## Hierarchy Method

### ROCK

parameters:

k : number of clusters

theta: the threshold

concept:

1 Neighbours: If similarity of two samples reach threshold, then they will become neighbour to each other. Normally we will use JACCARD coefficient.

2 Links: the number of common neighbours between two samples.

3 Criterion Function:

手机屏幕截图

中度可信度描述已自动生成

where Ci is the cluster, k is k,f(theta) = (1-theta)/(1+theta), ni is the number of samples in Ci.

We need to scale link(), otherwise we can reach the maximum El when we put all sample in one cluster, which should not be the case.

Expected link

We assume that every sample has neighbours. Now we pick a point **P,** every two neighbours of **P** will contribute at least one to link (Since they have common neighbour **P**), so the expected link in this cluster should be

ni\* () .

4 Goodness Measure

To reach the maximum value of Criterion Function, we import a formula

文本

描述已自动生成

the denominator is

a : the expected link if we take Ci Cj as one cluster.

b: the expected link in Ci.

c: the expected link in Cj.

so, a-b-c is the expected link between two clusters (this is also work as a scaler).

If g(Ci,Cj) is of great magnitude, then it means Ci Cj should become one Cluster.

文本

中度可信度描述已自动生成

Goodness Measure is used to merge clusters to find the best k clusters.

### BIRCH

Here we are going to use something called CF Tree.

BIRCH算法利用了一个树结构来帮助我们快速的聚类，这个数结构类似于平衡B+树，一般将它称之为聚类特征树(Clustering Feature Tree，简称CF Tree)。这颗树的每一个节点是由若干个聚类特征(Clustering Feature，简称CF)组成。从下图我们可以看看聚类特征树是什么样子的：每个节点包括叶子节点都有若干个CF，而内部节点的CF有指向孩子节点的指针，所有的叶子节点用一个双向链表链接起来。

图示

描述已自动生成

Cluster Feature(CF): a triple(N,LS(向量), SS（标量）).N means the number of samples in this CF.LS is the sum of features among this samples. SS is the square sum of features among this samples.

Cluster Feature vector is

文本

描述已自动生成

图形用户界面, 文本

描述已自动生成

centroid radius diameter of cluster

branching factor B : Each nonleaf node contains at most B entries of the form [Cfi, childi],where i = 1,2,...,B, childi is a pointer to its i-th child node.

threshold T : cluster diameter(or radius) should be less than T.

The CF Tree will dynamically update when new data come in.

文本, 信件

描述已自动生成

文本, 信件

描述已自动生成

workflow:

<https://zhuanlan.zhihu.com/p/29848130>

## Partitioning method

### PAM(partition around medoids) CLARAN CLARANS

k-mediods

这三个可以放在一起讲。对于n个data和k个簇

首先需要知道一个图(graph abstration)。图上总共会有个点，每一个点都是一个k-medoids 的集合(set)。

neighbour: 两个集合有且仅有一个元素不同，则互为neighbour，每个点有k(n-k)个neighbour 。

PAM: 从任意一个点P开始。

2遍历P所有的neighbour

3找到最优的那一个X，

4若X的cost对比P有减少

进入X重复 2-4

否则返回P

PAM也只能保证local optimal.

CLARA是从原来的n个data中随机选取一部分(40+2k, k就是簇的数量).对于新的数据,会有新的graph abstraction.我们在新的图上套用PAM算法。

这样的过程总共进行m次，最终返回的是这m次的最优解。CLARA找到的是样本子集的local optimal，它的有效性和选取的子集有很大的关联。

CLARANS和CLARA不太一样，他的搜索是在原始图（和PAM一样的图）进行的，只是他不再遍历所有的邻居，而是当发现一个比当前更优的解的时候就直接进入那个点开始下一轮循环，所以他连local minimum都不能保证，但这也提供了跳出local minimum的可能性，根据论文CLARANS的效果要比CLARA好一些。

DBSCAN

Knowledge Discovery Process概念：

图示, 示意图

描述已自动生成

Data is dirty: incomplete

noisy

inconsistent eg: Age=42 Birthday = 03/07/1980.

