完成于 2017 年 12 月 3 日，范鑫

58（9.18 笔试（概率论，矩阵论，编程题为文本框），9.22 面试，11. 14 offer）
一面（技术）：
Q：手写代码：给一个排序数组，找到两个和为 k 的数，要求时间 o(n)
A：（使用双指针解决，语言 c++）

Q：讲讲你的项目
A：搜狐视频推荐项目

Q：进程和线程的区别
A：进程有独立的地址空间，线程是进程的组成部分
Q：可以有一个进程不包含任何线程么
A：不可以

Q：java 如何停止一个线程
A：线程执行完就停止了
Q：嗯？
A：可以用 sleep,wait
Q：sleep 和 wait 有什么区别
A：sleep 不释放对象锁，wait 释放对象锁

Q：spark 中 case class 与 class 的区别
A：不知道，但是 case class 调参的时候经常用到

Q：hive 中如何解决数据倾斜
A：将一个参数改为 true，或者在表连接时把空置改为随机字符串
Q：把什么参数改为 true？
A：忘记了
Q：这样为什么可以避免数据倾斜？
A：如果某个 key 数量很多，可以将其分到不同的 reduce 中
Q：一个大表和一个小表 join，哪个放左边
A：小表放左边
Q：为什么
A：小表中 key 值少，先进内存可以减少连接次数

Q：讲讲 L1 和 L2 正则，有什么区别
A：都是回归中防止过拟合的操作，L1 在目标函数中加入参数的一范数，L2 加入二范数，最终结果其中 L1 将某些参数置为 0，L2 将某些参数降为接近于 0
Q：SVM 支持向量体现在哪
A：KKT 条件中的 ag(x)=0,保证不是支持向量的点的系数为 0

二面（技术）：聊项目中的难点和突出点
三面（hr）

Vivo（9.24 笔试（概率论，矩阵论，无编程题），9.25 面试，9.27sp offer）：

一面（技术）：

Q：项目

A：搜狐视频推荐项目

Q：Spark 中 map 和 foreach 的区别

A：用过，但是有什么区别不知道

Q：transform 和 action 的区别

A：没听过这两个概念

Q：（解释了这两个概念，transform 只记录，action 才会真正操作）

A：哦，我实际操作中见过这个，但是没总结为概念

Q：Spark 中 map 和 reduce 在什么数据结构上操作？

A：tuple

Q：你们平时怎么使用 spark

A：打成 jar 包，spark-submit 提交

Q：哦，我们在 spark-shell 中调试

Q：你们用几个核跑？

A：15 个 executor，每个 10　core

Q：太少，我们 1w+个核

二面（hr）

美团：（9.14 笔试（1.4/2ac），10.18 面试，offer：a 档白菜）

一面（技术）：

Q：四次挥手

A：（讲了一下，画了个图）

Q：为什么要四次挥手

A：不知道

Q：死锁的原因

A：同时占用一个资源，请求有环路，互相释放等待

Q：如何避免

A：破坏这三个条件，比如用银行家算法预先计算是否会死锁

Q：手写代码：去掉链表中倒数低 k 个节点

A：特殊情况太多了，写不完

Q：那先分析一下有什么特殊情况，比如说有环（呵呵，答案都说出来了）

A：比如头节点为空，比如没有 k 个节点，比如有环

Q：链表的环入口怎么找到？

A：双指针，一个每次走 1，一个每次走 2，相遇则有环，然后设置一个指针回链表首部，每次同时动 1，相遇的地方为环入口

Q：证明一下为什么

A：（思考 5 分钟后用方程证明）

Q：讲讲你的论文

A：（讲了论文，对方听不太懂）

Q：发在什么级别的会议上

A：三无会议，有可能升级为 C

Q：哦，我发过 A+的

A：（什么操作）厉害

Q：你觉得给你映像最深的论文是哪篇

A：group lasso

Q：你代码量多少

A：c++ 1W，java 几千

二面（技术）：

Q：项目

A：搜狐视频推荐项目

Q：讲讲 gbdt

A：n 棵二叉回归树，每一棵树拟合前面的负梯度

Q：spark 中 map().reduce().collection()，map 执行几次

A：两次

Q：为什么

A：有两个 action，map 操作得记录两次

Q：怎么让它只执行一次
A：Cache

Q：spark 的数据倾斜
A：某些 reduce 端分配到的数据量远远大于其他端，数据分配不均衡
Q：你们是如何解决的
A：没解决，我觉得时间还可以忍受
Q：那你觉得该怎么解决
A：（把 hive 那套讲了一遍）

Q：手写代码：给一个升序数组，判断能组成几个三角形
A：三层暴力循环
Q：你觉得怎么还可以再快点
A：没想出来，但我觉得得用双指针
Q：（提示）
A：左右两根指针放在数组首位为双指针，用第三根指针定三角形范围。若能找到，后面的指针移到第三根指针的位置；找不到，第一根指针加 1。

三面（技术）：
Q：用最快的速度让我了解你的论文
A：（你们这是要成立美团亚洲研究院么）讲了讲论文

Q：你平时如何学习机器学习，如何读论文
A：prml，论文读最新顶级会议的相关论文，遇到不会的概念读引用论文

Q：项目
A：搜狐视频推荐项目

Q：如何实现一个在线迭代的推荐系统，思路
A：构建用户和视频向量，每次有记录动态更新向量，推荐时选择点积大的 item.

