基于自动驾驶事故数据的根因分析

10215501403 沈桐乐

目录

- 研究背景和数据来源
- 数据预处理
- 数据统计展示
- **根因分析**

1. 研究背景和数据来源

Backgrounds & Data Sources

研究背景

- 近些年来，购买和驾驶新能源汽车逐渐成为一种趋势，研究自动驾驶领域也是一个非常热门的方向
- 然而，由于技术的不成熟等种种原因，事故频发的自动驾驶系统让人们逐渐担忧
- 本次实验是站在一个数据学生的视角，通过对于数据的研究来浅探事故背后的原因

数据来源

- NHTSA（美国国家公路交通安全管理局）

← LAWS & REGULATIONS

Standing General Order on Crash Reporting

For incidents involving ADS and Level 2 ADAS

2. 数据预处理

Backgrounds & Data Sources

数据预处理

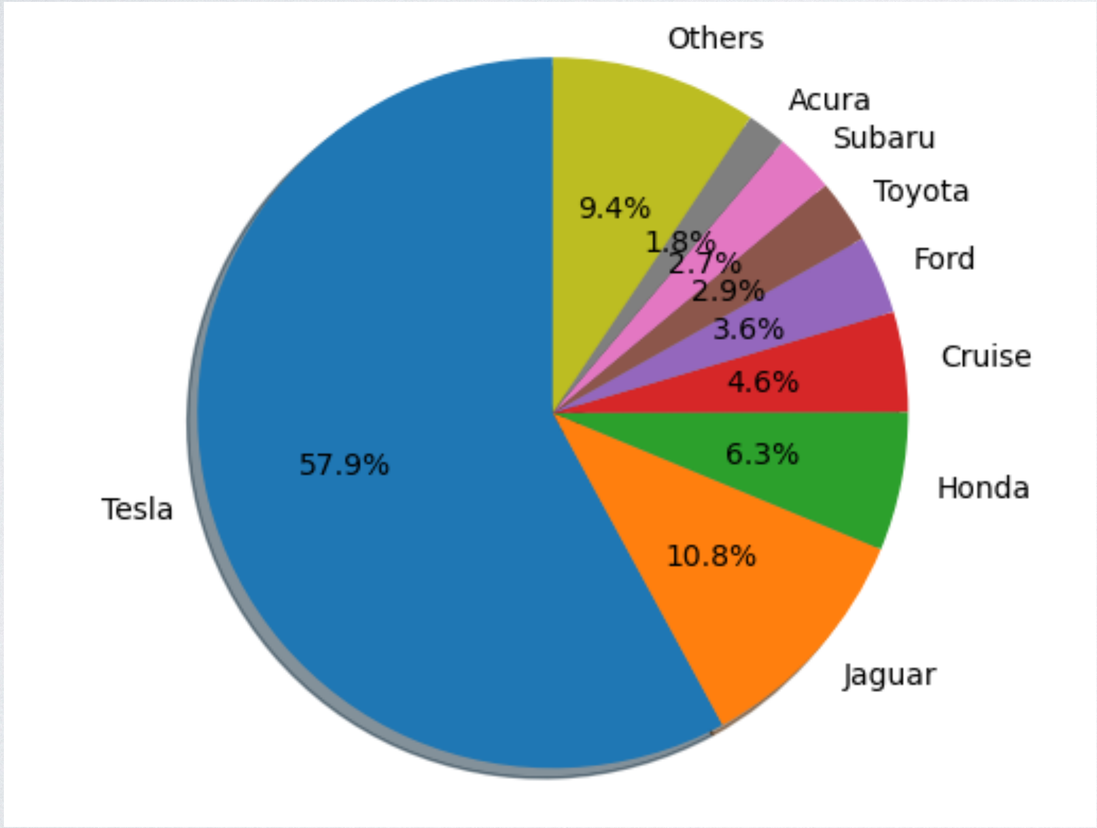
- 数据清洗，主要包括无关数据剔除，清洗无效数据，非数字变成数字，以上主要依赖pandas库

