论文题目1：基于不同匿名需求的隐私保护数据发布算法研究

三方面主要工作：

1. 提出了遵循（k，l）-anonymity匿名原则的 交叉桶泛化算法： 解决泛化算法对用户身份过度保护的问题，对用户身份和敏感属性进行相互独立的保护，身份暴露概率在1/k以内，敏感属性值泄露概率在1/l;启发式？1.减小等价组合桶包含的个体数2.缩小等价组中QI泛化值的值域范围。->进一步提高信息可利用性。
2. 定义个性化发布环境和提出局部分解算法：用户可自由设置属性值的敏感性，（根据包含数据值的类型 把属性分为半敏感、敏感、QI属性）。基于桶算法将带有敏感值的用户划分为桶，从而保证敏感值的安全。

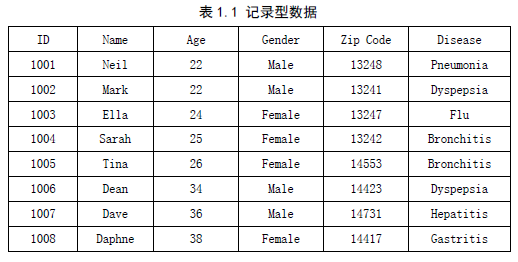
一般的匿名技术只将组成数据表的属性作为最小的敏感单位，当按照用户个人意愿并以数据值为最小敏感单位时，一个属性可能同时包括了非敏感值和敏感值。

1. 提出了局部分解泛化算法，在局部分解算法中加入泛化机制，泛化机制通过多维划分和ncp引导的启发式实现。



图 1 数据匿名及发布的过程

隐私保护数据发布技术 研究的两种匿名对象：1.记录型数据（匿名后发布）2.数据接受者向统计数据库发送查询语句后得到数据结果。攻击者可通过连续特定的查询语句攻击用户的隐私信息，所以需使用如满足差分隐私的噪音算法。



三种属性类型：

1.标识符属性：唯一或在很大程度上识别用户身份的属性，在发布数据中需要移除。

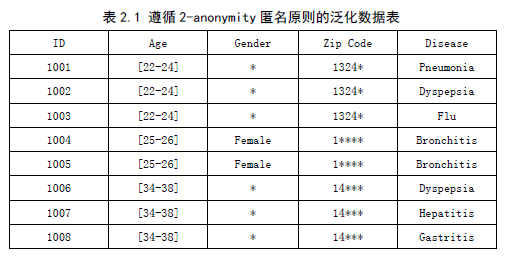
2.QI：用户的一般属性记录用户的非敏感信息，尽管单一项属性无法准确识别用户的身份，但多项QI值作为条件在数据表中匹配时会有极大的概率辨认用户的真实身份；

3.敏感属性：记录用户的敏感信息。

例如，在表 1.1 中，姓名属于标识符属性，年龄、性别和邮政编码属于 QI 属性，疾病属于 敏感属性。

**不同匿名原则的适用性简析**

1. k-Anonymity可以避免身份泄露，这是因为被k匿名化的数据集中的记录无法被映射回与原始数据集相对应的记录。



将表1.1中的标识符属性删除并且将QI值转化使其遵循2-anonymity匿名原则的结果，如表2.1所示。在表2.1中，攻击者使用任意个体的QI值进行匹配将至少得到两个无法区分的个体，因此，攻击者无法准确 地获得目标用户在数据表中的个体标识。

缺点：k-Anonymity不能保护敏感属性的泄露。比如：具有相同QI属性的k个病人，都患有相同的疾病，那只要确定了某个具有类似QI值的病人身份，就能推断出患有该病。如表2.1的1004和1005。

1. p-Sensitive k-Anonymity用于防止敏感属性的泄露，

定义：k>1 并且 p<=k,在满足k-Anonymity的前提下，具有相同QI值的记录至少有p种不同的敏感属性值。

缺点：该原则适用于不同敏感属性值出现频率相似的情况，如果不是这样，则会造成大量的数据丢失。比如：1000个病人中只有5个AIDS属性值为yes，其余为no，现在需要满足2-Sensitive k-Anonymity原则，那么最多只能划分为QI值相同的5组记录，这就意味着QI值需要进行粗粒度的泛化，从而造成信息的丢失。

1. l-Diversity解决k-Anonymity中存在的敏感属性泄露问题。

定义：每组具有相同QI值的记录，在每个敏感属性上至少有l种“well-represented”。对于“well-represented”，有以下几种定义方式：

1）不同值l-diversity。这种方式与l-sensitive k-anonymity相同。

2）熵l-diversity。对于一个组内特定敏感属性的熵的定义如下所示：

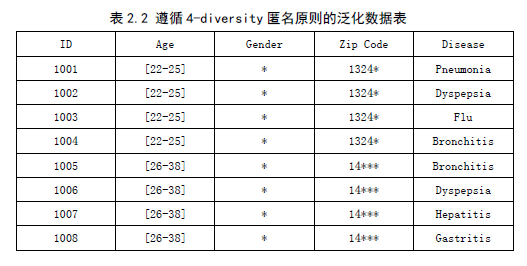
（2.1）

其中，p(G,c)是c在G中出现的概率，C是敏感属性值域，满足entropy l-diversity的数据集每个组G都有，。

3）Recursive (c,l)-diversity确保让最频繁的值出现的不要太多，让最不频繁的值出现的不要太少。m表示组G中敏感属性值的个数 表示G中第i个最频繁出现值得次数。如果，就说组G满足递归(c,l)-diversity，如果数据集中每个组都满足递归(c,l)-diversity，那么整个数据集就满足递归(c,l)-diversity。

