我的AI年---机器学习在游戏数据挖掘中的应用

AI是一个很大的话题，其背后更是巨大的数学理论的支撑，这里只做一件事，只介绍一种算法，旨在记录一点点个人心得和在机器学习中的体会。不对机器学习的各个方面做介绍了。在大量的游戏数据中我们其实可以发掘出很多有价值的信息，比如根据某个用户当前的数据预测一下这个用户下面将需要什么样的道具（进而可以进行精准推荐）；根据用户参与活动的记录和行为习惯来预测出用户偏好（进而可以知道在什么样的时间给玩家推荐什么样的活动）；根据大量的购买记录来找到行为相似的玩家（进而可以给他们相互推荐为好友）；根据游戏论坛中用户语言习惯和内容的训练模型进行智能的自动回复等等等......

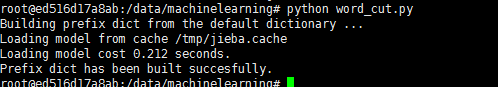
这里只拿FIFA近两个月的一小部分用户购买数据为例，来找出行为相似的用户。因为只使用了购买数据所以这里所讲的行为相似也就只是购买行为相似（如果还要加入其它因素的话就要再抓取其它方面的数据做训练和挖掘）。这在数学上来讲，其数学原理其实是很简单的，要进行数学建模的话只要两个就够了，一个是分词，一个是聚类。简单介绍一下概念，分词其实就是要想个办法让你的程序能正确理解你在说什么，能正确的进行断词断句，比如说你输入一句“南京市长江大桥”，你是希望你的程序把它理解为“南京市 / 长江大桥”呢，还是理解为"南京 / 市长 / 江大桥"呢，很明显大多数情况下你是在说“南京长江大桥”而不是“南京的市长叫江大桥”，用我们的数据来举例，比如有个玩家购买了“竞争参与奖励 绿茵礼包“，肯定是希望程序处理时把它理解为"用户领取了两个礼包一个是竞争参与奖 还有一个是绿茵礼包"，而不是"参加竞争就奖励一个绿茵礼包"，这就是分词。再简单介绍一下聚类，聚类其实就是在一大堆无规则的数据中通过我们的程序处理之后把这一堆数据分成若干类，至于分成几类分类标准是什么，我们自己也不知道，这和分类是不同的（机器学习中两大概念：clustering和classification），分类就是知道进行分类的标准是什么，再给我一条数据的时候我能计算出它是属于哪一类的，聚类就是不知道分类标准的情况下给我分好类，分完类之后每个类中的用户数据都是相似的。

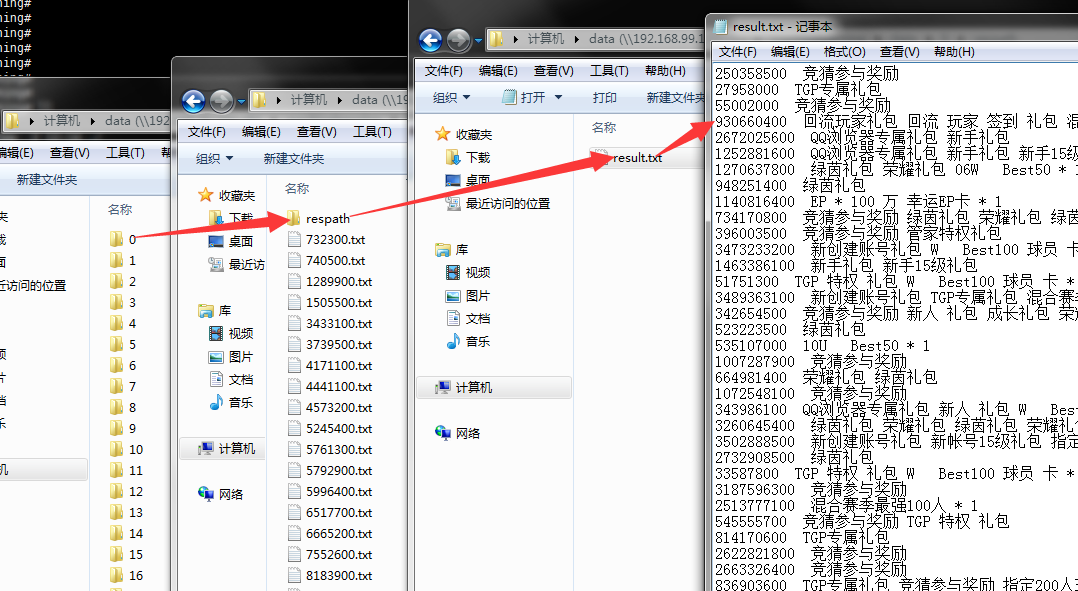
介绍完了概念了，那么具体应该怎么做呢。分词方面有不少比较成熟的实践和优秀的论文，可以在斯坦福的公开课或者谷歌论文中很容易的找到，比如用的比较多的贝叶斯或朴素贝叶斯方法，隐马尔科夫模型等等.... 这里使用贝叶斯方法进行分词，贝叶斯公式大家在概率论中肯定都见过，就是P(h | D) = P(h) \* P(D | h) / P(D)，其实就是一个求条件概率的过程，在D发生的情况下h发生的概率，应用到分词中也是一样，比如“混合赛季最强100人”，那就是出现了“混合赛季”之后出现“最强”这个词的概率，出现了“混合赛季最强”再出现“100人”的概率，也就是求一个联合概率P(An,An-1,An-2.....A1,A0)=P(A1)\*P(A2|A1)\*.....\*P(An-1)\*P(An|An-1) , 但是如果我们的语料库过于庞大，这个条件概率将会是巨大的，巨大到我们可能没办法计算，那么办呢，我们可以假设每个词语只与它前面那个词的出现有关，与再之前的无关，也就是说"混合赛季最强100人"中，"最强"只与"赛季"有关，与前面那个"混合"无关，也可以理解的，出现“绿荫赛季最强100人”毕竟是小概率事件吧，小概率事件就暂且忽略掉好了，这样假设之后这个联合概率的计算就简化为P(An,An-1)=P(An|An-1)，计算量骤降。

