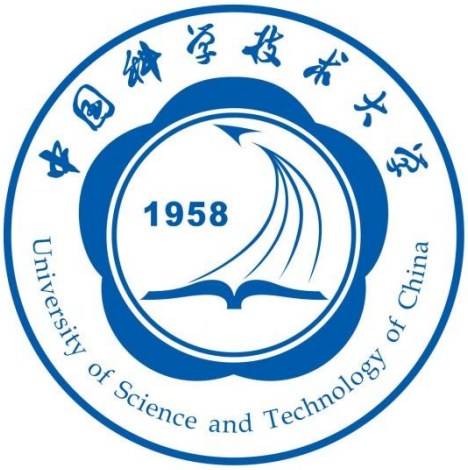
中国科学技术大学计算机学院
《数据隐私的方法伦理和实践》报告



实验题目：lab1_K-Anonymity

学生姓名：胡毅翔

学生学号：PB18000290

完成日期：2021 年 5 月 20 日

计算机实验教学中心制
2019 年 09 月

实验目的

1. 实现 `Samarati` 算法。
2. 实现 `Mondrian` 算法。
3. 测试不同的 `k` 和 `MaxSup` 对运行时间和 `LM` 的影响。
4. 调研并探究如何选择输出增加 `Samarati` 算法的可用性。
5. 增加 `Mondrian` 算法处理 `categorical` 类变量。

实验环境

1. PC一台
2. Windows操作系统
3. VSCode编辑器
4. Python 3.8.1

目录结构

> doc
>
> > report.pdf(本文档)
>
> src
>
> > samarati.py(Samarati算法主体)
> >
> > mondrian.py(Mondrian算法主体)
> >
> > adult.data(处理的数据集)
> >
> > adult.csv(data.py用，增加了属性行)
> >
> > adult_samarati.csv(Samarati算法的输出结果)
> >
> > adult_mondrian.data(Mondrian算法的输出结果)
> >
> > data.py(预处理)
> >
> > 其他
>
> README.txt

实验原理

Samarati算法

输入：

1. `k`：处理后的结果中每个相同的QI组合至少有k个元组。
2. `adult.csv`：待泛化的数据集
3. `MaxSup`：可以删去的不满足k匿名的元组数。
4. `T`：泛化的层次结构。
5. `Utility_evaluation`：可用性评估方法。

输出：

1. `Time`：算法运行时间。
2. `LM`：对应评估方法下的效用函数值。
3. `sol`：各个QI的泛化层次。
4. `adult_samarati.csv`：泛化的满足k匿名的数据集。

伪代码：

```
low = 0
high = height(T)
Best_LM = Inf
while low < high:
    mid = (low + high)/2
    flag = False
    for vec that sum(vec)=mid:
        data_samarati, reach, lm = satisfy(k,vec,MaxSup,Utility_evaluation)
        if reach && lm < Best_LM:
            data_anonymized = data_samarati
            sol = vec
            Best_LM = lm
            flag = True
    if flag == True:
        high = mid
    else:
        low = mid + 1
return data_anonymized, sol, Best_LM
```

注：`satisfy()` 函数用于测试在给定 `k`,`MaxSup` 和 `Utility_evaluation` 的情况下，泛化层次是否满足k匿名，返回泛化后的数据集，是否满足，效用函数值。

Mondrian算法

输入：

1. `k`：处理后的结果中每个相同的QI组合至少有k个元组。
2. `adult.data`：待泛化的数据集

输出：

1. `Time`：算法运行时间。
2. `LM`：Loss Metric。
3. `adult_mondrian.data`：泛化的满足k匿名的数据集。

伪代码：

```
Anonymize(Partition)
    if(no allowable multidimensional cut for Partition)
        return ϕ:Partition -> summary
    else
        dim <- choose_dimension()
        fs <- frequency_set(Partition,dim)
        splitVal <- find_median(fs)
        lhs <- {t∈Partition:t.dim ≤ splitVal}
        rhs <- {t∈Partition:t.dim > splitVal}
        return Anonymize(rhs)∪Anonymize(lhs)
```

注：

1. `choose_dimension()`函数用于选取被泛化程度最大的维度。
2. `frequency_set()`函数用于求取对应维度上的取值分布。
3. `find_median()`函数用于根据分布求取中位数。
4. 类别型属性先化为数值型进行处理，在输出过程中再化为类别型即可。