图示

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

文本

描述已自动生成

图形用户界面, 文本

描述已自动生成

## 距离计算：

距离的定义

Distances are normally used to measure the similarity or dissimilarity between two data objects

距离属性

d(i,j) >=0; d(i,i)=0 ; d(i,j) = d(j,i); d(i,j) <= d(i,k) + d(k,j)

非负 同一点距离为0 对称symmetric 三角形法则

数据类型

文本

描述已自动生成

文本, 信件

描述已自动生成

距离计算为欧氏距离（Euclidean）或曼哈顿距离（Manhattan）

图形用户界面, 文本, 应用程序

描述已自动生成

表格

描述已自动生成

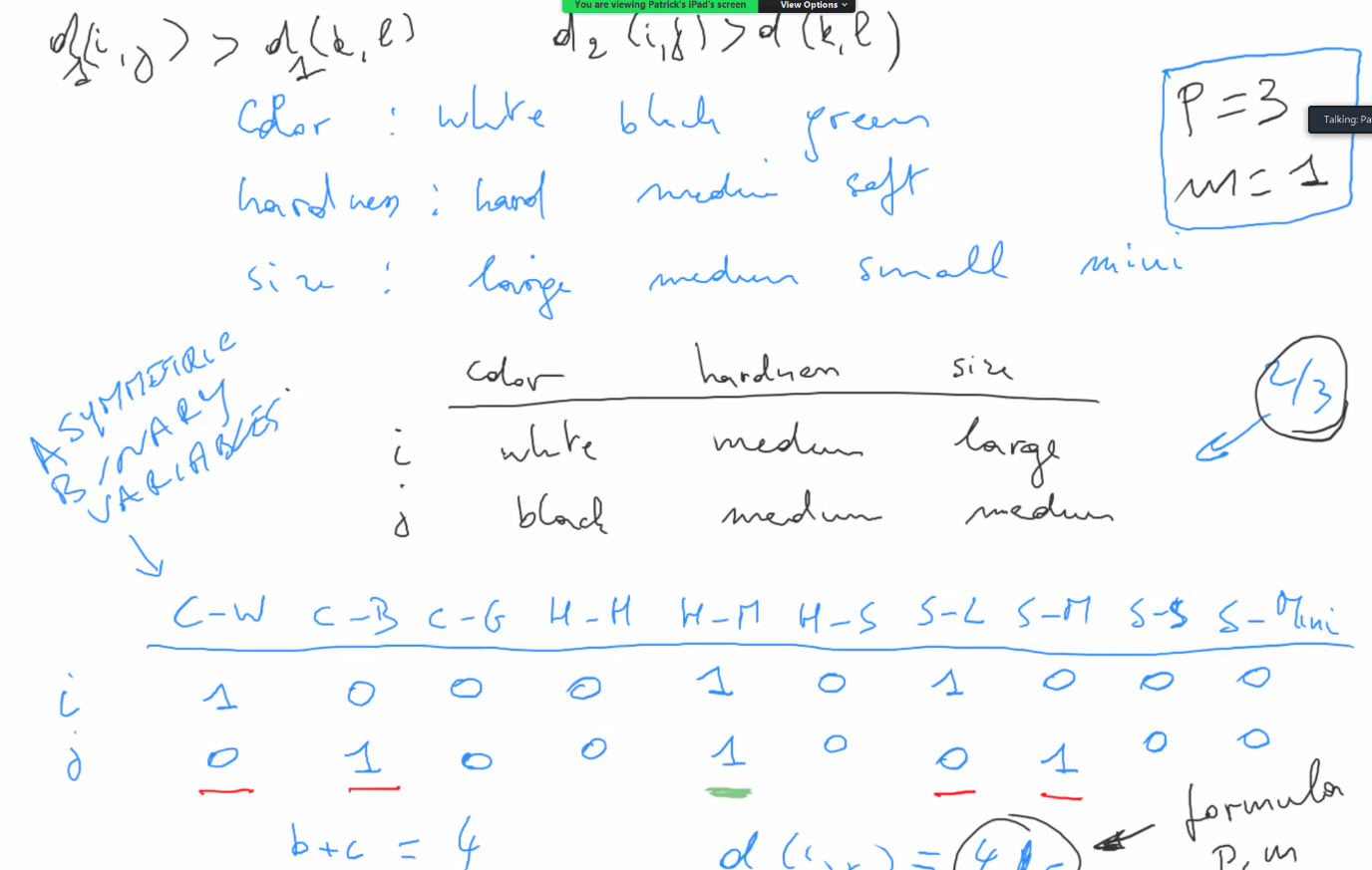
如果是非对称的那就没有d，即不考虑两个都为0的情况

Jaccard coefficient = 1-d(i,j) =

文本

描述已自动生成

这个过程像创建哑变量，在新生成的矩阵中同样定义p,m



d(i,j) =

文本, 应用程序

描述已自动生成

日历

描述已自动生成

## Aprior

首先有一个先前就确定好的minimum support 和minimum confidence

support 含有A的项P(X)

confidence 基于A条件概率 P(Y|X)

