# 数据隐私实验报告

魏钊PB18111699

## 一．Samarati算法

## 1．问题描述

使用 Samarati（基于二分法） 算法完成 k 匿名。

要求：

QI={age, gender, race, marital \_status} (categorical 型), S = {occupation}

输入： data, k, maxSup (data 是数据集， k 是 K-Anonymity 的参数， maxSup 表示最大 suppression 的个数)；

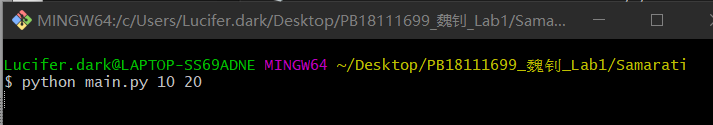
输出： 匿名后的数据集。

评价指标：运行时间和LM

## 2．程序使用指南以及程序分析

在Saramati文件夹下在终端执行main.py文件。

例如：K=10,Maxsup=20



注意：程序调用了pandas

程序的输出为Samarati文件下的Final.txt(泛化后的数据)，和终端中打印的三行数据。

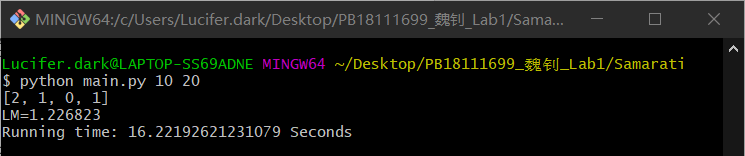
三行数据分别为：

划分向量[age,Gender,race, Marital\_status] 注：0为不泛化输出原始数据

LM

运行时间

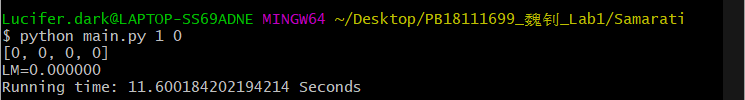
例如：



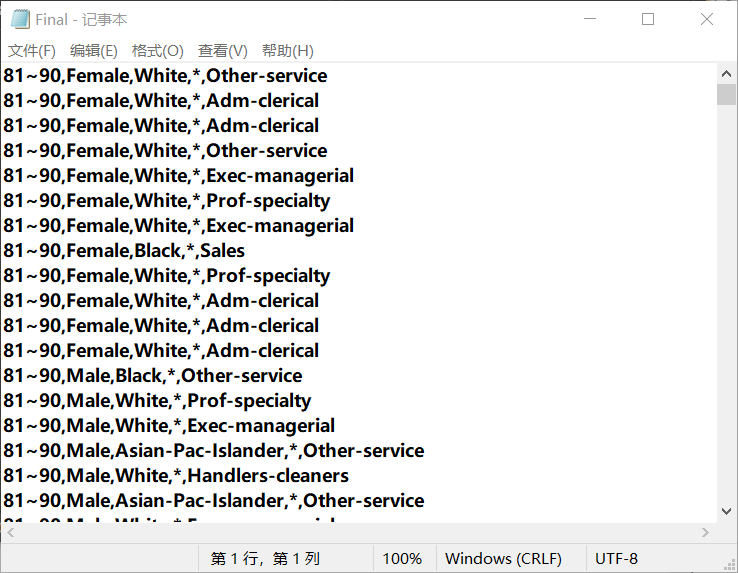
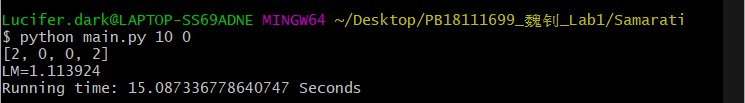
相关的程序分析在代码注释中进行了详细的叙述。

## 3．实验结果分析

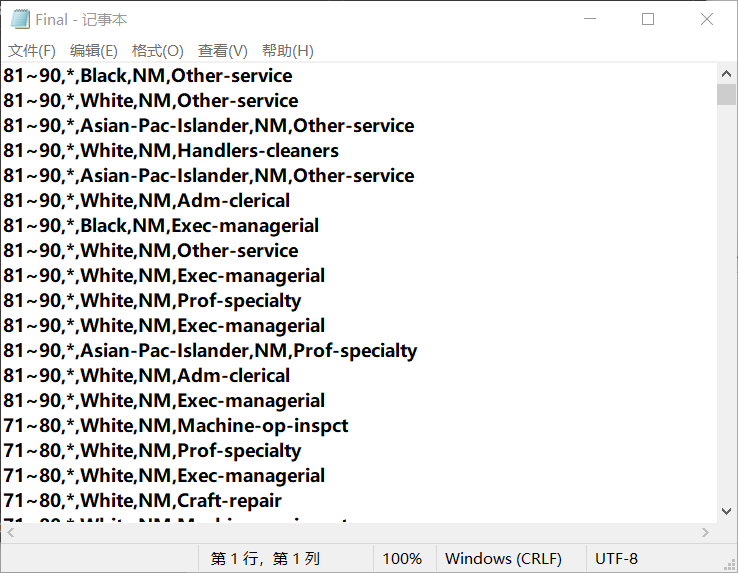
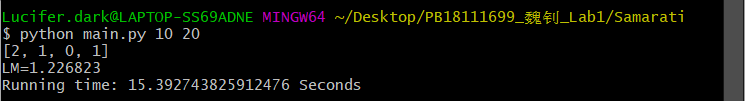
K=1,maxSup=0时，数据不泛化，即为原始数据。