举例

例如 ，表 2.2 为使用遵循 4-diversity 匿名原则的泛化算法对表 1.1 进行匿名之后的结果，并且匿名数据表中每个体敏感值暴露概率均为 1/4 。



缺点：难以且没必要实现，同2中例子。有以下两种攻击其无法防御：

——偏斜攻击Skewness attack。满足上述原则的组中，患有AIDS和未患有AIDS的病人数目相同，那么如果攻击者成功确认一位特殊病人属于这个组，那么就可以得到在这个5/1000的数据集中，该病人患有AIDS的概率有50%。

——相似性攻击Similarity attack。如果敏感属性值存在语义上的相似，隐私泄露还是会发生。比如在一个组中存在{lung cancer，liver cancer，stomach cancer}三种属性值，那攻击者可以推断出该组中的人患有癌症。

1. t-Closeness解决了l-diversity中固有的敏感属性泄露问题。

定义：每个组中敏感属性的分布和整个数据集属性的分布，之间距离小于阈值t。

因为两者分布接近，所以可以规避Skewness attack，此外，因为组中的属性分布模拟了整个数据集中的属性分布，所以只要数据集中没有语义相似，那么组中就不会存在语义相似。

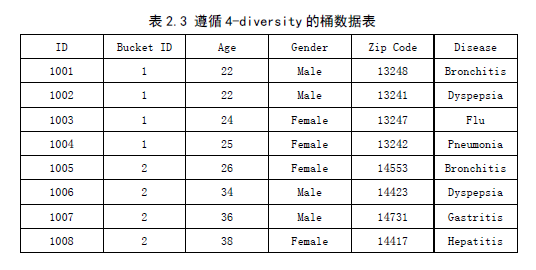
缺点：破坏了数据集的信息完整性，1.限制了有用信息的发布数量，2.破坏了QI和敏感信息值的相关性，不利于数据挖掘的进行，可以通过放宽t值来减小危害[15]。

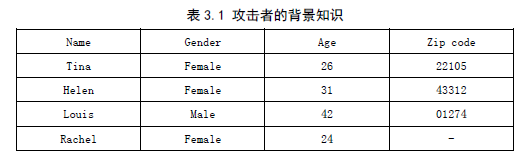
匿名算法 是为了使数据表满足匿名原则中的条件，对数据表进行匿名化的具体实现。

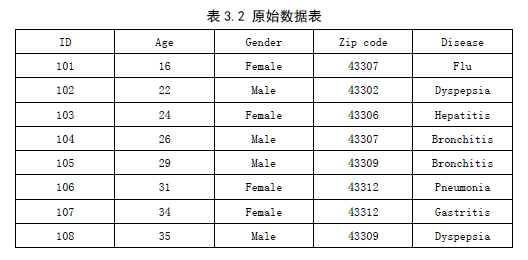
1. 泛化算法
2. 桶算法 将QI属性和敏感属性的关联切断，从而保护数据表中用户的敏感属性值信息。

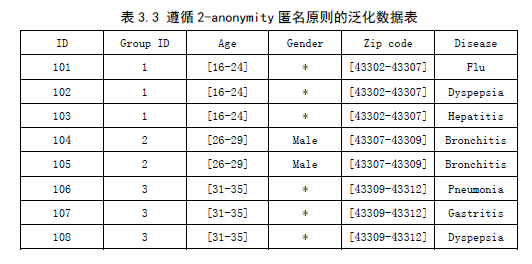
仅能保护敏感属性，数据信息可利用性高。

如表2.3所示，为使用遵循4-diversity匿名原则的桶算法对表1.1进行匿名的结果，其中，在每个桶中的QI属性与敏感属性之间的联系被切断，即任意用户对应于相应桶内每个敏感值的概率相等。

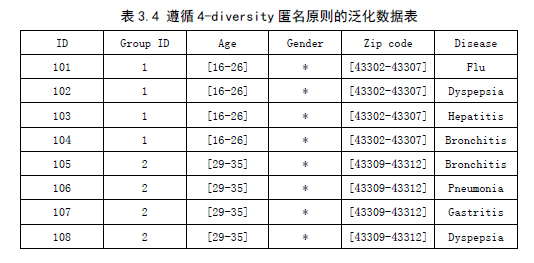




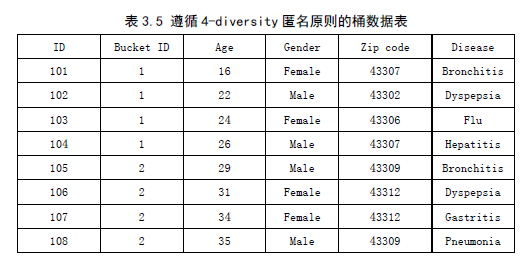




在未经匿名处理的表3.2中，攻击者可以轻易地通过匹配性别、年龄和邮政编码的属性值判断Helen的病症为肺炎。但是在表3.3中，攻击者无法判断Helen的ID为106、107或108，因此攻击者无法准确地推测出Helen的具体病症。此外，假设当攻击者不确定Helen的数据是否存在于数据表中时，表3.3也可以在一定程度上防止Helen的成员身份暴露，即攻击者无法确定Helen的信息存在于该数据表中。



比较表3.3和3.4的信息完整程度，表3.4的QI值精度较低



遵循 k-anonymity 匿名原则的泛化算法仍然存在比较严重的敏感属性值泄漏的问题。如在表 3.3 中，第 2 个等价组包含的疾病值全部为支气管炎。假设攻击者已知目标用户 ID 为 104 或 105 的 QI 值信息时，即使没有造成目标用户的身份泄露，攻击者仍然能推测出目标用户的病症为支气管炎。

