Generating Microdata with *P*-Sensitive *K*-Anonymity Property

Traian Marius Truta[1], Alina Campan[2], Paul Meyer[1]

[1] Department of Computer Science, Northern Kentucky University,
Highland Heights, KY 41099, U.S.A.,
{trutat1, meyerp1}@nku.edu
[2] Department of Computer Science, Babes-Bolyai University,
Cluj-Napoca, RO-400084, Romania,
alina@cs.ubbcluj.ro

Abstract. Existing privacy regulations together with large amounts of available data have created a huge interest in data privacy research. A main research direction is built around the *k*-anonymity property. Several shortcomings of the *k*-anonymity model have been fixed by new privacy models such as *p*-sensitive *k*-anonymity, *l*-diversity, (α, *k*)-anonymity, and *t*-closeness. In this paper we introduce the *EnhancedPKClustering* algorithm for generating *p*-sensitive *k*-anonymous microdata based on frequency distribution of sensitive attribute values. The *p*-sensitive *k*-anonymity model and its enhancement, extended *p*-sensitive *k*-anonymity, are described, their properties are presented, and two diversity measures are introduced. Our experiments have shown that the proposed algorithm improves several cost measures over existing algorithms.

Keywords: Privacy, *k*-anonymity, *p*-sensitive *k*-anonymity, attribute disclosure.

1 Introduction

The increased availability of individual data has nowadays created a major privacy concern. Legislators from many countries have tried to regulate the use and disclosure of confidential information (or data) [2]. New privacy regulations, such as the *Health Insurance Portability and Accountability Act* (*HIPAA*) [7], along with the necessity of collecting personal information have generated a growing interest in privacy research. Several techniques that aim to avoid the disclosure of confidential information by processing sensitive data before public release have been presented in the literature. Among them, the *k*-anonymity model was recently introduced [16, 17]. This property requires that in the *released* (a.k.a. *masked*) *microdata* (datasets where each tuple belongs to an individual entity, e.g. a person, a company) every tuple will be indistinguishable from at least (*k*-1) other tuples with respect to a subset of attributes called *key* or *quasi-identifier* attributes.

Although the model's properties, and the techniques used to enforce it on data, have been extensively studied [1, 4, 11, 16, 18, 20, etc.], recent results have shown

that *k*-anonymity fails to protect the privacy of individuals in all situations [14, 19, 23, etc.]. New enhanced privacy models have been proposed in the literature to deal with *k*-anonymity's limitations with respect to *sensitive attributes disclosure* (this term will be explained in the next section). These models follow one of the following two approaches: the *universal* approach uses the same privacy constraints for all individual entities, while the *personalized* approach allows users or data owners to customize the amount of privacy they need. The first category of privacy protection models, based on the universal approach, includes: *p*-sensitive *k*-anonymity [19] with its extension called extended *p*-sensitive *k*-anonymity [5], *l*-diversity [14], (α, *k*)-anonymity [22], and *t*-closeness [13]. The only personalized privacy protection model we are aware of is personalized anonymity [23].

In this paper we introduce an efficient algorithm for anonymizing a microdata set such that its released version will satisfy *p*-sensitive *k*-anonymity. Our main interest in developing a new anonymization algorithm was to obtain better *p*-sensitive *k*-anonymous solutions w.r.t. various cost measures than the existing algorithms by taking advantage of the known properties of the *p*-sensitive *k*-anonymity model.

In order to describe the algorithm, the *p*-sensitive *k*-anonymity model, extended *p*-sensitive *k*-anonymity model, and their properties are presented. Along with existing cost measures such as *discernability measure (DM)* [3] and *normalized average cluster size metric* (*AVG*) [12], two diversity measures are introduced. The proposed algorithm is based on initial microdata frequency distribution of sensitive attribute values. It partitions an initial microdata set into clusters using the properties of the *p*-sensitive *k*-anonymity model. The released microdata set is formed by generalizing the quasi-identifier attributes of all tuples inside each cluster to the same values. We compare the results obtained by our algorithm with the results of those from both the *Incognito* algorithm [11], which was adapted to generate *p*-sensitive *k*-anonymous microdata, and the *GreedyPKClustering* algorithm [6].