- 数据划分，把强相关的类型放在一组当中，比如天气情况的矩阵

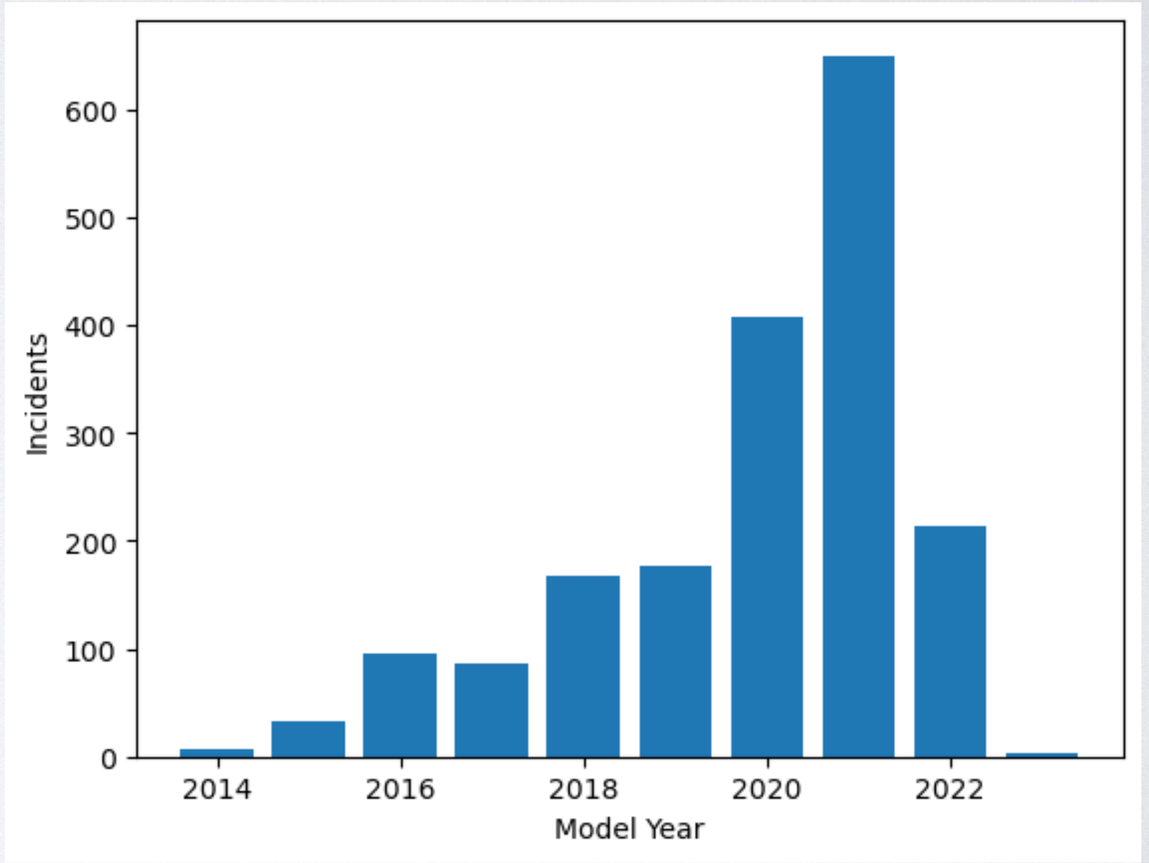
3. 数据统计展示

Display

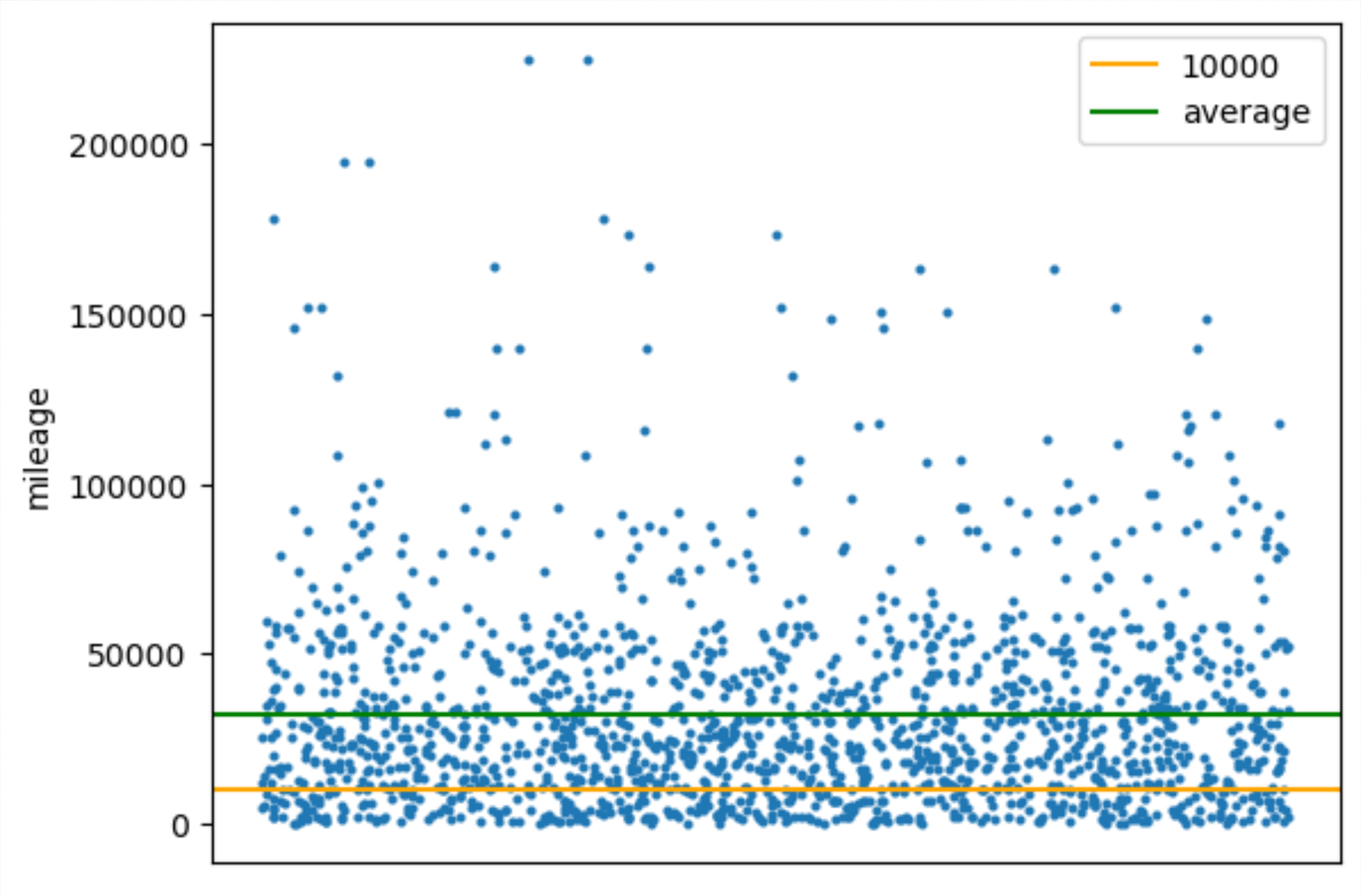
制造厂商

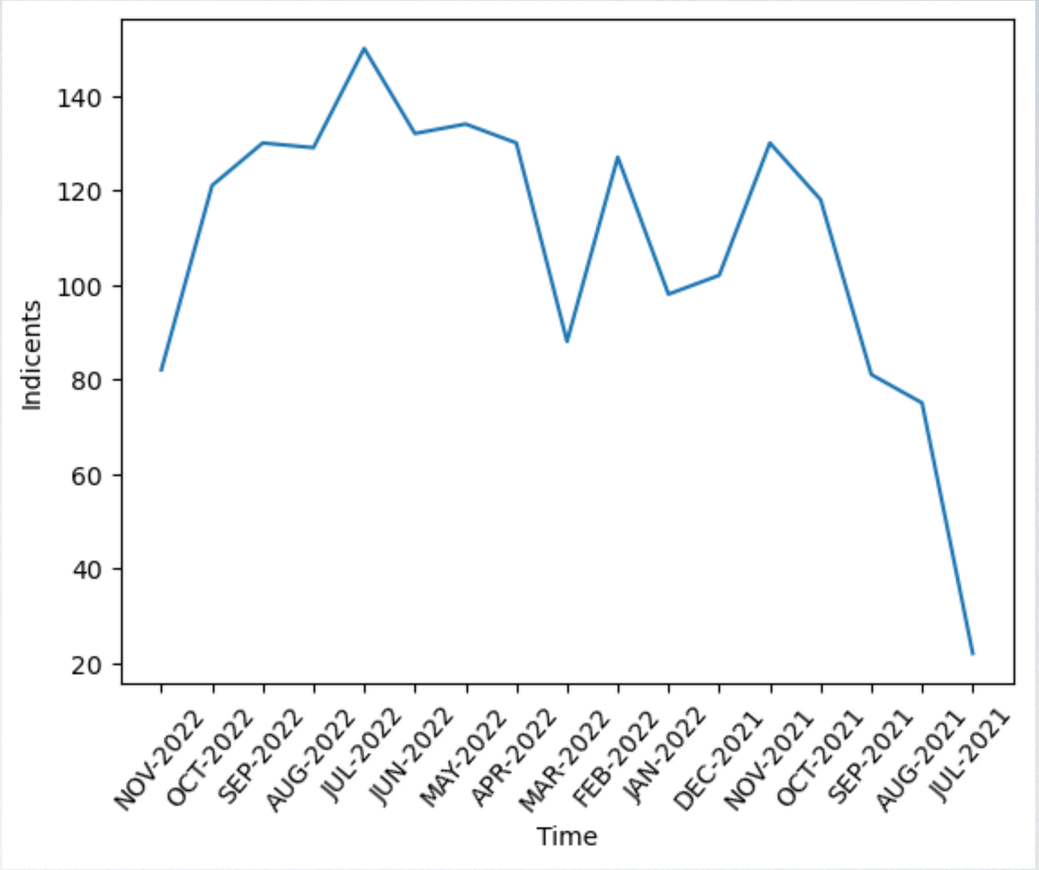

Others
Acura
Subaru
Toyota
Ford
Cruise
Honda
Jaguar
Tesla
9.4%
1.8%
2.7%
2.9%
3.6%
4.6%
6.3%
10.8%
57.9%

模型年份


Incidents
0
100
200
300
400
500
600
2014
2016
2018
2020
2022
Model Year

里程数


10000
average
200000
150000
100000
50000
0
mileage


140
120
100
80
60
40
20
Indicents
NOV-2022
OCT-2022
SEP-2022
AUG-2022
JUL-2022
JUN-2022
MAY-2022
APR-2022
MAR-2022
FEB-2022
JAN-2022
DEC-2021
NOV-2021
OCT-2021
SEP-2021
AUG-2021
JUL-2021
Time

报错月份

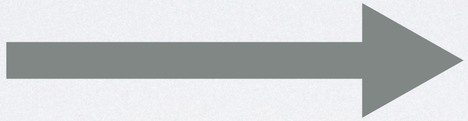
4.数据处理和根因分析

Data Process & RCA

根因分析（RCA）

- Root Cause Analysis，主要用于定位问题的根本原因，是近些年来自动化领域兴起的一种研究方向
- 在本次实验中使用主要是为了研究，哪些因素的**组合**最有可能导致事故的发生

Input = matrix(factors)



Output:
Rank(Combination(factors))