评估方法

LM(Loss Metric)

`LM`的计算方式如下：

- 对于数值型属性：

$$LM = \frac{U_i - L_i}{U - L}$$

- 对于类别型属性：

$$LM = \frac{|M| - 1}{|A| - 1}$$

  其中`|A|`为所有类别数，`|M|`为以该泛化节点为根节点的叶子节点数。

- 每个属性的`LM`为其所有泛化后元素的`LM`的平均值。
- 整体的`LM`为所有属性的`LM`之和。

`LM`考虑了泛化层次带来的信息损失，但认为不同属性的信息量是相同的（这点可以考虑在求和时，对不同属性的`LM`加权来调整，默认均为1），而且没有考虑被删去的元组的损失。

DM(Discernability Metric)

`DM`的计算方式如下：

- 对于未被删除的元组，其损失为泛化后`QI`与其相同的元组数。
- 对于被删除的元组，其损失为数据集的大小。
- 整体的损失为所有元组的损失之和。

  即：

$$DM = \sum_{\forall E \text{ s.t. } |E| \geq k} |E|^2 + \sum_{\forall E \text{ s.t. } |E| < k} |D||E|$$

`DM`考虑了泛化带来的损失（同一等价类的元组数越多，损失越大），还考虑了被删除元组带来的损失，不过缺乏对泛化层次的影响的度量。

non-uniform entropy

`non-uniform entropy`的计算方式如下：

- 被泛化的数据中，同一记录值的记录集合在一个属性`V`（`n`个类别其泛化到同一个值）上的损失为：

$$H_V^r = -\sum_{i=1}^{n} p_i' \log\left(p_i'\right)$$

其中:

$$p_i' = \frac{p_i}{\sum_{i=1}^{n} p_i}$$

$$p_i = \frac{\#(V=i)}{|D|}$$

`|D|`是同一记录值的集合大小，`#(V=i)`为该集合中属性`V`属于类别`i`的元素数。

- 假设各个变量相互独立，则对于`m`个属性泛化后的记录，泛化后记录值同为`l`的集合的损失为各个属性上的损失之和，即：

$$H_l^r = \sum_{j=1}^{m} H_{V_j}^r$$

泛化数据的总损失为所有泛化记录值的损失之和，即：

$$H_{total}^r = \sum_{l=1}^{L} H_l^r$$

- 对于被删除的数据，同一记录值的记录集合在一个属性上的损失为：

$$H_V^s = -\sum_{i=1}^{n} p_i' \log\left(p_i'\right)$$

其余定义同泛化数据，则有：

$$H_l^s = \sum_{j=1}^{m} H_{V_j}^s$$

$$H_{total}^s = \sum_{l=1}^{L'} H_l^s$$

- 对于整个数据集，总损失为泛化数据和删除数据的损失之和，即：

$$H_{total}^{r+s} = H_{total}^r + H_{total}^s$$

`non-uniform entropy`参考信息熵，考虑了泛化前后的数据集的信息量差异，差异越大说明泛化程度越大，泛化后的信息量越少。

实验过程

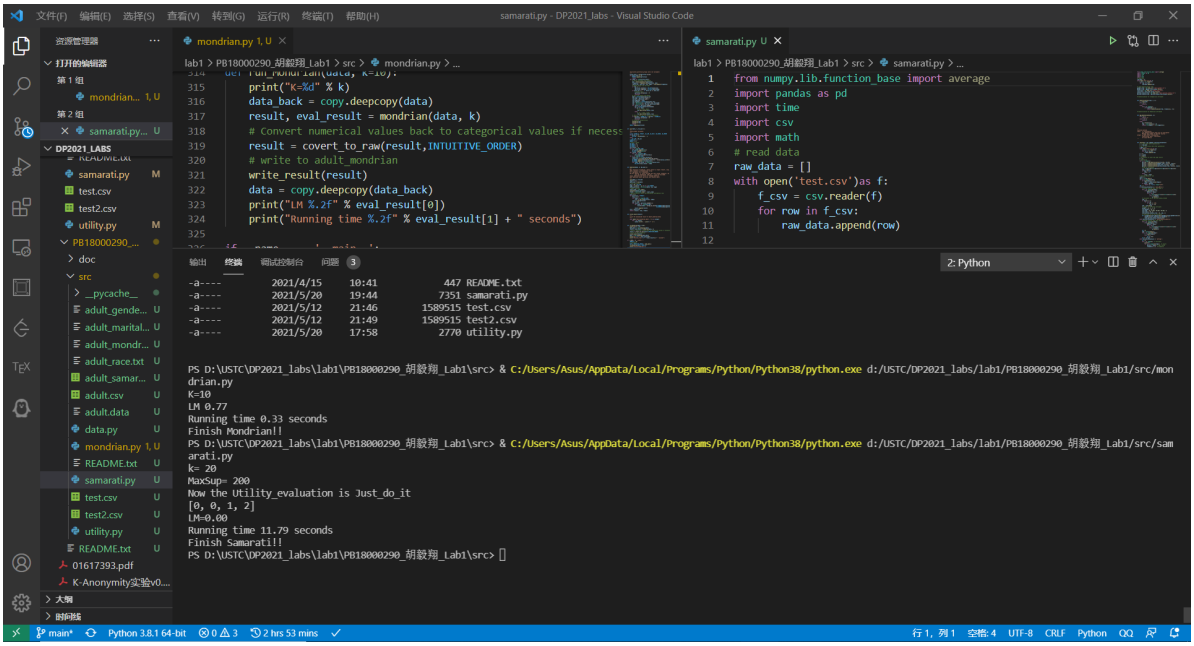
Samarati

1. 设置参数`k`，`MaxSup`。
2. 设置评估策略`Utility_evaluation`，其中`Just_do_it`为寻找泛化层次最低的输出，其他则为对应效用函数下的最优结果。
3. 执行，返回评估函数值（显示为`LM=xx`）（若使用`Just_do_it`，则效用函数固定为0），泛化层次向量，运行时间，结果保存至`adult_samarati.csv`。

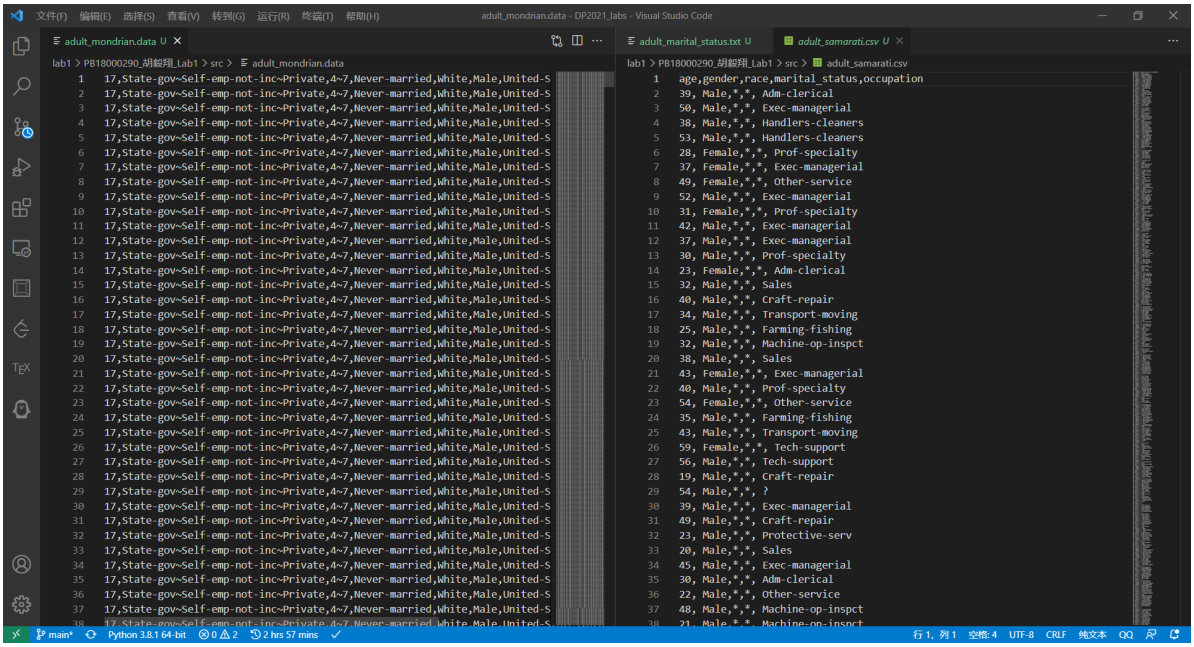
Mondrian

1. 设置参数`k`。
2. 执行，返回`LM`和运行时间，结果保存至`adult_mondrian.data`。

运行



输出结果



实验结果

Samarati（仅考虑height)