The paper is structured as follows. Section 2 presents the *p*-sensitive *k*-anonymity model along with its extension. Section 3 introduces the *EnhancedPKClustering* algorithm. Experimental results and conclusions are presented in Sections 4 and 5.

2 Privacy Models

2.1 *p*-Sensitive *k*-Anonymity Model

The *p*-sensitive *k*-anonymity model is a natural extension of *k*-anonymity that avoids several shortcomings of this model [19]. Next, we present these two models.

A microdata is a set of tuples in the relational sense. The initial dataset (called initial microdata and labeled *IM*) is described by a set of attributes that are classified into the following three categories:

- I_1, I_2,..., I_m are *identifier* attributes such as *Name* and *SSN* that can be used to identify a record.
- K_1, K_2,…, K_q are *key* or *quasi-identifier* attributes such as *ZipCode* and *Sex* that may be known by an intruder.

- $S_1, S_2, \ldots, S_r$ are *confidential* or *sensitive* attributes such as *Diagnosis* and *Income* that are assumed to be unknown to an intruder.

In the released dataset (called *masked microdata* and labeled $\mathcal{MM}$) only the quasi-identifier and confidential attributes are preserved; identifier attributes are removed as a prime measure for ensuring data privacy. Although direct identifiers are removed, an intruder may use record linkage techniques between externally available datasets and the quasi-identifier attributes values from the masked microdata to glean the identity of individuals [21]. To avoid this possibility of disclosure, one frequently used solution is to further process (modify) the initial microdata through generalization and suppression of quasi-identifier attributes values, so that to enforce the *k*-anonymity property for the masked microdata. In order to rigorously and succinctly express *k*-anonymity property, we use the following concept:

Definition 1 (***QI-cluster***): Given a microdata, a ***QI-cluster*** consists of all the tuples with identical combination of quasi-identifier attribute values in that microdata.

There is no consensus in the literature over the term used to denote a *QI*-cluster. This term was not defined when *k*-anonymity was introduced [16, 17]. More recent papers use different terminologies such as *equivalence class* [22] and *QI-group* [23].

We define *k*-anonymity based on the minimum size of all *QI*-clusters.

Definition 2 (***k-anonymity property***): The ***k-anonymity property*** for a $\mathcal{MM}$ is satisfied if every *QI*-cluster from $\mathcal{MM}$ contains *k* or more tuples.

Unfortunately, *k*-anonymity does not provide the amount of confidentiality required for every individual [14, 19, 22]. To briefly justify this affirmation, we distinguish between two possible types of disclosure; namely, *identity disclosure* and *attribute disclosure*. *Identity disclosure* refers to re-identification of an entity (person, institution) and *attribute disclosure* occurs when the intruder finds out something new about the target entity [10]. *K*-anonymity protects against identity disclosure but fails to protect against attribute disclosure when all tuples of a *QI*-cluster share the same value for one sensitive attribute [19]. This attack is called *homogeneity attack* [14] and can be avoided by enforcing a more powerful anonymity model than *k*-anonymity, for example *p*-sensitive *k*-anonymity. A different type of attack, called *background attack*, is presented in [14]. In this attack, the intruder uses background information that allows him / her to rule out some possible values of the sensitive attributes for specific individuals. Protection against background attacks is more difficult since the data owner is unaware of the type of background knowledge an intruder may posses. To solve this problem particular assumptions should be made, and anonymization techniques by themselves will not fully eliminate the risk of the background attack [22]. Still, enhanced anonymization techniques try to perform as well as possible in case of background attacks.