据表的外部数据源。例如，攻击者在桶数据表3.5中匹配Rachel的QI值，可以推测出Rachel很可能曾经生病并去过发布数据表的医院就医。同时，攻击者还获得了Rachel的邮政编码值43306，从而增加了攻击者的背景知识。因此，尽管桶算法的信息可利用性十分出色，但是在隐私保护需求较高的场景中应用的范围非常有限。

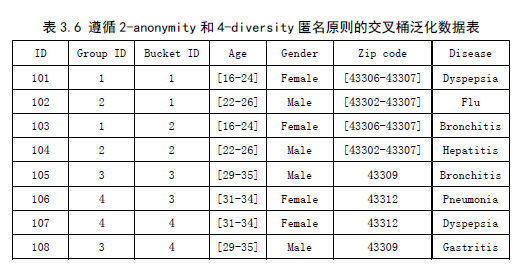
故提出了交叉桶泛化算法。

Cross-Bucket Generalization for Information and Privacy preservation

IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 30, NO. 3, MARCH 2018

交叉桶泛化算法可以分别根据用户身份和敏感属性的 匿名需求提供适当的保护。

1. 首先通过泛化算法将数据表中的个体划分为多个等价组，->满足用户身份匿名
2. 再将泛化后的个体划分为多个桶，来对敏感属性进行匿名保护。



与表3.1对比

例如，当攻击者使用Helen的QI值在匿名表3.6中进行匹配时，会得到Helen的信息在第四等价组中，但是第四等价组中包含的两个个体分别被划分在第三个和第四个桶中，因此，Helen的疾病值可能为第三和第四个桶中的任意值，即攻击者推测Helen的真实疾病值为肺炎的概率为1/4。

当交叉桶泛化算法遵循(*k*,*l*)-anonymity匿名原则时，对于匿名数据表中的任意个体，其用户身份和敏感属性值暴露的概率分别最多为1/𝑘和1/𝑙。

并且，为了尽可能提高匿名数据表中的信息可利用性，交叉桶泛化算法会尽量减少各个等价组和桶中包含的个体数量，同时尽可能缩小各个等价组中泛化QI值的值域范围。

通过实验对比传统的泛化算法和桶测试交叉泛化算法的实际匿名效果。其中，对比主要内容包括：

1. 交叉桶泛化算法比其他算法提供了更加安全的敏感属性保护；
2. 通过对比鉴别力度量和查询回答错误率的结果，证明交叉桶泛化算法相比传统的极大地提高了信息可利用性。此外，还通过调整参数k和l的值，对遵循 (k,l)-anonymity 匿名原则的交叉桶泛化算法产生的匿名影响进行研究。

基本概念和原理

1. 值域
2. 值
3. QI组
4. 等价组（QI组泛化）
5. 桶（QI组桶化）
6. 交叉桶泛化算法定义

对于一个数据表*T*，交叉桶泛化算法是将*T*划分为许多个等价组和桶，并且数据表中每个个体只能属于唯一的等价组和桶。假设数据表*T*被划分为*m*个等价组和*n*个桶,分别记为{EG1, EG2,…, EGm}和{B1, B2,…, Bn},并且有和，以及对任意,有和对任意，有。

数据表*T*包含了*d*个QI属性𝐴1𝑄𝐼,𝐴2𝑄𝐼,…,𝐴𝑑𝑄𝐼以及一个敏感属性𝐴𝑆𝐴，其中，每个属性都为数字类型或者分类类型。

令𝐷[𝐴]表示属性*A*的值域。对于任意个体𝑡∈𝑇，𝑡[𝐴]表示个体*t*中属性*A*的值。

重点1：原理

定理1：在一个使用交叉桶泛化算法进行匿名的数据表中，对于任意个体𝑡∈𝑇，其身份泄露的概率最多为1/|𝐺(𝑡)|，其中𝐺(𝑡)为包含*t*的等价组。

推论3.1：对于使用交叉桶泛化算法进行匿名的数据表，如果匿名数据表遵循*k*-anonymity匿名原则，则匿名表中每个等价组需要至少包含*k*个个体。

划重点 分析交叉桶泛化算法对敏感属性的保护

由交叉桶泛化算法定义3.4知，在同一个等价组中的个体也被分配到不同的桶中。

？？所以攻击者首先需要确定目标用户可能被划分在哪些桶里。

定义3.5：对于任意个体𝑡∈𝑇，其桶定位概率记为𝑝(𝑡,𝐵)，即个体*t*在桶*B*中的概率。

定义3.6：在一个交叉桶泛化数据表*T*中，对于任意个体𝑡∈𝑇，*MB*为个体*t*的一个匹配桶，当且仅当𝑡[𝐴𝑄𝐼]∈𝑀𝐵[𝐴𝑄𝐼]，其中𝑡[𝐴𝑄𝐼]为*t*的QI值，𝑀𝐵[𝐴𝑄𝐼]为*MB*中QI值的集合。

定理3.2：在一个使用交叉桶泛化算法进行匿名的数据表中，对于任意个体𝑡∈𝑇，其敏感属性值*s*暴露的概率满足：

(3.2)

其中，|𝑀𝐵(𝑠′)|为在匹配桶*MB*中出现次数最多的敏感值*s*’的数量，并且|𝑀𝐵|为*MB*中包含个体的数量。

证明：为了得到目标个体*t*的敏感值*s*，攻击者首先需要计算个体*t*在匿名数据表中每个桶的定位概率，以及个体*t*的敏感值为*s*的概率。因此，攻击者有

(3.3)