下面，怎么做聚类呢，同样这也是一个比较成熟的话题，在谷歌的论文以及一些技术论坛上都能找到，常用的比如高斯模型，混合高斯模型，K-Means，WaveCluster， ROCK等等，每种算法都有它自己擅长和适宜处理的数据类型，没办法说哪一个算法好哪一个算法差，这不是一个理论的论证而更是一个工程学方面的论题，是不能一概而论的，是需要根据你要进行处理的数据类型进行慢慢地调试、慢慢修改参数、慢慢优化的过程。所以，这里我们暂且选择善于处理分割问题的K-Means算法。简单介绍一下K-Means，KMeans应该算是比较常用的一种算法了，毫不夸张地说应该任何一本数据挖掘的书籍中都有这个算法的介绍，简单的说就是随机选取K个质心点，然后重复计算每个点到每一个质心点的方差（也可能是n次方差）直到收敛。怎么叫重复计算呢，就是根据这K个质心点计算一遍每个点到每个质心点的方差后就能判断出每一个点属于哪个质心点了吧（就是方差最小距离最短那个），这也叫簇，然后再重新在每一个簇的质心点，然后再对所有的点重新做一遍上述过程。拿宇宙中星星来举例，假设我们想把宇宙中所有星星划分为K个星团，那么我们先随机选K个星星做质心点，然后计算每个星星到每个质心点的距离，这样计算一遍后肯定每个星星都能找到一个距离自己最近的质心点，然后对这K个临时星团再依次重新计算它的质心（求平均坐标就可以了），然后就得出新的K个质心点了，OK那再以这新的K个质心点重新计算上述过程一直到在新簇中求出的新质心与上一次的质心是同一个星星，做到这一步那么我们就将宇宙自动划分成了K个星团了。