The *p*-sensitive *k*-anonymity model considers several sensitive attributes that must be protected against attribute disclosure. Although initially designed to protect against homogeneity attacks, it also performs well against different types of background attacks. It has the advantage of simplicity and allows the data owner to customize the desired protection level by setting various values for *p* and *k*. Intuitively, the larger the parameter *p*, the better is the protection against both types of attacks.

Definition 3 (***p-sensitive k-anonymity property***): A $\mathcal{MM}$ satisfies ***p-sensitive k-anonymity property*** if it satisfies *k*-anonymity and the number of distinct attributes for each confidential attribute is at least *p* within the same *QI*-cluster from the $\mathcal{MM}$.

To illustrate this property, we consider the masked microdata from Table 1 where *Age* and *ZipCode* are quasi-identifier attributes, and *Diagnosis* and *Income* are confidential attributes:

Table 1. Masked microdata example for *p*-sensitive *k*-anonymity property.

Age	ZipCode	Diagnosis	Income
20	41099	AIDS	60,000
20	41099	AIDS	60,000
20	41099	AIDS	40,000
30	41099	Diabetes	50,000
30	41099	Diabetes	40,000
30	41099	Tuberculosis	50,000
30	41099	Tuberculosis	40,000

The above masked microdata satisfies 3-anonymity property with respect to *Age* and *ZipCode*. To determine the value of *p*, we analyze each *QI*-cluster with respect to their confidential attribute values. The first *QI*-cluster (the first three tuples in Table 1) has two different incomes (*60,000* and *40,000*), and only one diagnosis (*AIDS*), therefore the highest value of *p* for which *p*-sensitive 3-anonymity holds is 1. As a result, a presumptive intruder who searches information about a young person in his twenties that lives in zip code area 41099 will discover that the target entity suffers from *AIDS*, even if he doesn't know which tuple in the first *QI*-cluster corresponds to that person. This attribute disclosure problem can be avoided if one of the tuples from the first *QI*-cluster would have a value other than *AIDS* for *Diagnosis* attribute. In this case, both *QI*-clusters would have two different illnesses and two different incomes, and, as a result, the highest value of *p* would be 2.

From the definitions of *k*-anonymity and *p*-sensitive *k*-anonymity models we easily infer that 2-sensitivity 2-anonymity is a necessary condition to protect any masked microdata against any type of disclosure, identity or attribute disclosure. Unfortunately, the danger of disclosure is not completely eliminated since an intruder may "guess" the identity or attribute value of some individuals with a probability of ½. For many masked microdata such a high probability is unacceptable, and the values of *k* and/or *p* must be increased.

2.2 *p*-Sensitive *k*-Anonymity Model Properties

We introduce the following notations, which will be used for expressing several properties of *p*-sensitive *k*-anonymity and for presenting our anonymization algorithm. For any given microdata set $\mathcal{M}$, we denote by:

- n – the *number* of tuples in $\mathcal{M}$.
- r – the number of confidential attributes in $\mathcal{M}$.
- s_j – the number of distinct values for the confidential attribute S_j ($1 \le j \le r$).