其中，𝑝(𝑠|𝑡,𝐵)为当个体*t*在桶*B*中时敏感值为*s*的概率。此外，攻击者可以排除不包含目标个体*t*的QI值的桶，即当𝑡[𝐴𝑄𝐼]∉𝐵[𝐴𝑄𝐼]时，有：

𝑝(𝑡,𝐵)=0 (3.4)

因此，根据定义3.6，有：

(3.5)

其中，*MB*为个体*t*的匹配桶。对任意*MB*中出现次数最高的敏感值*s*’，有：

|𝑀𝐵(𝑠)|≤|𝑀𝐵(𝑠′)| (3.6)

因此，有

(3.7)

根据式(3.5)，有

(3.8)

推论3.2：对于使用交叉桶泛化算法进行匿名的数据表，如果匿名数据表遵循*l*-diversity匿名原则，则对数据表中的所有个体需要满足如下条件：（1）在其所有匹配桶中，每个敏感值仅出现一次；（2）对于包含了目标个体的敏感值的匹配桶，需要满足

(3.9)

证明：根据条件（1），对于任意个体𝑡∈𝑇，由于在其所有的匹配桶中，每个敏感值只出现一次，如表3.6;因此只有一个匹配桶*MB*’包含了*t*的敏感值*s*，即有：

(3.10)

并且，对任意*t*的匹配桶*MB*，有：

(3.11)

根据定理3.2，有：

(3.12)

在条件（2）的限制下，有：

(3.13)

综上，当满足条件（1）和（2）时，交叉桶泛化算法遵循*l*-diversity匿名原则。

定义3.7：对于使用交叉桶泛化算法进行匿名的数据表，如果匿名数据表遵循(*k*,*l*)-anonymity匿名原则，当对任意个体𝑡∈𝑇，其用户身份泄露的概率最多为1/𝑘，并且有：

𝑝(𝑡,𝑠)≤1/𝑙 (3.14)

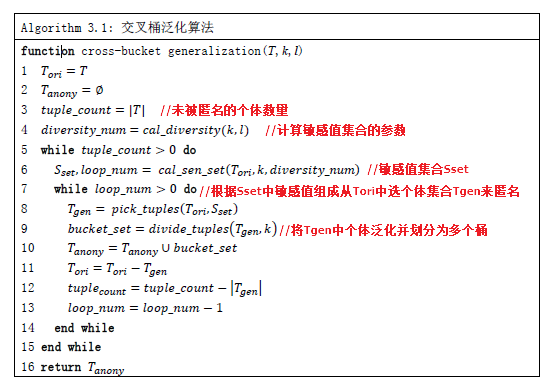
其中，*s*为*t*的敏感值。

交叉桶算法的额外目标（减少数据的损失）：

1. 等价组和桶中包含个体的数量尽可能的最少（原因1：等价组中个体少，说明有相同QI泛化值的个体少，即泛化程度低；原因2：桶中包含个体数少，说明桶的数量多，桶的数量越多说明？）
2. 泛化QI值得值域范围尽可能缩小，使得泛化QI值更加精确。

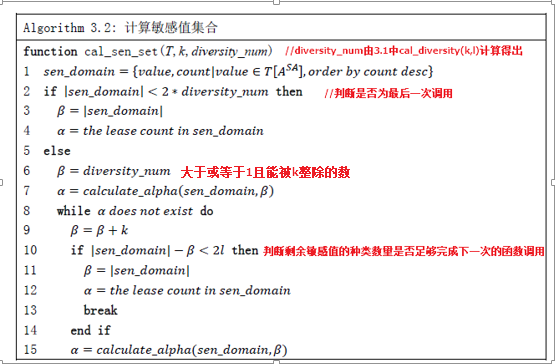
[96] Lefevre K, Dewitt D J, Ramakrishnan R. Mondrian Multidimensional K-Anonymity[C]. International Conference on Data Engineering. IEEE, 2006.

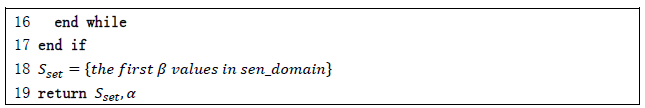
在这个阶段中，交叉桶泛化算法根据上一个阶段中计算的敏感值集合𝑆𝑠𝑒𝑡，在还未匿名的个体集合𝑇𝑜𝑟𝑖中选出合适的个体进行匿名。选出的个体需要与𝑆𝑠𝑒𝑡中的敏感值一一对应，并且为了提高匿名数据的信息可利用性，所选个体的QI值范围应该尽可能最小。



[113] Xiao X, Tao Y. m-invariance: towards privacy preserving re-publication of dynamic datasets[C]. Proceedings of the 2007 ACM SIGMOD international conference on Management of data. ACM, 2007: 689-700.

在这个阶段中，交叉桶泛化算法计算出一个敏感值集合及一个循环次数。计算出的敏感值集合结果将被用于在下一个阶段中选取进行匿名的个体集合，并直到满足循环次数为止。我们对*m*-invariance算法中分配阶段的算法进行了修改，用于实现算法3.1中的*cal\_sen\_set*(*T*,*k*,*diversity\_num*)函数。算法3.2给出了计算敏感值集合算法的主要描述。





？？

分配阶段接受前一个阶段（balance）的s\_作为参数，s\_=T(n)-T(n-1)