表格

描述已自动生成

以这个表为例 先计算A ->D

同时含有AD的项为3项，所以

support = 3/5=0.6

confidence是support除以A的概率，因为A的概率也是3/5所以这里confidence = 1

再来看D->A

D 的support是0.8，而confidence就是 0.6/0.8 = 0.75

接下来定义frequent/large itemset

意思是满足最小support要求的itemlist。像在上面的例子中 a b c d e f分别是不同的item，然后对于minimum support = 0.5的时候， {A,D}就是一个满足调节的frequent itemset，因为{A，D}的support是0.6超过了0.5.

通过两个步骤来找到关联：

1 找到所有的frequent itemset 使用apriori或者FP grow algorithm

2 Generate all rules with minimum confifenci from the frequent itemsets.

文本

描述已自动生成

所有frequent 的子集都是frequent的，这个很好理解（土一点讲，如果ABC在一起都能出现足够的次数，那么分开看AB,BC,AC出现的次数肯定不会小于ABC一起出现的次数）

superset就是超集，说白了就是如果A是B的superset 那么B属于A，如果B真属于A，那么A就是B的真超集。所有不频繁的itemset的超集都是不频繁的，这个也很好理解。

文本, 信件

描述已自动生成

第一步先找到单个item构成的 ‘large itemset’ 或者称为 ‘frequent itemset’

表格

描述已自动生成

如上图所示，总共Tid有4个，假设我们要求minsup = 2，那么因为D的sup只有1所以他被舍弃了。也就是说L1里面只有A B C E四个东西。

接下来的循环从2开始，一直到无法再创建新的frequent itemset为止。首先先取得上一步的结果，并用Apriori-gen生成一个Ck。Ck的产生如下图

文本, 信件

描述已自动生成输入是frequent itemset。需要注意的是，每一个L实际上一个由集合（里面有k-1个item）组成的集合。接下来其实就是从上一个大集合里选出所有可能的小集合的pair，其中每个pair里的小集合满足他们只有一个元素不相同。这样把这些pair分别取并集我们可以得到新的大集合，这个大集合里的每一个元素是由pair生成的小集合（长度为K）组成的，也就是算法中第4行循环结束以后的Ck。随后对于Ck中的每一个小集合c进行校验。对于c中的任意一个k-1子集s，如果这个子集s不在Lk-1中，那么说明这个子集就是一个infrequent itemset，所以这个子集所在的c也就是infrequent itemset（性质2），然后将这个c移除。全部完成之后就能返回Ck。此时的Ck就是一个Lk-1的超集！！！！

随后我们返回原算法：

文本

描述已自动生成

对于原先数据集中的D进行逐行遍历，同时对于Ck中的每一个小子集c进行遍历，计算小子集c在D中出现过的次数。如果出现过的次数大于minsup，那么这个c就是一个合法的c，将其加入Lk，最后返回Lk。这时候再看下面这张图应该是很好理解的。