数据处理：降维

- RCA本质是一个NP问题，比如一共有n个影响因素，则每个因素有/没有就会产生至少2^n种可能的结果，随n呈指数关系
- 想要尽可能地加速算法，在计算之前，我们就需要想办法让每个事故所带的因子的信息量尽可能减少，与此同时尽可能地保留信息，也就是聚类降维
- 经过预处理一共有13个影响因素

降维前因子

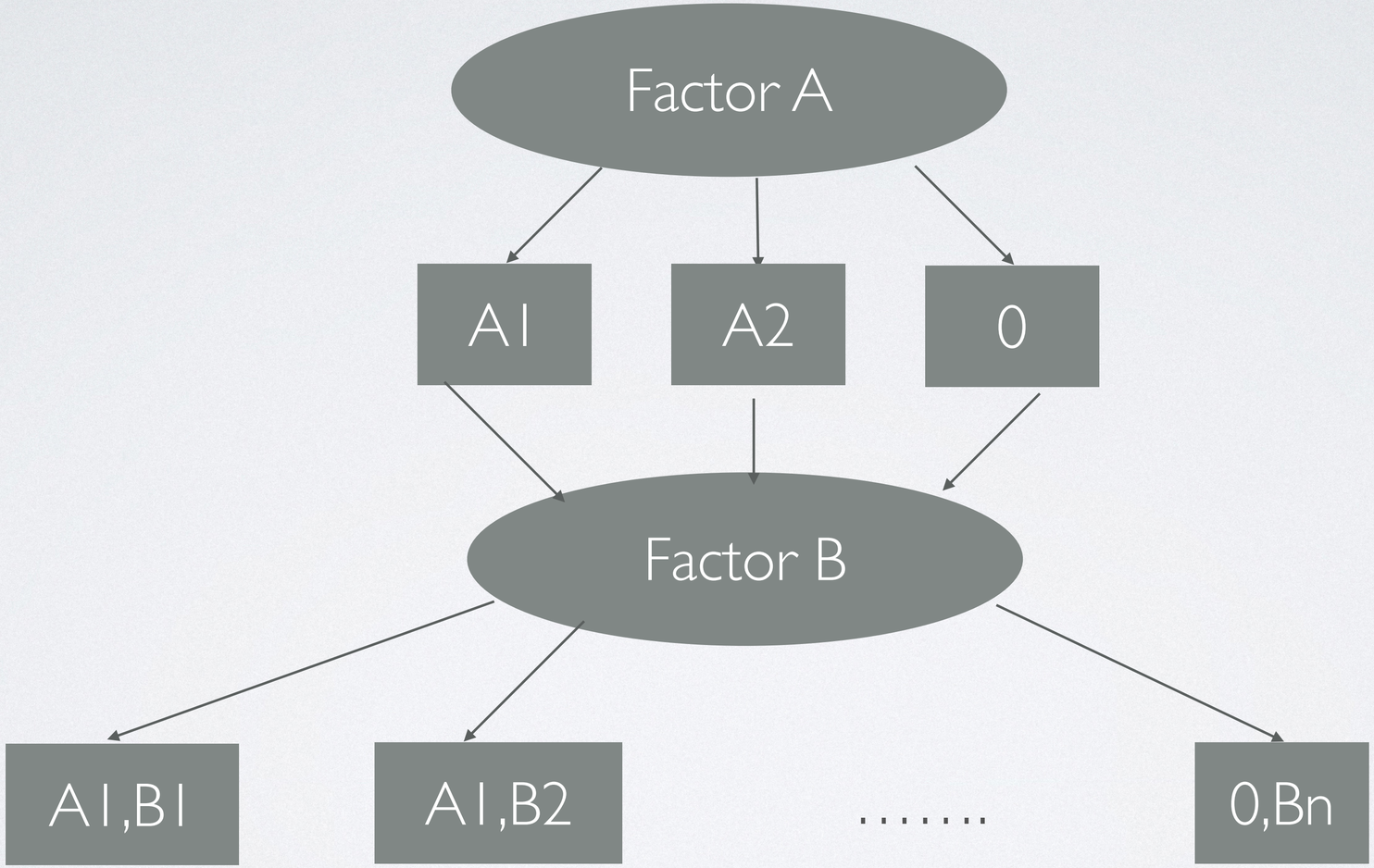
- 里程数(inf)
- 报错时间(inf)
- 限速情况(inf)
- 撞车速度(inf)
- 行驶路况(5)
- 行驶路面(10)
- 光照情况(6)
- 天气情况(2*9)
- 撞车对象(12)
- 撞车之前的自动操作(12)
- 实际损坏区域(10*2)
- 区域压力值(10*2)
- 气囊释放情况(2)