四面（hr）

滴滴（9.5 号笔试（2/2ac），9 月 29 号一二面，10 月 19 号三面，三面通过但是无消息）

一面（技术）：

Q：项目

A：搜狐视频推荐项目

Q：手写代码：长度为 n 的数组中存着 1-（n-1）的整数，找到其中的重复元素

A：置换法，把位置 1 的数字置换为 1，依次类推，则 o(n)可找出

Q：介绍你知道的分类模型，聚类模型

A：分类：感知机，svm，决策树，随机森林，gbdt，xgboost，神经网络，逻辑斯蒂回归，knn

聚类：Kmeans，其他软聚类方法

二面（技术）：

Q：每人发 5 张牌，选三张最大的求和，a 赢 b 的概率；假如 a 作弊了摸了 6 张牌，a 赢 b 的概率

A：（思考了 5 分钟）不会

Q：那就简单一点，假如我是 JQK，你赢我的概率

A：我赢你的情况有 JKK,QQK,QKK,KKK 这四种情况，总共有 xxx 中情况，两个一除即可

Q：你确定只有四种情况赢我？

A：嗯

Q：换道题，一个用户发出打车请求，如何最块找到附近 1 公里的车辆

A：sqrt((x-x0)^2+(y-y0)^2)，小于 1 即可

Q：这个太慢了，能快点么

A：把地图划分为 1km*1km 的格子，在格子里找

Q：嗯，那在一个格子里，怎么找小于 200m 的车子

A：sqrt((x-x0)^2+(y-y0)^2)，小于 0.2 即可

Q：…能快点么

A：先用街区距离算一下，先得到一部分解，街区距离比 0.2 大的再用公式算

Q：你这个能节约多少计算量

A：1/sqrt(2)/sqrt(2)=0.5

Q：如果有那种特别优秀的司机，可以放宽距离条件，怎么设计

A：假如他的优秀程度为 n,则若距离人为 n 个格子也把他推荐过来

三面（技术）：

Q：手写代码：快排

A：（逻辑没写错，但有语法错误）

Q：快排的时间复杂度怎么推导

A：用递归式推导，一顿操作后为 o（nlogn）

Q：怎么改进快排

A：选择尽量靠中间的值作为分界点

Q：怎么选

A：不知道

Q：还有呢
A：最后还有 2-3 个元素后就不递归了，直接排

Q：项目
A：搜狐视频推荐项目

Q：如何判断一个司机是不是好司机
A：抽取用户特征，打好标签，训练模型，其他人做测试集，得到结果
Q：标签怎么打
A：抽取用户评论，打分等数据
Q：一般司机的用户差评很少，低分很少，他们用你这个方法区分不开
A：我想不出来了，我就想到这里了

网易（9.9 笔试（2/2ac），10.20 面，10.27offer）：
一面（技术）：
Q：项目
A：搜狐视频推荐项目

Q：svd 和 svd++的区别
A：svd++在计算用户特征时使用了其相关物品的隐变量
Q：svd 和协同过滤的联系和区别
A：svd 使用矩阵分解构建隐变量，协同过滤通过相似用户/物品来进行推荐；
表面看起来不同，其实本质是相同的，可以互相转化。
在矩阵分解中，相似用户分解后的隐变量时相似的。

Q：什么是互信息？
A：不知道，但是好像和熵有关
Q：什么是熵
A：（公式写了一遍）

Q：讲讲最小二乘
A：（写出目标函数，固定一个变量，对另一个变量求导，通过导数等于 0 求解，同理另一个方向再做一遍）

Q：c++，java 哪个用的多
A：我在搜狐用 scala
Q：那你说说 spark action 算子
A：count，collect，reducebykey（错的），sortbykey（错的），join（错的）