一次将α\*β个元组（称为Srmv）从s\_中移到 含有至少β>=m个敏感值的桶中。

α：每个桶中的敏感值集合，需要受Srmv中α个元组所支配。

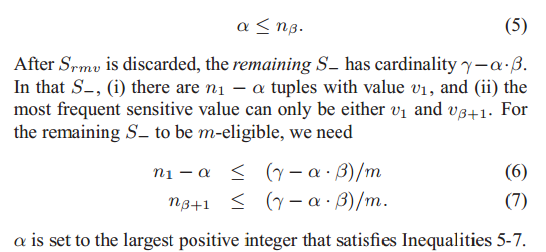
β：大于等于m

α尽可能大,以减小迭代次数

β尽可能小，因为更少的签名集合 更易于实现

α值的确定方法：

对 *m*-invariance 算法中分配阶段的算法



β值的获得方法：

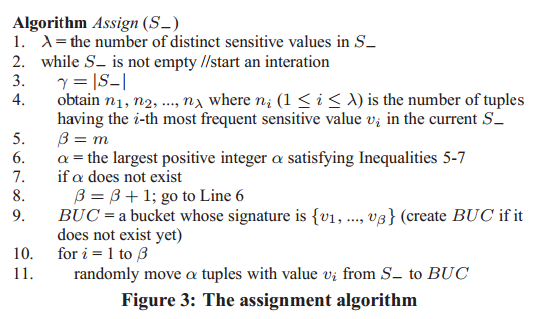
β的用法：

choose the β most frequent sensitive values v1, ..., vβ to form the signature of BUC.

从S\_中选取前β个频繁的敏感值作为桶的签名。

α的用法：pick α tuples in S− having the value vi 1<=i<=β

对每一个vi，i从1到β，从S\_中取α个含有敏感值vi的元组



m-eligible的意义:最多有1/m的元组有相同的敏感值。（防止无法平均分成m份，到相应的buc中）

从而保证平衡：在一个桶里 每个敏感值都 被相同数量的元组所拥有

Recursive (c,l)-diversity。确保让最频繁的值出现的不要太多，让最不频繁的值出现的不要太少。m表示组G中敏感属性值的个数 表示G中第i个最频繁出现值得次数。如果，就说组G满足递归(c,l)-diversity，如果数据集中每个组都满足递归(c,l)-diversity，那么整个数据集就满足递归(c,l)-diversity。

步长为什么要为k？

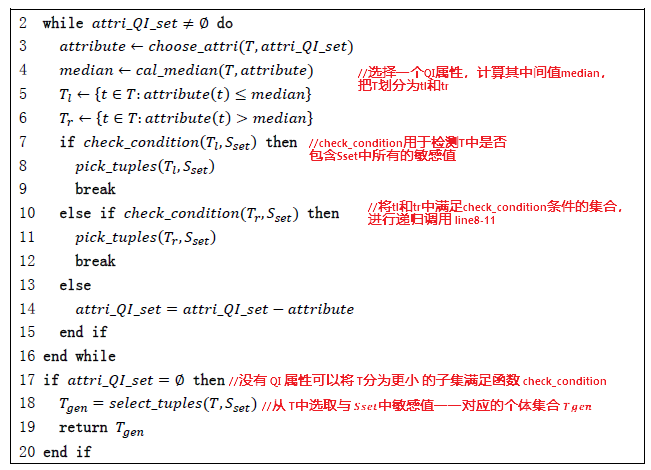
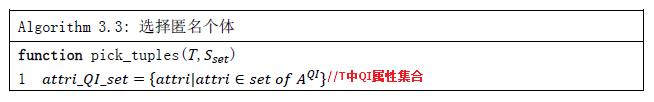
算法3.2生成的敏感值集合Sset满足两个性质：

1.每个包含于Sset的敏感值都只出现且仅出现一次（因为每次都选前β个频繁的敏感值）

2．如果Sset中包含的个体数量能被k整除，则Sset中包含的个体数量大于L；（diversity\_num大于l；应该是通过diversity\_num使得β大于l）否则，Sset中包含的个体数量大于2l。

（剩余不足2l的，将剩余的加上β一起作为Sset；而β大于l，）

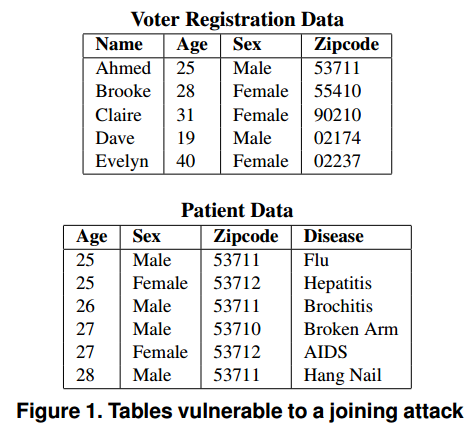
在这个阶段中，交叉桶泛化算法根据上一个阶段中计算的敏感值集合𝑆𝑠𝑒𝑡，在还未匿名的个体集合𝑇𝑜𝑟𝑖中选出合适的个体进行匿名。选出的个体需要与𝑆𝑠𝑒𝑡中的敏感值一一对应，并且为了提高匿名数据的信息可利用性，所选个体的QI值范围应该尽可能最小。我们对Mondrian算法进行了修改用于实现算法3.1中的*pick\_tuples*(*T*,𝑆𝑠𝑒𝑡)函数。算法3.3给出了选择匿名个体算法的主要描述。