降维后因子（降维方法）

- 超速(2，大小判断)
- 路面干湿(2，分类成干和下雨等不干的情况)
- 气囊打开情况(2)
- 传感器误差(2，通过矩阵1范数的距离分类，参数为距离阈值)
- 天气情况(2，通过加权和矩阵乘法，参数为权重向量)
- 光照情况(3 光照良好/一般/很差, 聚合光照和时间，赋予权重)
- 速度情况(3 ,简单数值分类，参数为超速值)
- 撞车对象(3, 大/小/其他，根据对象体型分类)
- 里程数(3 ,简单数值分类，参数为汽车寿命阈值)
- 自动移动操作(3,直行/非直行/静止)
- 路种类(4,高速/普通路面/复杂路段/其他)

哪怕经过降维处理，在这种情况下也需要$3^5 \times 4^5 \times 5 = 1244160$种组合，数据处理已经做到极限了，必须要优化算法

搜索树和剪枝

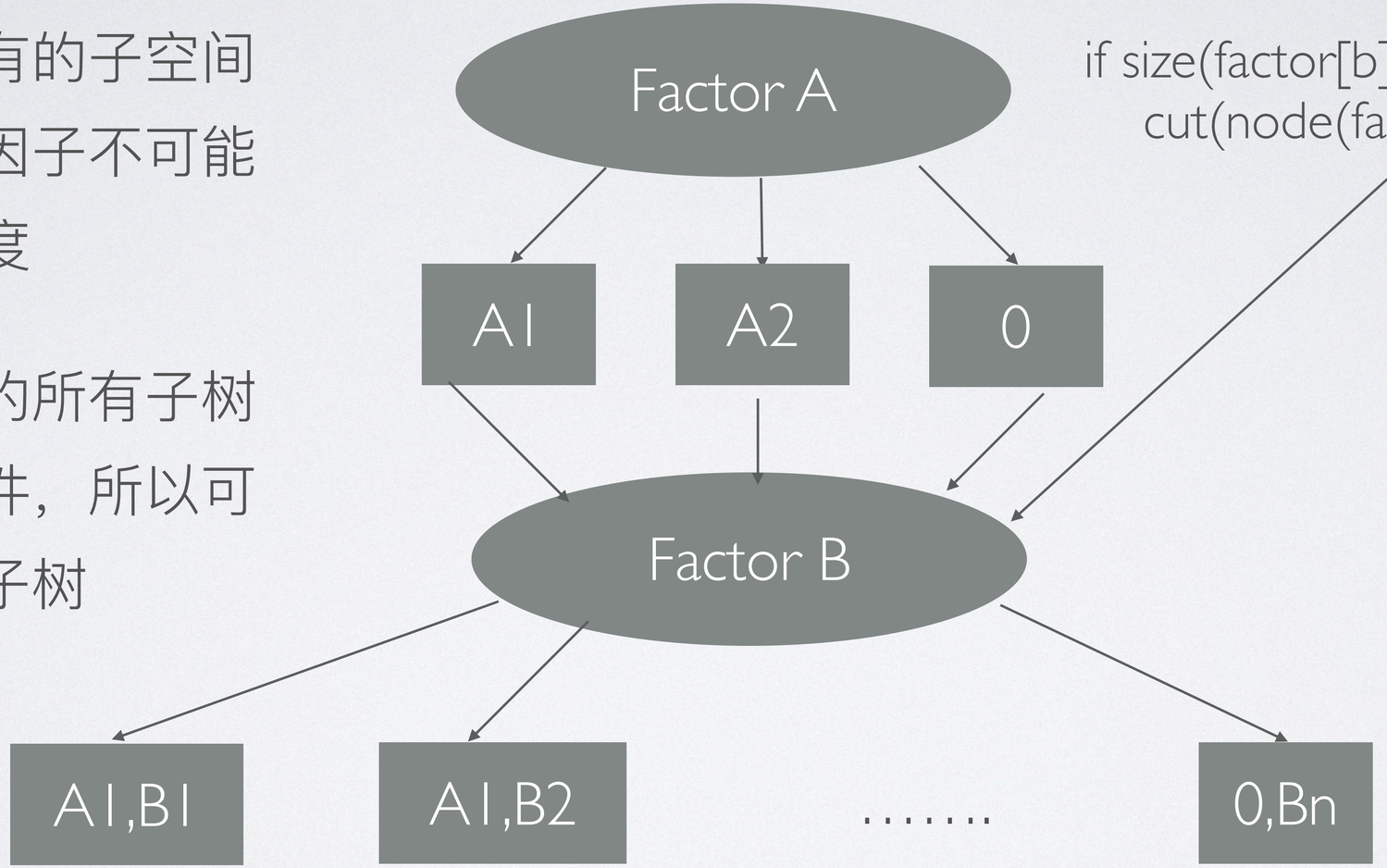


优化和剪枝算法：排序

- 把因子的维数从小到大排序，由于剪枝的可能性存在，这样做可以让接近根节点的地方分支尽量少
- **sort**(*factors*, ascending = *True*)

基于影响的剪枝

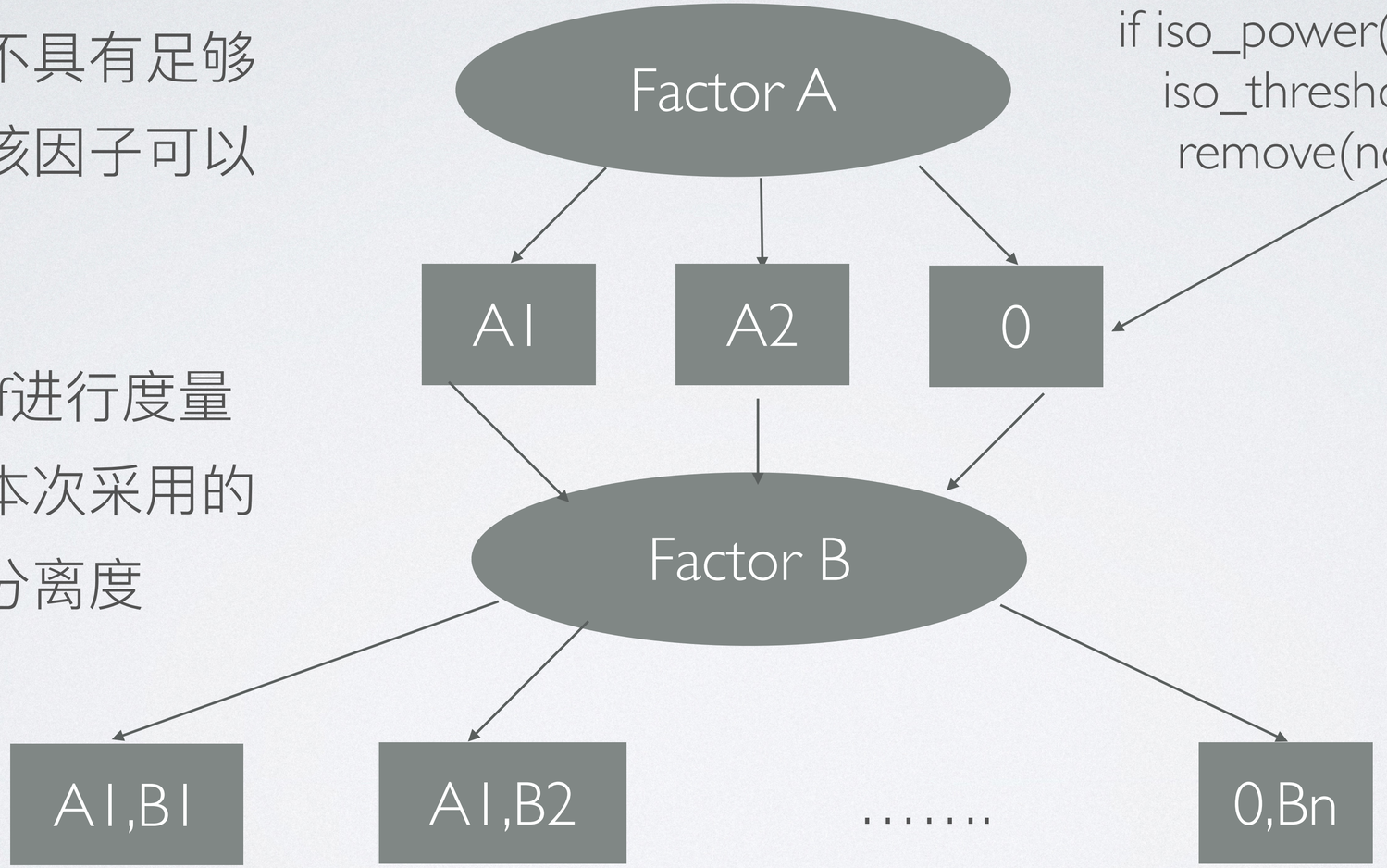
- 如果因子所占有的子空间过小，那么该因子不可能对数量有区分度
- 同理，该因子的所有子树也满足这个条件，所以可以直接剪去因子树




```
if size(factor[b].space) < size_threshold:
    cut(node(factor[b]))
```