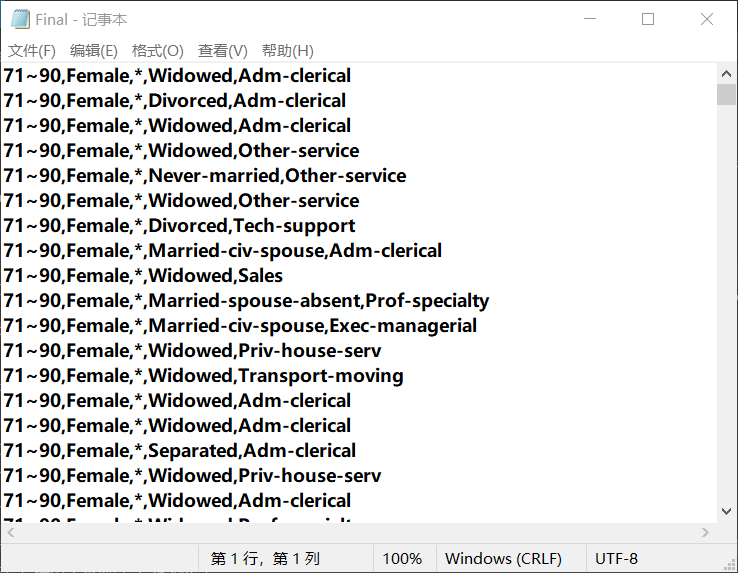
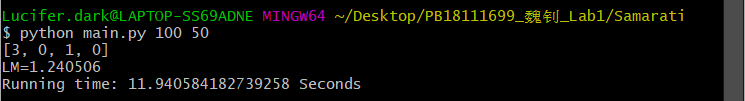
K=10,maxsup=0时：



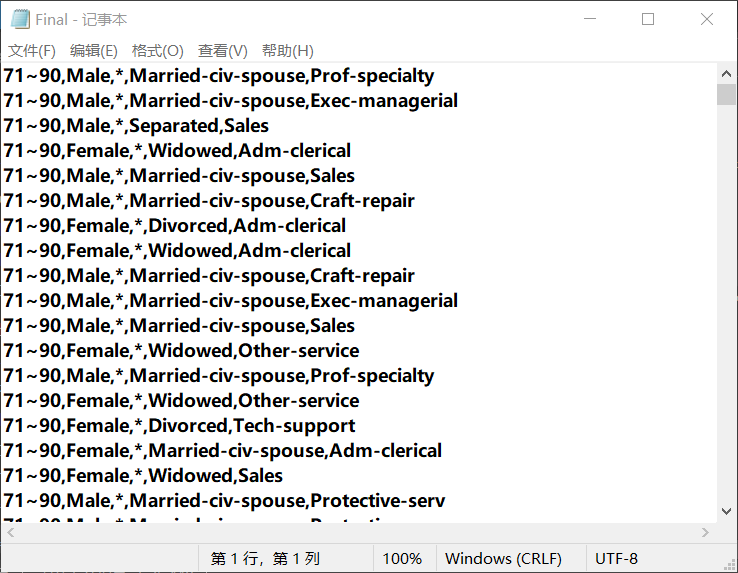
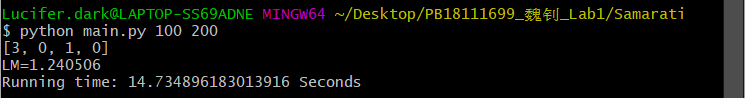
K=10，maxSup=20时：