- v_i^j – the distinct values for the confidential attribute S_j in descending order of their occurrences ($1 \le j \le r$ and $1 \le i \le s_j$).
- f_i^j – the number of occurrences of the value v_i^j for the confidential attribute S_j; in other words the ***descending ordered frequency set*** [11] for the confidential attribute S_j ($1 \le j \le r$ and $1 \le i \le s_j$). For each sensitive attribute S_j the following inequality holds: $f_1^j \ge f_2^j \ge \ldots \ge f_{s_j}^j$.
- SEC_i^j – the set of tuples from $\mathcal{M}$ such that they all have the value v_i^j for S_j ($1 \le j \le r$ and $1 \le i \le s_j$), in other words $SEC_i^j = \sigma_{S_j = v_i^j}(\mathcal{M})$. We use the term of ***a sensitive equivalence class*** or attribute S_j to refer to any SEC_i^j. The cardinality of SEC_i^j is f_i^j.
- cf_i^j – the ***cumulative descending ordered frequency set*** for the confidential attribute S_j ($1 \le j \le r$ and $1 \le i \le s_j$) [19]. In other words, $cf_i^j = \sum_{k=1}^{i} f_k^j$.
- $cf_i = \max_{j=1,r}(cf_i^j)$ $(0 \le i \le \min_{j=1,r}(s_j))$ – the ***maximum between i^{th} cumulative descending ordered frequencies,*** for all sensitive attributes. We define $cf_0 = 0$.
- $pSEC_i^j = \begin{cases} SEC_i^j, & if \quad i < p \\ SEC_p^j \cup SEC_{p+1}^j \cup \ldots \cup SEC_{s_j}^j, & if \quad i = p \end{cases}$, ($1 \le j \le r$ and $1 \le i \le p$). We call each $pSEC_i^j$ as a ***p-sensitive equivalence class*** of attribute S_j. Each sensitive attribute S_j partitions the tuples in $\mathcal{M}$ in p p-sensitive equivalence classes. Moreover, the size of these equivalence classes descends from the $pSEC_1^j$ to $pSEC_{p-1}^j$. The last p-sensitive equivalence class, $pSEC_p^j$, does not follow this pattern.

P-sensitive *k*-anonymity can not be enforced for any given $\mathcal{IM}$, for any *p* and *k*. We present next two necessary conditions that express when this is possible [19].

Condition 1 (*First necessary condition for an $\mathcal{MM}$ to have p-sensitive k-anonymity property*): The minimum number of distinct values for each confidential attribute in $\mathcal{IM}$ must be greater than or equal to *p*.

A second necessary condition establishes the maximum possible number of *QI*-clusters in the masked microdata that satisfy *p*-sensitive *k*-anonymity. To specify this upper bound we use the maximum between cumulative descending ordered frequencies for each sensitive attribute in $\mathcal{IM}$ [19].

Condition 2 (*Second necessary condition for a $\mathcal{MM}$ to have p-sensitive k-anonymity property*): The maximum possible number of *QI*-clusters in the masked microdata is $\textbf{\textit{maxClusters}} = \min_{i=1,p}\left\lfloor \frac{n - cf_{p-i}}{i} \right\rfloor$.

Proof: We assume that for a given $\mathcal{IM}$, *k* and *p*, the maximum possible number of

QI-clusters in the masked microdata $maxClusters > \min_{i=1,p} \left\lfloor \frac{n - cf_{p-i}}{i} \right\rfloor$. Let *iVal* be the *i* value for which $\frac{n-cf_{p-i}}{i}$ is minimum. We have:

$$maxClusters > \frac{n - cf_{p-iVal}}{iVal} \text{ and } maxClusters \cdot iVal > n - cf_{p-iVal}. \quad (1)$$

Since cf_{p-iVal} tuples have only $p - iVal$ distinct values for a confidential attribute (from the definition of cumulative frequencies), the remaining tuples ($n - cf_{p-iVal}$) must contribute with at least *iVal* tuples to every cluster. In other words: $n - cf_{p-iVal} \geq maxClusters \cdot iVal$, relation that contradicts (1). Q.E.D.

Condition 2 provides a superior limit of the number of *p*-sensitive *QI*-clusters that can be formed in a microdata set, and not the actual number of such clusters that exist in data. Therefore, even the optimal partition w.r.t. the partition cardinality criterion could consist in less *p*-sensitive *QI*-clusters than the number estimated by Condition 2. Next, we give such an example where *maxClusters* value calculated according to Condition 2 is strictly greater than the maximum number of *p*-sensitive equivalence classes within the microdata. Fig. 1 contains a microdata described by 3 sensitive attributes together with the corresponding f_i^j and cf_i^j values.