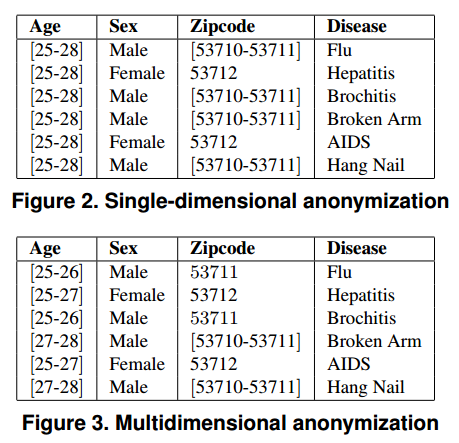


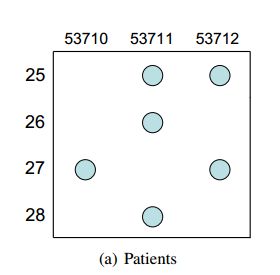
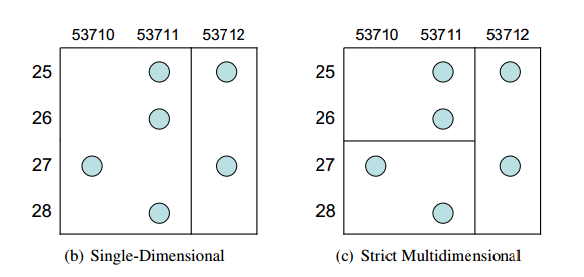
Mondrian Multidimensional K-Anonymity 论文

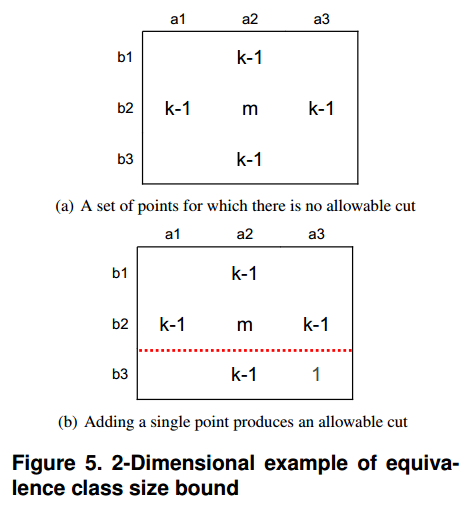
全局编码

多维k匿名 多维指的是 不同属性的值域向量，每个属性是一维，每个属性的值区间是也是 一维值区间。



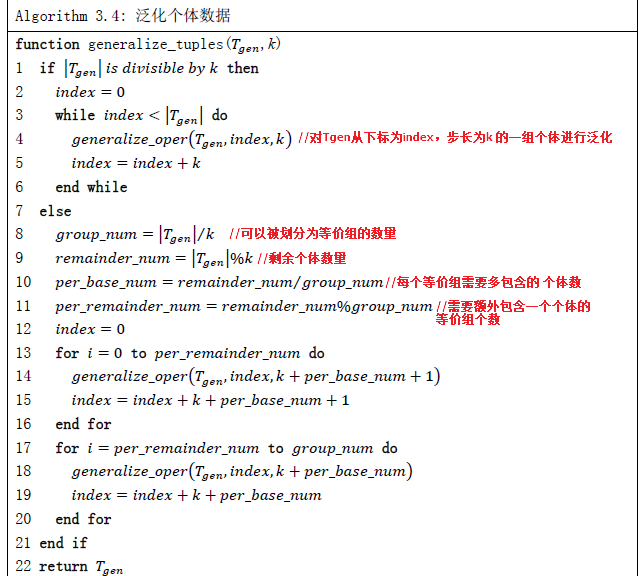




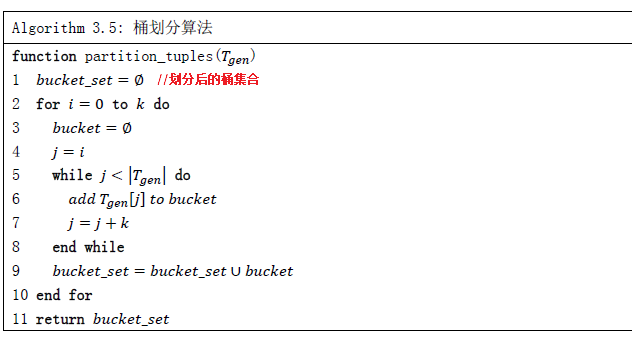
用于高维空间中的快速最近邻和近似最近邻查找技术——Kd-Tree（Kd树）

算法3.4 divide\_tuples函数第一部分将 𝑇𝑔𝑒𝑛进行泛化并使其遵循 k-anonymity 匿名原则



因为函数对Tgen进行泛化时，步长参数step的值一直不小于k，所以匿名后每个等价组至少包含了k个个体。

第二部分，算法3.5将𝑇𝑔𝑒𝑛进行桶划分并使其遵循*l*-diversity匿名原则



命题3.2：𝑇𝑔𝑒𝑛在经过算法3.5进行匿名之后遵循*l*-diversity匿名原则。

证明：由于𝑇𝑔𝑒𝑛与𝑆𝑠𝑒𝑡中的敏感值一一对应，因此𝑇𝑔𝑒𝑛同样具有性质3.1和3.2。由于𝑇𝑔𝑒𝑛中任意个体的所有匹配桶都包含于𝑇𝑔𝑒𝑛中，并且根据性质3.1，𝑇𝑔𝑒𝑛中的敏感值都只出现且仅出现一次，所以𝑇𝑔𝑒𝑛中的每个个体都满足推论3.2中的第一个条件。

接下来，我们证明𝑇𝑔𝑒𝑛中的每个个体都满足推论3.2中的第二个条件。考虑𝑇𝑔𝑒𝑛的两种情况：（1）𝑇𝑔𝑒𝑛中包含个体的数量能被*k*整除。根据算法3.4，每*k*个个体组成一个等价组。在算法3.5中，每个桶中正好包含了每个等价组中的一个个体。由于每个等价组都包含了*k*个个体，因此有：

(3.15)

和

(3.16)