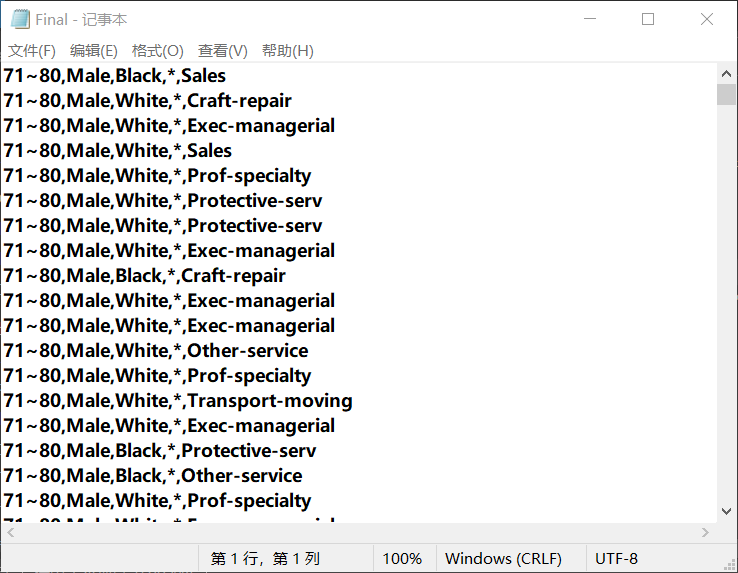
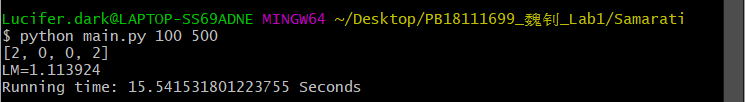
K=100，maxsup=50时：



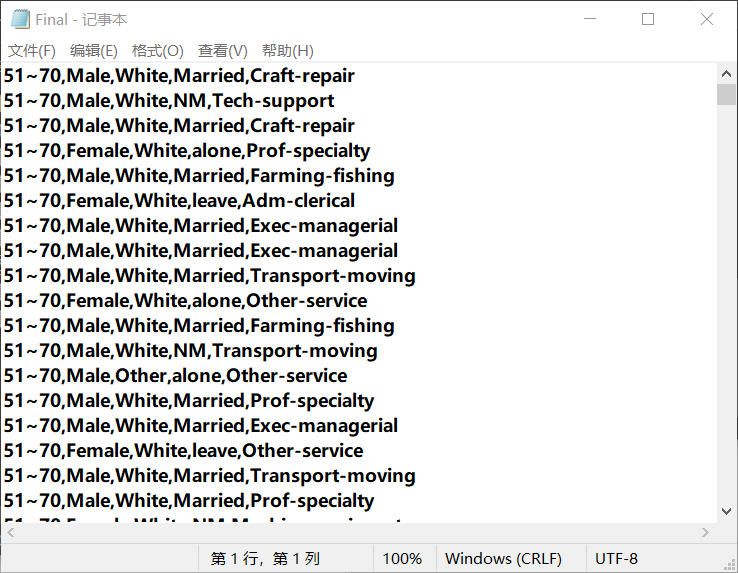
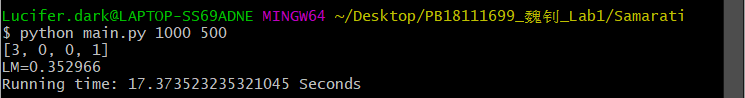
K=100，maxsup=200时：



K=100，maxsup=500时：



K=1000，maxsup=500时：



观察上面的测试结果可知：

当 k 不变,maxsup变大时，运行时间变小，LM 变小（ 因为可以通过不发布数据来减小泛化深度， 因为 maxsup 变大到一定程度， 对于不同泛化深度尝试次数减少， 运行时间会减少）。

## 4， 讨论与总结

不同的k与maxsup对算法输出和性能有影响。同时算法理论上来说可以得到很多解满足要求。 而具体的运行时间在不同测试环境下均有一定偏差。

## 二．Mondrian算法

## 1．问题描述

使用 Mondrain 算法完成 k 匿名。

要求：

QI={age, education\_num}（数值型） , S = {occupation};

输入： data, k, maxSup (data 是数据集， k 是 K-Anonymity 的参数， maxSup 表示最大 suppression 的个数)；