A	B	C
1	a	α
1	b	β
2	a	β
2	b	α

	s_j	f_1^j	f_2^j
$j = 1$	A	2	2
$j = 2$	B	2	2
$j = 3$	C	2	2

cf_1^j	cf_2^j
2	4
2	4
2	4
cf_1	cf_2
2	4

Fig. 1. A microdata with corresponding frequency / cumulative frequency set values.

For *p*=2, $maxClusters = \min_{i=1,p} \left\lfloor \frac{n - cf_{p-i}}{i} \right\rfloor = \left\lfloor \frac{4-2}{1} \right\rfloor = 2$. In fact, only one group that is 2-sensitive can be formed with these tuples!

2.3 Extended *p*-Sensitive *k*-Anonymity Model

The values of the attributes, in particular the categorical ones, are often organized according to some hierarchies. Although Samarati and Sweeney introduced the concept of value generalization hierarchy for only quasi-identifier attributes [16, 17], these hierarchies can be applied and used for sensitive attributes as well. For example, in medical datasets, the sensitive attribute *Illness* has values as specified by the *ICD9* codes (see Fig. 2) [8]. The data owner may want to protect not only the leaf values as in the *p*-sensitive *k*-anonymity model, but also values found at higher levels. For example, the information that a person has *cancer* (not a leaf value in this case) needs to be protected, regardless of the cancer type she has (*colon cancer*, *prostate cancer*, *breast cancer* are examples of leaf nodes in this hierarchy). If *p*-sensitive *k*-anonymity

property is enforced for the released microdata, it is possible that for one *QI*-cluster all of the *Illness* attribute values to be descendants of the *cancer* node in the corresponding hierarchy, therefore leading to disclosure. To avoid such situations, the extended *p*-sensitive *k*-anonymity model was introduced [5].

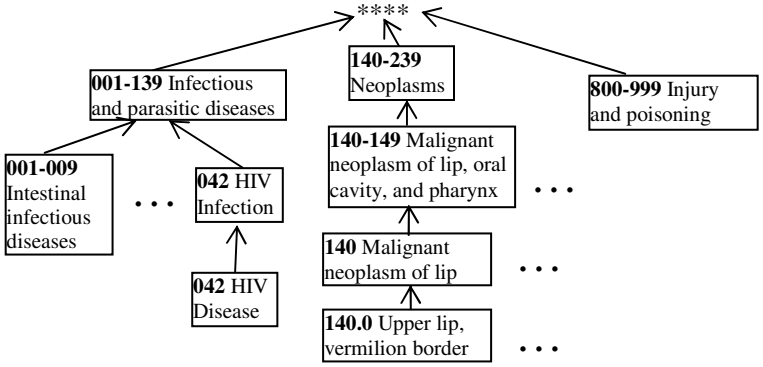

Fig. 2. ICD9 disease hierarchy and codes.

For the sensitive attribute S we use the notation HV_S to represent its value generalization hierarchy. We assume that the data owner has the following requirements in order to release a masked microdata:

- All ground values in HV_S must be protected against disclosure.
- Some non-ground values in HV_S must be protected against disclosure.
- All the descendants of a protected non-ground value in HV_S must also be protected.

Definition 4 (*strong value*): A protected value in the value generalization hierarchy HV_S of a confidential attribute S is called ***strong*** if none of its ascendants (including the root) is protected.

Definition 5 (*protected subtree*): We define a ***protected subtree*** of a hierarchy HV_S as a subtree in HV_S that has as root a strong protected value.

Definition 6 (*extended p-sensitive k-anonymity property*): The masked microdata ($\mathcal{MM}$) satisfies ***extended p-sensitive k-anonymity property*** if it satisfies *k*-anonymity and for each *QI*-cluster from $\mathcal{MM}$, and the values of each confidential attribute S within that group belong to at least p different protected subtrees in HV_S.