手机屏幕的截图

描述已自动生成

## Rule generation algorithm

对于由frequent itemset，我们可以去挖掘他的关联规则

文本

描述已自动生成

这里的X -> L/X是 X推出L减去X。底下的意思是如果 X不能满足 X->L/X,那么X的任何一个子集X‘也不能而且得出的conf只会更小。

换成数学语言就是那句话下面的不等式，理解起来也很简单。X比X‘有更多的项他的support 肯定不会比X’来的大，所以sup（X‘）肯定大于sup（X）。

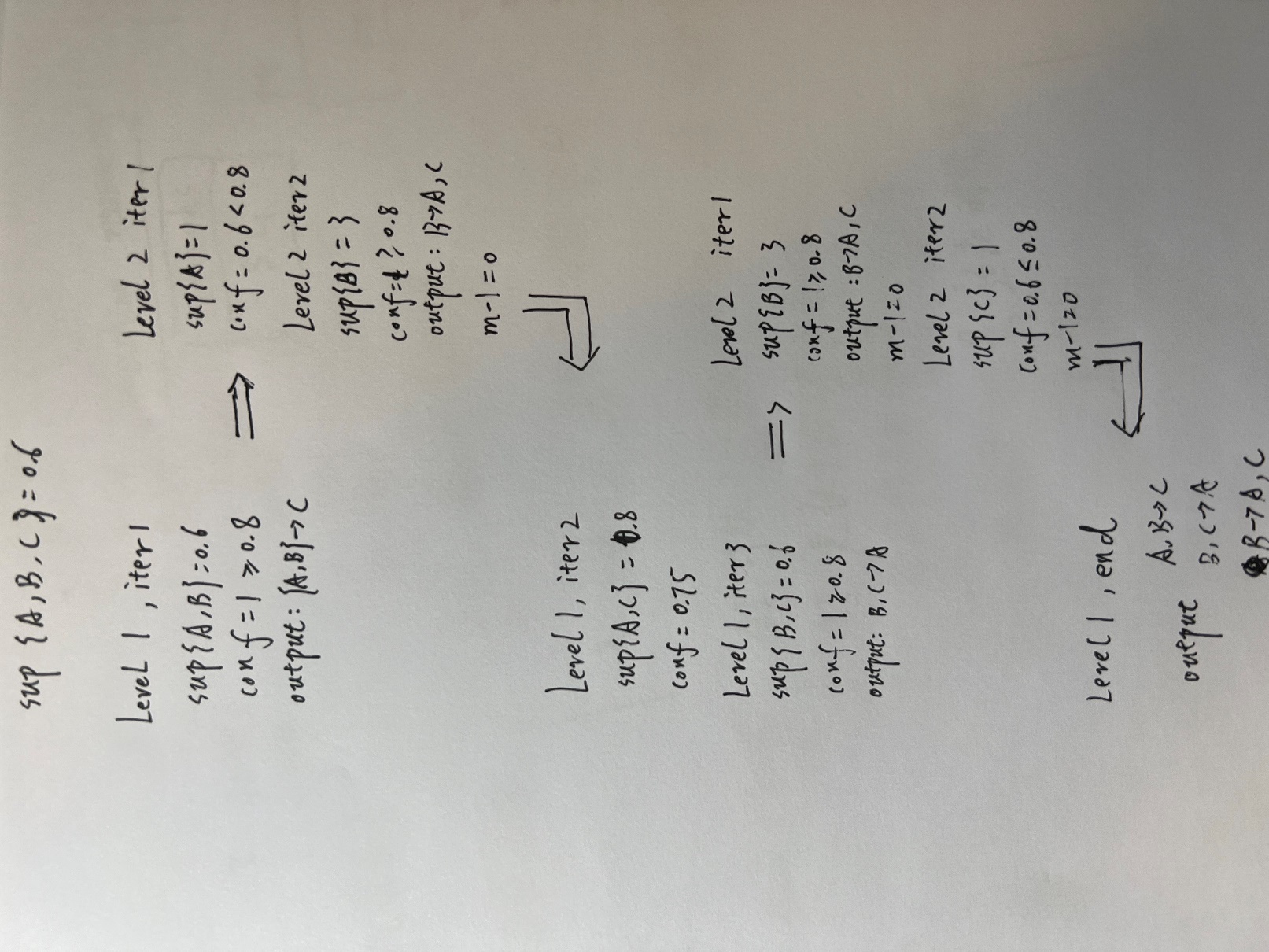
图形用户界面, 文本, 应用程序

描述已自动生成

算法是递归地进行的。对于要计算的lk，函数的开始输入是(lk,lk,conf)。对于每一次进入递归的am，取出所有m-1子集并进行循环，对于每一个iter计算conf，如果大于等于要求的conf就输出这个规则，然后再将这个iter中的m-1子集带入到下一个循环中计算。因为我们之前已经提到过如果A不满足那么A的子集一定不满足，所以当A的conf小于minconf的时候就没有必要对这个分支深究下去。

运算过程如下（例题手绘）

图形用户界面, 文本, 应用程序

描述已自动生成

FP grow

表格

描述已自动生成

图表, 雷达图

描述已自动生成首先在原列表里删掉infrequent的单项，每一行里再按照sup降序排列，如上图所示，得到新表。

**根据左表，按照每一个id循环，填充出右边这样的树。根据这个树得到**表格

描述已自动生成**Conditional database**

文本

中度可信度描述已自动生成

**后续参加ppt 2**

## Aprior算法背景下的constraints

**如果是monotone， 也就是大推小（非小推非大）。所以此时如果某个小集合不满足，就不需要再去考虑他接下来的情况。**

表格

描述已自动生成

**如果是antimonotone，也就是小推大，此时不能和上面一样，因为 这时候是 非大推非小， 小集合不满足条件不代表大集合不满足。所以只需要按照正常aprior算法进行，在输出的术后，如果满足条件就输出，否则就不输出。**

**FP 下的constraints**