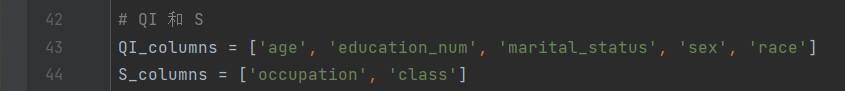
输出： 匿名后的数据集。

评价指标：运行时间和LM

## 2．程序使用指南以及程序分析

注：本程序也实现了附加的Mondrian算法实现非数值型的泛化

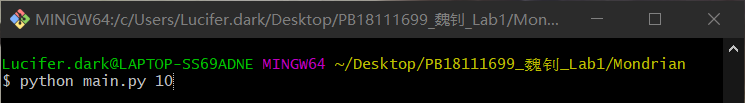
只需在修改如下代码QI\_colums处需要泛化的属性即可：



这里对age，education\_num,marital\_status,sex,race进行了泛化，S集为occupation,class

在Mondrian文件夹下在终端执行main.py文件。

例如：K=10



程序的输出为Mondrian文件下的Final.txt(泛化后的数据)，和终端中打印的两行数据。

两行数据分别为：

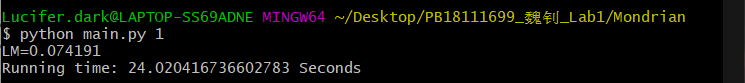
LM

运行时间

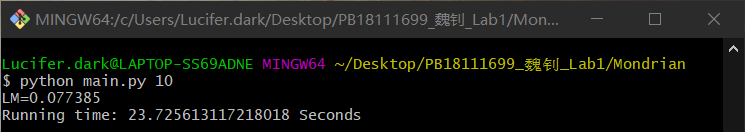
相关程序分析，在代码注释中有详细的描述。

## 3．实验结果分析

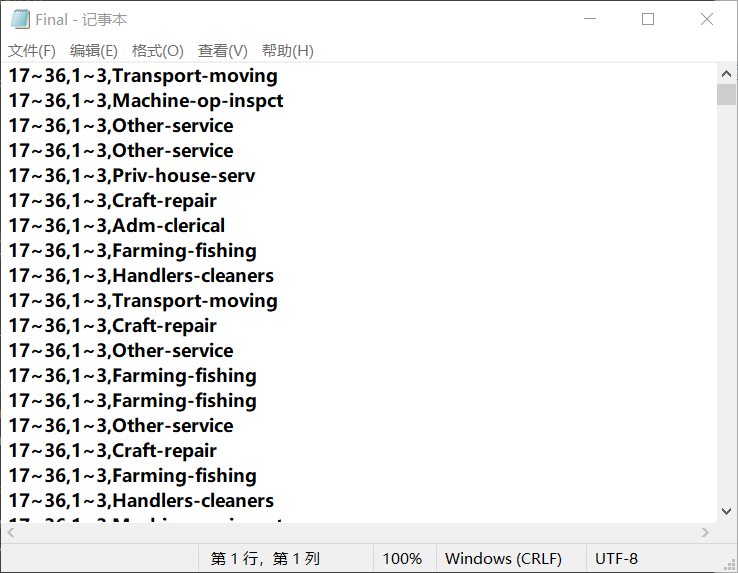
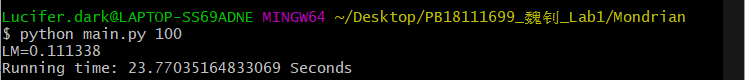
K=1时:



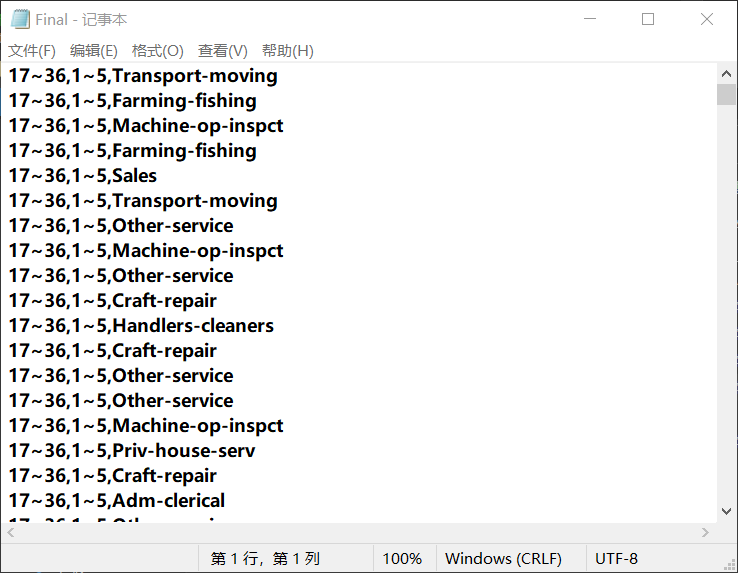
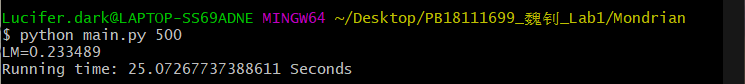
K=10时:



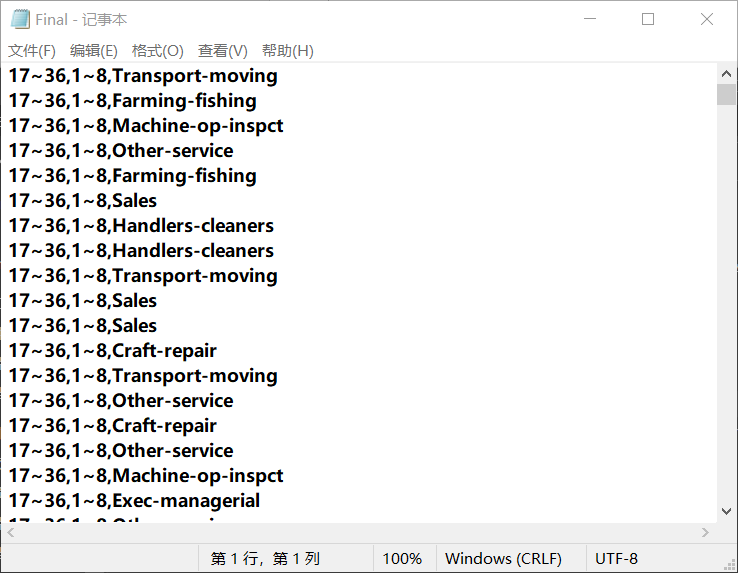
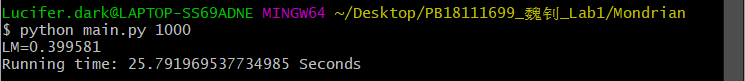
K=100时：



K=500时：



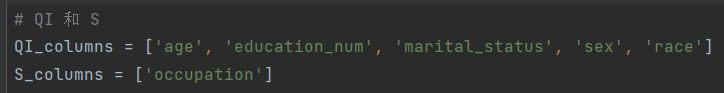
K=1000时：



可以看到当k增加时，LM 变大。因为k越大，二分的层数就越小，而 LM（loss 就越大。

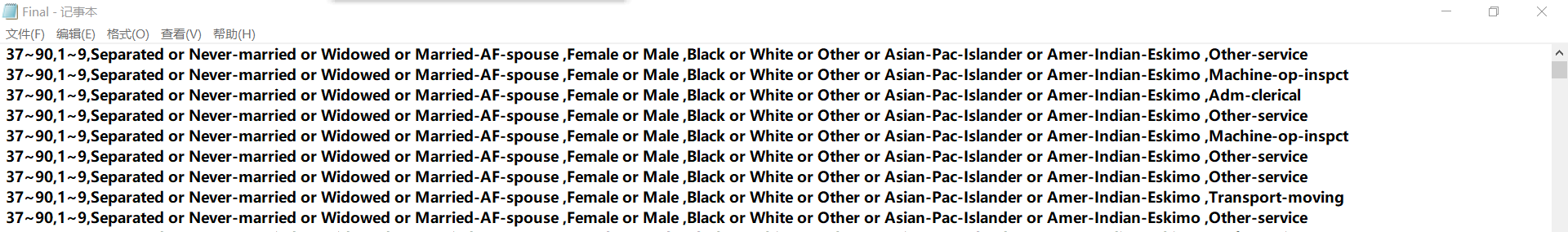
下面进行选做的测试：

QI和S集如下：



输出格式如下图所示。

K=1000时：



K=10时：



## 三．选做部分

1．Samarati算法可能会有很多解满足要求，调研并探究如何选择输出保证结果的可用性尽可能大，说说你的启发。

可以注意到在Samarati算法中当取得中间高度的向量组时，不同的枚举方式可以得到不同的向量组顺序，因而得到的第一个满足 k 匿名的向量组有所不同， 所以导致多解性。

通过对比不同枚举顺序导致的发布的数据的差别可以看出，当我们枚举向量组时， 可以优先枚举泛化层数小的属性， 这样可用性，隐私和 LM 可以得到一个更好的平衡， 数据可用性也更大。

抛开数据属性本身的泛化层数和属性范围来说， 可以根据实际情况以及不同属性的可用性角度来控制枚举的顺序。 比如：对于QI中可能发布后比较敏感的属性，可以最后枚举（这样泛化程度一般更高） 或者对于QI中可能一些需要精确发布的属性， 可以优先枚举。 从而可以通过控制枚举顺序来得到不同的发布数据，在满足同样的 k 匿名的前提下， 数据可用性变大。

2.Mondrian算法处理categorical，已经在Mondrian部分说明过了。