The necessary conditions to achieve extended *p*-sensitive *k*-anonymity on microdata are similar with the ones presented for *p*-sensitive *k*-anonymity model.

At a closer look, extended *p*-sensitive *k*-anonymity for a microdata is equivalent to *p*-sensitive *k*-anonymity for the same microdata where the confidential attributes values are generalized to their first protected ancestor starting from the hierarchy root (their strong ancestor). Consequently, in order to enforce extended *p*-sensitive *k*-anonymity to a dataset, the following two-steps procedure can be applied:

- Each value of a confidential attribute is generalized (temporarily) to its first strong ancestor (including itself).
- Any algorithm which can be used for *p*-sensitive *k*-anonymization is applied to the modified dataset. In the resulted masked microdata the original values of the confidential attributes are restored.

The dataset obtained following these steps respects the extended p-sensitive k-anonymity property.

3 Privacy Algorithms

Anonymization algorithms, besides achieving the properties required by the target privacy model (p-sensitive k-anonymity, l-diversity, (α, k)-anonymity, t-closeness), must also consider minimizing one or more cost measure. We know that optimal k-anonymization is a NP-hard problem [1]. By simple reduction to k-anonymity, it can be easily shown that p-sensitive k-anonymization is also a NP-hard problem. Several polynomial algorithms that achieve a suboptimal solution currently exist for enforcing p-sensitive k-anonymity and other similar models on microdata. In [6] we described a greedy clustering algorithm (*GreedyPKClustering*) for p-sensitive k-anonymity. For both l-diversity and (α, k)-anonymity the authors proposed to use adapted versions of Incognito as a first alternative [14, 22]. For (α, k)-anonymity a second algorithm based on local-recoding, called Top Down, was also presented [22]. Incognito and Top Down can be adapted for p-sensitive k-anonymity as well (in fact, we used such an adapted version of Incognito in our experiments for comparison purposes). The new anonymization algorithm will take advantage of the known properties of the p-sensitive k-anonymity model in order to improve the p-sensitive k-anonymous solutions w.r.t. various cost measures.

In the next two subsections we formally describe our approach to the anonymization problem, we present several cost measures, and we introduce our anonymization algorithm.

3.1 Problem Description

The microdata p-sensitive k-anonymization problem can be formulated as follows:

Definition 7 (*p-sensitive k-anonymization problem*): Given a microdata IM, the ***p-sensitive k-anonymization problem*** for IM is to find a partition $S = \{cl_1, cl_2, \ldots, cl_v\}$ of IM, where $cl_j \subseteq IM$, j=1..v, are called clusters and: $\bigcup_{j=1}^{v} cl_j = IM$; $cl_i \bigcap cl_j = \varnothing$, i, j = 1..v, $i \neq j$; $|cl_j| \geq k$ and cl_j is p-sensitive, j=1..v ; and a cost measure is optimized.

Once a solution S to the above problem is found for a microdata IM, a masked microdata MM that is p-sensitive and k-anonymous is formed by generalizing the quasi-identifier attributes of all tuples inside each cluster of S to the same values. The generalization method consists in replacing the actual value of an attribute with a less specific, more general value that is faithful to the original [17].

For categorical attributes we use generalization based on predefined hierarchies [9]. For numerical attributes we use the hierarchy-free generalization [12], which consists in replacing the set of values to be generalized to the smallest interval that includes all the initial values. For instance, the values: 25, 39, 36 are generalized to

the interval [25-39]. It is worth noting that the values for sensitive attributes remain unchanged within each cluster.

The anonymization of the initial microdata must be conducted to preserve data usefulness and to minimize information loss. In order to achieve this goal, we generalize each cluster to the least general tuple that represents all tuples in that group. We call *generalization information* for a cluster the minimal covering tuple for that cluster, and we define it as follows.