基于分离度的剪枝

- 如果因子分布不具有足够的分离度，则该因子可以忽略
- 分离度是对p.d.f进行度量的一种方式，本次采用的是信息熵作为分离度



if iso_power(distribute(a1,…,an)) <
iso_threshold:
remove(node(a1,…,an))

结果证明，两种剪枝方法的组合是非常有效的，最后的组合数降到了15189，剪枝率达到了98.78%！

结果排序

- 当搜索树达到了最底层的时候，需要把组合放置到一个集合当中排序，而排序的依据就是聚类结果的显著程度，本次实验采用了KL散度作为距离的度量，再通过归一化方法得到好看的结果

```
Valid result:  15189
Best result:  ([1, -1, 1, 1, 1, 1, 1, 1, 1, 0, 3], 28.686613826612206)
```

4.总结和改进

Conclusion & Improvement

总结

- 本次实验主要思考和复现了自己的RCA算法，经历了一个数据处理的全流程

- 对于RCA算法还有很多的改进空间，比如粒度分析，评估函数，聚类方法等等，也是当前世界上RCA技术的主要难点所在

- 由于大部分数据基于公开数据集，有一些关键数据被厂商保留，但是该算法是有很强大的商用前景的，数据集本身空间的增加能让结果更加好看

On October [XXX], 2022, Daimler Truck was notified of an incident involving a 2022 Freightliner Cascadia traveling [XX
[REDACTED, MAY CONTAIN CONFIDENTIAL BUSINESS INFORMATION]
[REDACTED, MAY CONTAIN CONFIDENTIAL BUSINESS INFORMATION]
[REDACTED, MAY CONTAIN CONFIDENTIAL BUSINESS INFORMATION]
[REDACTED, MAY CONTAIN CONFIDENTIAL BUSINESS INFORMATION]

参考算法

- iDice算法，[ICSE'16],微软亚洲研究院
- Adtributor, ICSE'16& Recursive Adtributor
- HotSpot, IEEE ACCESS 18
- 如果有兴趣的话可以看一下这几篇算法论文，思想非常巧妙！

THANKS!!