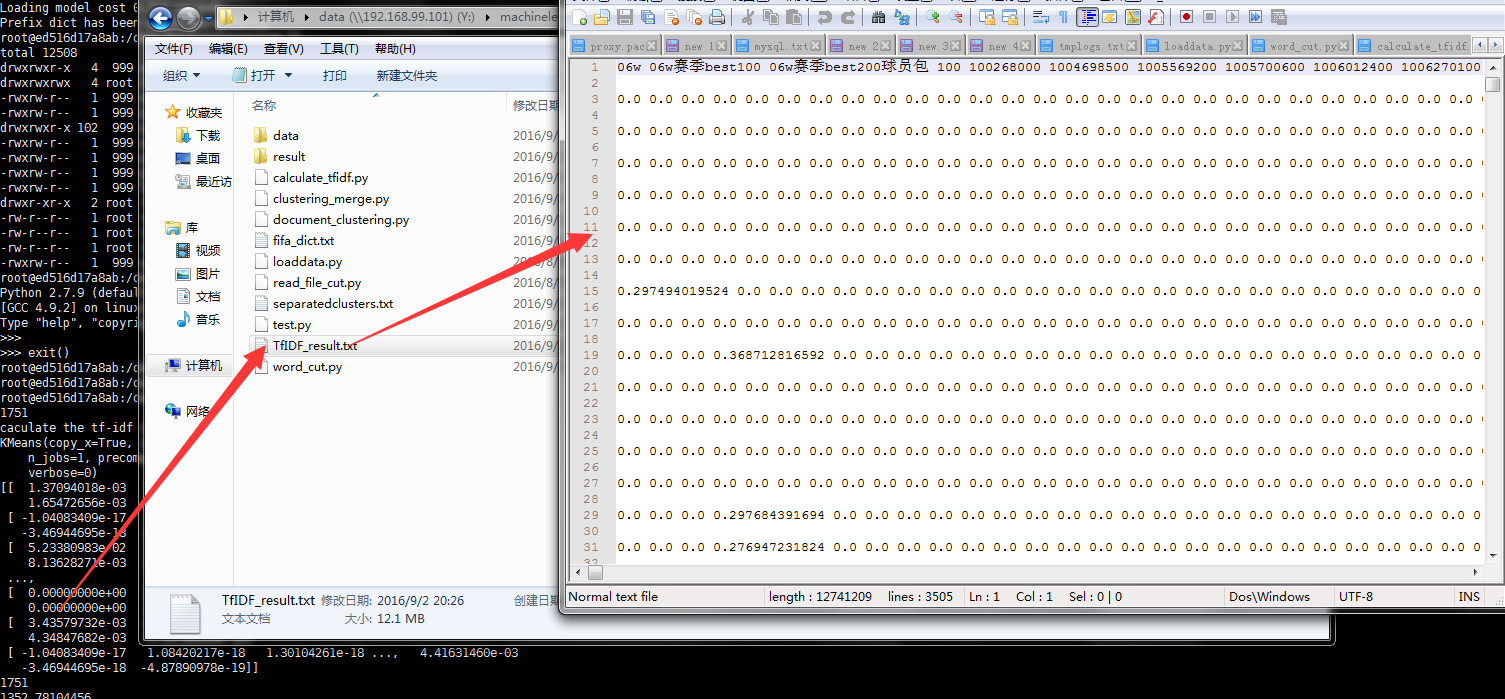
理论依据介绍完毕，下面我们来做具体的实践：

首先，我们把所有的数据做一个简单的前期处理，当然这个处理与你自己写的挖掘程序需要的格式是紧密相关的，无统一规格要求，比如我们可以按照用户QQ尾号取模100将所有用户数据暂时分成100份。然后对所有数据进行分词处理：

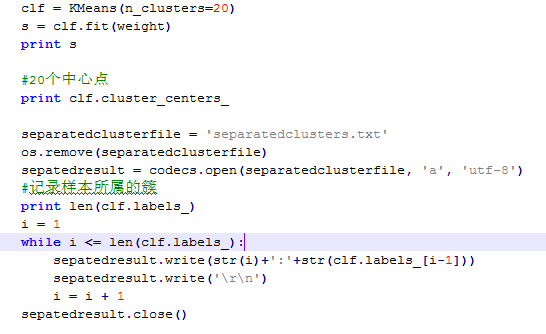


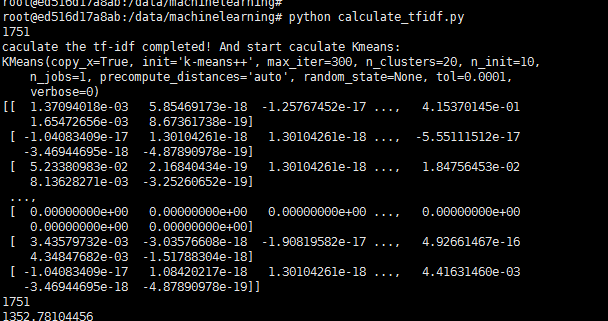


分词之后，计算其TI-IDF值（Term Frequency - Inverse Document Frequency）, TI-IDF也就是词频和反文档频率，简单地说就是如果一个词在这个文档中出现的频率高但在其他文档中出现的频率低那么就认为这个词有很好的区分能力。这里计算出来之后其实是一个概率的稀疏矩阵：