k	MaxSup	vec[age,gender,race,marital_status]	Time(s)
10	200	[1, 0, 1, 0]	18.65
10	100	[1, 0, 1, 1]	17.89
10	50	[3, 0, 1, 0]	14.06
5	200	[1, 0, 1, 0]	20.71
5	100	[1, 0, 1, 0]	16.89
5	50	[2, 0, 1, 0]	18.73
20	200	[1, 1, 1, 0]	20.31
20	100	[2, 1, 1, 0]	12.54
20	50	[1, 0, 1, 2]	11.16

Samarati（考虑不同的评估方法）

k	MaxSup	LM	DM	non-uniform entropy
10	200	1.05[1, 0, 1, 0]	39728101[1, 0, 1, 0]	91.69[1, 0, 1, 2]
10	100	1.16[1, 0, 1, 1]	14746387[0, 0, 1, 2]	91.69[1, 0, 1, 2]
20	200	2[0, 0, 1, 2]	19107497[0, 0, 1, 2]	91.69[1, 0, 1, 2]
30	200	2.05[1, 0, 1, 2]	27583737[0, 1, 1, 2]	91.69[1, 0, 1, 2]
20	50	2.05[1, 0, 1, 2]	23809399[0, 1, 1, 2]	91.69[1, 0, 1, 2]

由表中数据可以看出：

1. 在相同的`k`和`MaxSup`下，不同评价指标的输出结果（泛化层次向量）并不相同，比如`k=10`，`MaxSup=100`时。
2. 在不同评价指标下，效用最高时的`k`和`MaxSup`也不相同。
3. `non-uniform entropy`的计算过程仅考虑了数据出现的概率，而未考虑实际数据大小，分布对指标的影响，不是良好的评价指标。

Mondrian

k	LM	Time(s)
10	0.77	0.42
20	0.93	0.31
40	1.15	0.30
80	1.46	0.25
160	1.99	0.22
320	2.88	0.22

由表中数据可以看出：

1. `Mondrian` 算法的 `LM` 随 `k` 的增大逐渐减小，表明泛化程度越大，可用性越差，与理论相符。
2. `Mondrian` 算法的运行时间随 `k` 的增大而减小，符合实验预期。

讨论与总结

本次实验完成了k匿名的两种算法——`Samarati` 和 `Mondrian`。算法均按照讲义中的伪代码进行实现。

在探究 `Samarati` 不同输出的可用性时，还实现了 `LM`，`DM` 和 `non-uniform entropy`。比较不同的评估方法，可以看出 `LM` 和 `DM` 的评估方法，可以较好地表现出数据可用性。此外，若要进一步改进评价指标，可以考虑给不同维度增加权重，因为在实际应用中，不同属性的价值有所不同。

在 `Mondrian` 算法中，进一步实现了对类别型数据的处理。具体实现方法为，先把类别型数据转为数据型。在执行完 `Mondrian` 算法后，再将转化的数据型数据还原为类别型数据。

在执行时间上，`Samarati` 算法的耗时较长，可能由于需要考虑同一泛化高度的所有可能情况带来的开销。`Mondrian` 算法在 `k` 值成倍增加时，执行时间并没有成倍减少，可能是数据读写等带来的影响，也可能算法的复杂度与 `k` 并不是反比关系。

参考文献

- El Emam K, Dankar FK, Issa R, et al. A globally optimal k-anonymity method for the de-identification of health data. *J Am Med Inform Assoc*. 2009;16(5):670-682. doi:10.1197/jamia.M3144
- Bayardo R, Agrawal R. Data privacy through optimal k-anonymization 2005. Proceedings of the 21st International Conference on Data Engineering.
- Samarati P. Protecting respondents identities in microdata release[J]. IEEE transactions on Knowledge and Data Engineering, 2001.
- LeFevre K, DeWitt D J, Ramakrishnan R. Mondrian multidimensional k-anonymity[C].(ICDE'06). IEEE, 2006.
- V. S. Iyengar , “Transforming data to satisfy privacy constraints,” in ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002.
- de Waal T, Willenborg L. Information loss through global recoding and local suppression *Neth Off Statistics* 1999;14:17-20.