根据性质3.2，𝑇𝑔𝑒𝑛中包含个体的数量不少于*l*，因此有：

(3.17)

（2）𝑇𝑔𝑒𝑛中包含个体的数量不能被*k*整除。根据算法3.4，𝑇𝑔𝑒𝑛中包含了两种个体数量不同的等价组，分别为*k*+*per\_base\_num*和*k*+*per\_base\_num*+1。并且，由于*remainder\_num*小于*k*，所以这两种等价组包含的个体数量都小于2*k*。在算法3.5中，每个桶中最多包含两个个体在同一个等价组中，即有

(3.18)

和

(3.19)

根据性质3.2，𝑇𝑔𝑒𝑛中包含个体的数量大于2*l*，因此有

(3.20)

综上所述，𝑇𝑔𝑒𝑛中的每个个体都满足推论3.2中的两个条件，因此𝑇𝑔𝑒𝑛在经过算法3.5进行匿名之后遵循*l*-diversity匿名原则。

实验：

1.数据

美国人口普查数据,删除了丢失属性值的个体，其中，QI属性包括了性别、年龄、家庭关系、婚姻状况、种族、教育情况、每周工作时长和职业，敏感属性为薪水。表3.7详细地介绍了这些属性的相关信息。

2.实验分三部分:

Mondrian算法和Anatomy算法与交叉桶泛化算法进行对比，其中，对比内容包括了敏感属性保护和信息可利用性两方面。

除此之外，还对(*k*,*l*)-anonymity匿名原则中参数*k*和*l*的变化对交叉桶泛化算法的匿名影响进行了研究。

1. 敏感属性保护

攻击者背景：

在测试敏感属性保护的实验中，我们假定最坏的情况：攻击者已知目标用户的信息存在于匿名表中，并且已经获得目标用户的所有QI值信息。攻击者将通过目标用户的QI值在匿名表中进行匹配，并且试图获得目标用户的敏感属性值。

计算方法：

我们将对原始数据中的所有用户进行测试，并且计算敏感属性值暴露概率的平均值进行比较。

未使用桶算法匿名的数据表中，t的敏感值s的暴露概率为：

表示t的匹配个体的数量

表示匹配个体中敏感值为s的数量

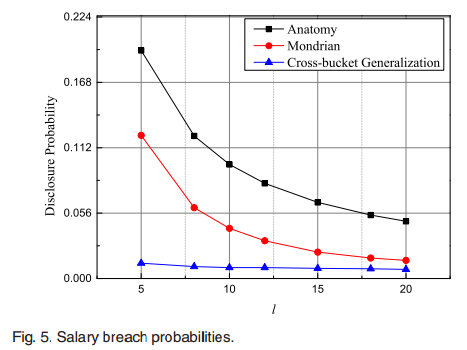
使用桶算法的数据表：

表示MB中t的匹配个体的数量

表示MB中敏感值为s的个体数量

我们使Mondrian算法和Anatomy算法都遵循*l*-diversity匿名原则，交叉桶泛化算法遵循(*k*,*l*)-anonymity匿名原则，并将参数*k*的值固定为3。图3.6显示了使用各个匿名算法时敏感属性值暴露概率的结果。

从图3.6中可以发现，Anatomy算法发挥了遵循*l*-diversity匿名原则的预期表现。但是，由于遵循*l*-diversity匿名原则的Mondrian算法在等价组中包含了更多的个体数量，因此其保护效果要比Anatomy算法更加出色。交叉桶泛化算法的保护效果比前两者更加优秀，这是由于在匿名表中等价组包含的个体被分配至不同的桶中，所以当攻击者通过目标用户的QI值进行匹配时，会得到很多目标用户的匹配桶，从而增加了获取敏感值的难度。

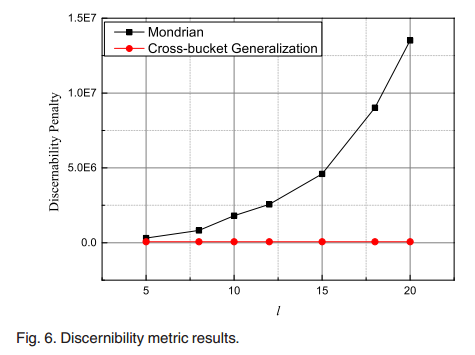


信息可利用性

在本次实验中，采用了与之前相同的实验测试环境，即令Mondrian 算法遵循*l*-diversity匿名原则，交叉桶泛化算法遵循(*k*,*l*)-anonymity匿名原则，并将参数*k*的值设置为3。我们首先对交叉桶泛化算法和Mondrian算法的鉴别力度量进行测试。图3.7即为使用这两个算法对数据表进行匿名之后的鉴别力度量惩罚指数结果。

鉴别力度量（Discernibility Metric）：鉴别力度量主要用于对使用泛化算法匿名的数据表进行评估。它的表达公式为：

其中，*EG*表示匿名数据表中的等价组，|𝐸𝐺|表示等价组*EG*中包含个体的数量。



随着参数 *l* 的增加， Mondrian 算法的鉴别力度量惩罚指数上升非常快，已经和交叉桶泛化算法的结果产生了量级的差距。

对于使用交叉桶泛化算法进行匿名的数据表，其等价组包含个体的数量远远小于使用Mondrian 算法进行匿名时的情况。

交叉桶泛化算法的鉴别力度量惩罚指数一直保持不变，即对于使用交叉桶泛化算法进行匿名的数据表，其等价组包含个体的数量几乎不会被参数 *l* 所影响。因此，可以推断当改变对敏感属性的保护强度时，对用户身份的保护效果不会受到影响。

