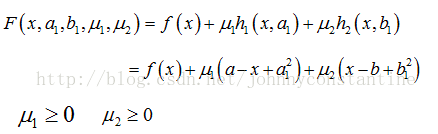


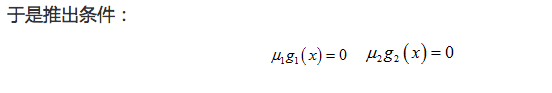


至于KKT我了解的其实不多，也是回来之后恶补了一下，通过例子入手：



求解上面这个问题的化，我们需要考虑构造两个约束变量a1，b1，使得  
h1(x，a1)＝g1(x)＋a1^2＝a－x＋a1^2=0  
h2(x，b1)＝g2(x)＋b1^2＝b－x＋b1^2=0  
在根据普通拉格朗日乘子的方法对下面公式的每一项求偏导：





这个条件就是KKT条件  
其实我觉得，http://www.cnblogs.com/zhangchaoyang/articles/2726873.html，这篇文章写的挺好的，想要详细了解的可以仔细参考一下。

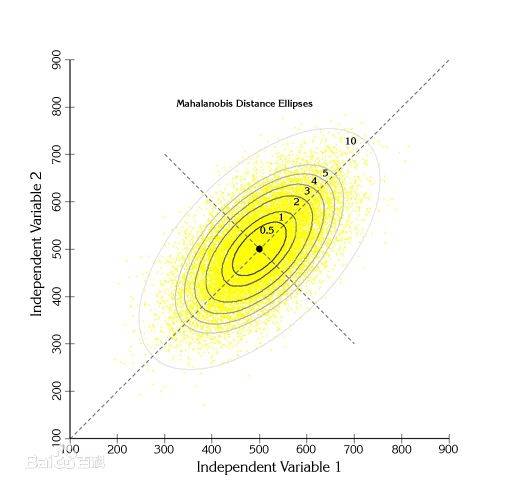
**10**

**听说你做过风控，异常点检测你用过什么办法？**

之前正好整理过，内心大喜：  
1、6个西格玛的原理  
2、箱式图大于3/2QI＋Q3，小于Q1－3/2Qi  
3、基于距离离群检测（聚类），包括欧式、马氏距离、街道距离  
这边被打断了，问了马氏距离的细节，好处：



追问了协方差Sigma怎么算：  
Cov(X,Y)=E(XY)-E(X)E(Y)  
追问了什么时候用马氏距离比较好：  
举例很有名的曲线分布图，如下：



4.pca的基于特征值压缩的方法

5.基于isolation forest识别的方法。

这边被追问了一次原理：

method:   
1.从原始数据中随机选择一个属性feature；   
2.从原始数据中随机选择该属性的下的一个样本值value；   
3.根据feature下的value对每条记录进行分类，把小于value的记录放在左子集，把大于等于value的记录放在右子集；   
4.repeat 1-3 until： 　　　　  
4.1.传入的数据集只有一条记录或者多条一样的记录； 　　　　  
4.2.树的高度达到了限定高度；

以s(x,n)为判断数据是否异常的衡量指标。





其中，h(x)为x对应的节点深度，c(n)为样本可信度，s(x,n)~[0,1]，正常数据来讲s(x,n)小于0.8，s(x,n)越靠近1，数据异常的可能性越大。