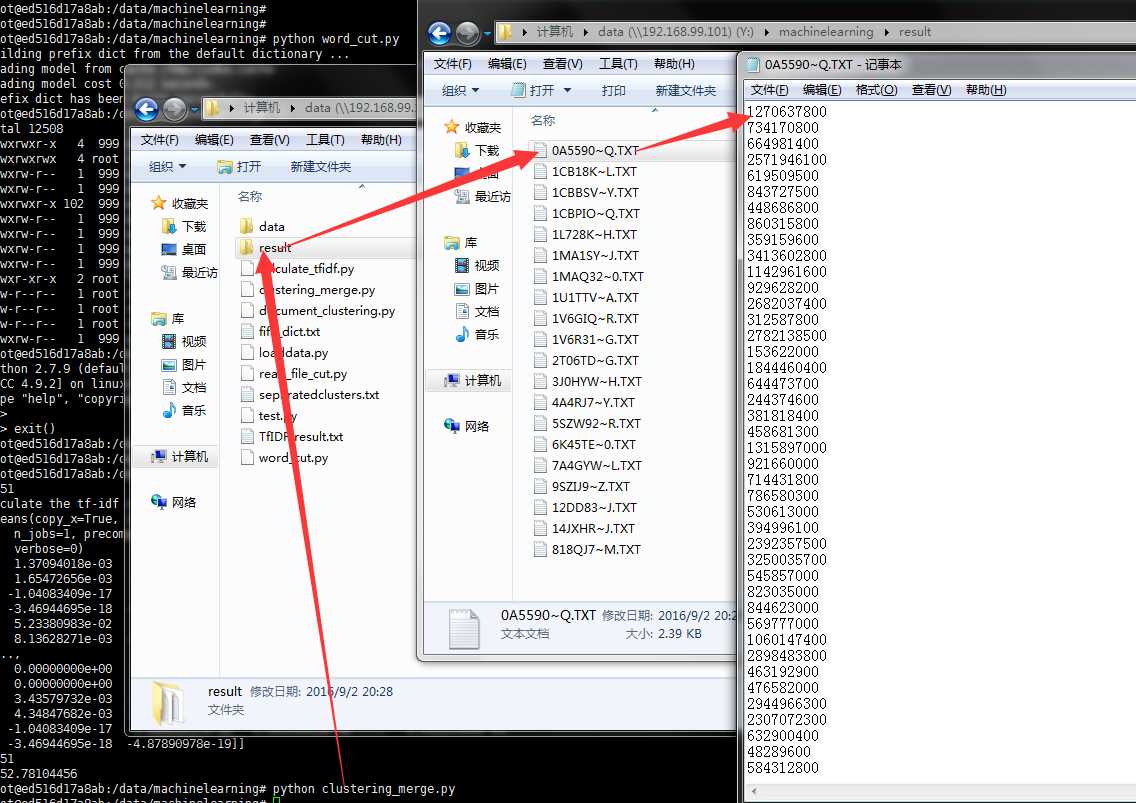
在之后我们就可以进行Kmeans聚类了，如下代码片段所示我们取20个中心点并记录每个数据所属的簇把它写到一个文件中保存起来：



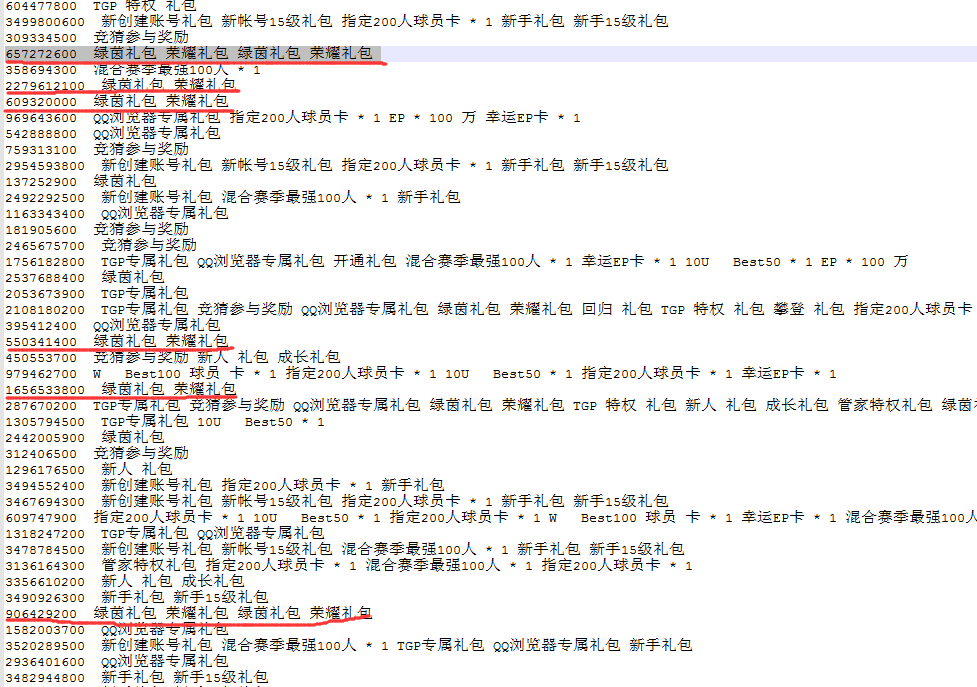


如下所示我们可以看下计算生成了20个簇，然后我们把这些数据和原始数据merge一下：

clipboard.png



然后就能看到最终的计算结果了，所有数据被分成了20类，每个类中的用户购买行为都是极其相似的，可以挑选其中的几条数据简单验证一下：



这里只是用一些简单的礼包数据在本地自己搭建的docker系统上的一些计算，后续再尝试优化计算模式和处理方法，也尽量更多样化的数据和更复杂的算法来处理，如有共同爱好的童鞋们欢迎能一起研究学习。最后引用一句话共勉“基于规则的系统是傻子，基于统计的系统是疯子”。