Q： left outer join 和 right outer join
A：一个左面的 key 保留，一个右面的 key 保留

二面（技术）：
Q：说几个离散分布
A：二项分布，泊松分布
Q：解释一下泊松分布
A：一段时间内某事件发生的次数

Q：如何找到离群点
A：回归分析
Q：嗯？
A：统计分析也行，找 3 西格玛外的点

Q：left outer join 和 right outer join
A：（老哥，你们的题库太小了；而且，估计进去得经常写 sql）左保留，右保留

Q：如何提取特征

A：降维，互信息，卡方分析
Q：如何解决过拟合
A：把模型复杂度加入到目标函数中

Q：java 熟练？
A：只用 java 写过作业

三面（hr）

百度（9.27 笔试（1.1/2ac），10.22 面试，10.27 白菜 offer）：

一面：

Q：手写代码：二分查找

A：（设置 begin,end,mid, c++实现）

Q：如果有重复元素如何找到最后一个

A:（修改代码判断条件）

Q：简单讲一下 LRU

A：内存页面置换算法，上一次使用时间距现在最久的被淘汰

Q：如果让你实现，你会用什么数据结构

A：栈+hash

Q：你再想想

A：优先队列+hash

Q: 是双向链表+hash

Q：如果判断两个网页的相似度

A: 分词，使用 tf-idf 构建页面向量，通过计算向量夹角计算相似度

Q: 这样做准确率太低了

A：那用 word2vec 构建词向量，构建文章矩阵计算相似度

Q：你听说过 c++ &*#!*（没听懂）么？

A: 没听过

Q：你没用 c++写过多线程么？

A：我用 java 写过

Q：linux 给一个文件 ip+信息，找到出现次数最多的 ip

A：awk -F \t '{print $1}'|unique|sort|head -1

Q:如果我要得到此 ip 后面的其他信息呢

A：awk 里写，但是具体忘记了，得百度一下

Q：你平常最常用哪种机器学习模型

A：svm

Q：讲讲 svm 和 lr 的相同点和区别吧

A：一个只看支持向量的点，距离分类面远的点再分类中作用为 0,；一个距离分类面越远的点权值越小，这点类似，但处理方法又不同

Q：你觉得哪个算法复杂度低

A：svm，因为使用了对偶，只计算支持向量的点

Q:设计一个贴吧垃圾回复过滤系统

A:设置一些敏感词，判断回复里是否有，构建 01 向量后用 svm 分类

Q:万一只有 1 个词，比如顶，赞,LZSB 这种呢？

A：设置阈值 k，小于 k 直接判定为垃圾回复

Q：万一过滤掉有用的呢？

A：...

二面（技术面）：
Q：项目
A: 搜狐视频推荐项目

Q: 给两个玻璃球，测试楼层的安全高度（试验出在哪层可以摔碎玻璃珠）
A：第一个球在一半的时候扔下去，第二个球看情况一阶一阶实验
Q：你觉得这样最快么？
A：我觉得好像是在 sqrt(n)的地方扔最好
Q：请证明
A：（推了一半，被喊停）

Q：手写代码：字符串编辑距离
A：什么是编辑距离
Q：两个字符串变为一样最少的操作数
A:（先求两个字符串的最大公共子串，再求 max(两个串与子串的长度差)）
Q：你觉得你的最大公共子串怎样能减少空间复杂度
A：每次保存最近的两行
Q：怎么计算对其位置
A：看最后一行

三面（技术面）：
Q：项目
A: 搜狐视频推荐项目

Q: svd 推导
A: （简单推推）
Q：与 svd++有什么区别
A: svd++在构建用户隐变量是使用了相关物品信息

Q：物品分布为 长尾分布，冷启动问题
A：按照与现有物品相似度推荐，相似度可用标签，描述计算

接下来聊人生

爱奇艺（9.6 笔试（3/3ac），10.25 面试，10.30 sp offer）：
一面（技术面）：
Q：字符串编辑距离
A：（和百度一样）

Q：手推 svd,xgboost
A：（和前面的公司一样）

Q：xgboost 和 gbdt 的区别
A：二阶导数，列抽样，可以加正则项

Q：堆添加元素，删除元素
A：添加：尾部添加元素后从底部根开始调整
　　删除：堆顶元素和尾部元素交换后调整根

Q：项目
A：搜狐视频推荐项目

二面（技术面）：
Q：项目
A：搜狐视频推荐项目

Q：4 亿个数字，有 25w 种情况，排序
A：计算每种情况的数量

Q： 10 个篮子分配 1000 个苹果，要求可以组成任意数字
A：2 的 0-9 次幂

Q：设计一个在线迭代的推荐系统
A：（和美团三面重复了）

Q：gbdt+lr 中 gbdt 的作用
A：处理特征吧

Q：代码题：
输入 3[K]　　输出 KKK
输入 3[K2[AB]]　输出 KABABKABABKABAB
A:递归，递归元为："数字[xxx]"

三面（技术面）：
Q：项目
A：搜狐视频推荐项目

Q：代码题：树的高度，宽度

A：什么是宽度
Q：节点数量最多的层的节点数量
A：高度 1+max(depth(left), depth(right))
A：宽度用队列进行树的层次遍历