用查询回答错误率对交叉桶泛化算法的匿名效果进行测试

匿名效果进行测试。

通过随机生成1000条查询语句对每个匿名表计算查询回答错误率的平均值，并将结果进行对比。

查询语句的形式如下：SELECT SUM(𝑠𝑎𝑙𝑎𝑟𝑦) FROM Microdata WHERE 𝑝𝑟𝑒𝑑(𝐴1𝑄𝐼) AND 𝑝𝑟𝑒𝑑(𝐴2𝑄𝐼) AND 𝑝𝑟𝑒𝑑(𝐴3𝑄𝐼) AND 𝑝𝑟𝑒𝑑(𝐴4𝑄𝐼)

其中，查询语句的条件中随机包含了四个QI属性，并且将薪水的总和作为比较结果。对于分类QI属性，𝑝𝑟𝑒𝑑(𝐴𝑄𝐼)表示为：

(𝐴𝑄𝐼=𝑣1 𝑜𝑟 𝐴𝑄𝐼=𝑣2 𝑜𝑟… 𝑜𝑟 𝐴𝑄𝐼=𝑣𝑚) (3.23)

其中，𝑣𝑖(1≤𝑖≤𝑚)为在𝐷[𝐴𝑄𝐼]中的随机值。而对于数字QI属性，𝑝𝑟𝑒𝑑(𝐴𝑄𝐼)表示为：(𝐴𝑄𝐼>𝑣) 𝑜𝑟 (𝐴𝑄𝐼<𝑣) 𝑜𝑟 (𝐴𝑄𝐼=𝑣) 𝑜𝑟 (𝐴𝑄𝐼≥𝑣)

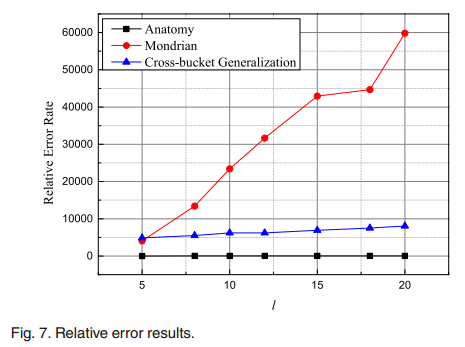
𝑜𝑟 (𝐴𝑄𝐼≤𝑣) 𝑜𝑟 (𝐴𝑄𝐼≠𝑣) (3.24)

其中，*v*为在𝐷[𝐴𝑄𝐼]中的随机值。

查询回答错误率的公式为：

(3.25)

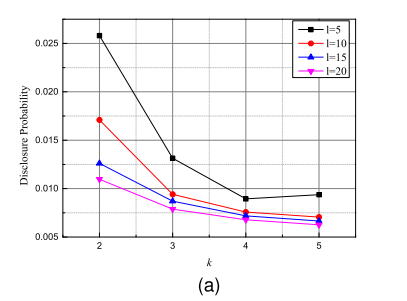
其中𝑆𝑢𝑚𝑢𝑝𝑝𝑒𝑟和𝑆𝑢𝑚𝑙𝑜𝑤𝑒𝑟分别为薪水总和的上限和下限，并且𝑆𝑢𝑚𝑎𝑐𝑡为薪水的实际总和。图3.8显示了使用各个匿名算法时查询回答错误率的结果。



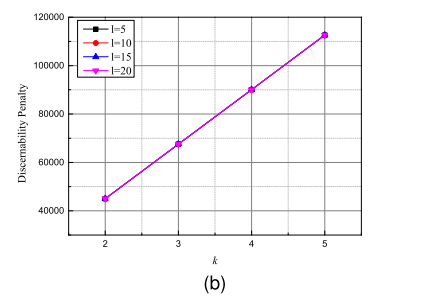
通过对图 3.8 的观察可以发现交叉桶泛化算法的查询回答错误率高于Anatomy 算法。这是由于 Anatomy 算法完全保留了原始数据表中的 QI 值信息，使得匿名数据表获得了接近于原始数据表的信息可利用性。

但是，交叉桶泛化算法不仅实现了保护用户身份的功能，而且当随着参数 *l* 的不断增加时，交叉桶泛化算法的查询回答错误率上升的非常缓慢，以至于远远低于 Mondrian 算法。因此，综合上述实验结果来看，当匿名数据表需要同时为用户身份和敏感属性提供保护时，交叉桶泛化算法是相对于 Mondrian 算法和 Anatomy 算法更加合适的选择。

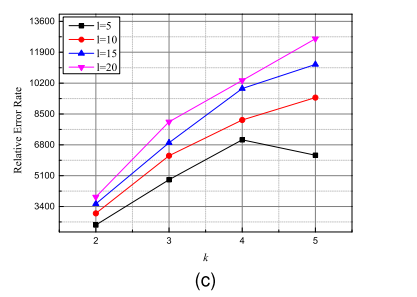
过调整参数 *k* 和 *l* 的值，对遵循(*k*,*l*)-anonymity匿名原则的交叉桶泛化算法产生的影响进行测试和分析。参数 *l* 的值分别固定为5、 10、 15 和 20，其他实验环境与之前的实验相同。相关的实验结果分别如图3.9、 3.10 和 3.11。



当敏感值暴露的概率在 0.01 以上时，提高参数 *k* 的值  
对提高敏感属性的保护效果具有比较显著的作用。



四条线已经完全重合。这表明对于使用遵循(*k*,*l*)-anonymity 匿名原则的交叉桶泛化算法进行匿名的数据表，等价组包含个体的数量完全取决于参数 *k* 的值，而与参数 *l* 的值无关。



当参数 *l* 的值越大时，随着参数 *k* 的值增加，查询回答错误率的上升也越快。