Definition 8 (*generalization information*): Let $cl = \{r_1, r_2, \ldots, r_q\} \in \mathcal{S}$ be a cluster, $\mathcal{KN} = \{N_1, N_2, ..., N_s\}$ be the set of numerical quasi-identifier attributes and $\mathcal{KC} = \{C_1, C_2,,\ldots, C_t\}$ be the set of categorical quasi-identifier attributes. The ***generalization information of cl***, w.r.t. quasi-identifier attribute set $\mathcal{K} = \mathcal{KN} \cup \mathcal{KC}$ is the "tuple" $gen(cl)$, having the scheme $\mathcal{K}$, where:

- For each categorical attribute $C_j \in \mathcal{K}$, $gen(cl)[C_j]$ = the lowest common ancestor in $\mathcal{H}_{Cj}$ of $\{r_1[C_j], r_2[C_j], \ldots, r_q[C_j]\}$, where $\mathcal{H}_C$ denotes the hierarchies (domain and value) associated to the categorical quasi-identifier attribute C;
- For each numerical attribute $N_j \in \mathcal{K}$, $gen(cl)[N_j]$ = the interval $[\min\{r_1[N_j], r_2[N_j], \ldots, r_q[N_j]\}, \max\{r_1[N_j], r_2[N_j], \ldots, r_q[N_j]\}]$.

For a cluster cl, its generalization information $gen(cl)$ is the tuple having as value for each quasi-identifier attribute, numerical or categorical, the most specific common generalized value for all that attribute values from cl tuples. In $\mathcal{MM}$, each tuple from the cluster cl will be replaced by $gen(cl)$.

There are several possible cost measures that can be used as optimization criterion for the p-sensitive k-anonymization problem [3, 4, etc.]. A simple cost measure is based on the size of each cluster from $\mathcal{S}$. This measure, called *discernability metric* (*DM*) [3] assigns to each record x from $\mathcal{IM}$ a penalty that is determined by the size of the cluster containing x:

$$DM(\mathcal{S}) = \sum_{j=1}^{v} (|cl_j|)^2 . \quad (2)$$

LeFevre introduced an alternative measure, called the *normalized average cluster size metric* (*AVG*) [12]:

$$AVG(\mathcal{S}) = \frac{n}{v \cdot k} , \quad (3)$$

where n is the size of the $\mathcal{IM}$, v is the number of clusters, and k is as in k-anonymity.

It is easy to notice that the *AVG* cost measure is inversely proportional with the number of clusters, and minimizing *AVG* is equivalent to maximizing the total number of clusters.

The last cost measure we present is the information loss caused by generalizing each cluster to a common tuple [4, 20]. This is an obvious measure to guide the partitioning process, since the produced partition $\mathcal{S}$ will subsequently be subject to cluster-level generalization.

Definition 9 (*cluster information loss*): Let $cl \in \mathcal{S}$ be a cluster, $gen(cl)$ its generalization information and $\mathcal{K} = \{N_1, N_2, .., N_s, C_1, C_2, .., C_t\}$ the set of quasi-

identifier attributes. The ***cluster information loss*** caused by generalizing cl tuples to $gen(cl)$ is:

$$IL(cl) = |cl| \cdot \left(\sum_{j=1}^{s} \frac{size(gen(cl)[N_j])}{size\left(\left[\min_{r \in IM} r[N_j],\ \max_{r \in IM} r[N_j]\right]\right)} + \sum_{j=1}^{t} \frac{height(\Lambda(gen(cl)[C_j]))}{height(H_{C_j})} \right), \quad (4)$$

where:

- $|cl|$ denotes the cluster cl cardinality;
- $size([i_1, i_2])$ is the size of the interval $[i_1, i_2]$ (the value i_2-i_1);
- $\Lambda(w)$, $w \in H_{Cj}$ is the subhierarchy of H_{Cj} rooted in w;
- $height(H_{Cj})$ denotes the height of the tree hierarchy H_{Cj}.

Definition 10 (***total information loss***): ***Total information loss for*** a solution $\mathcal{S} = \{cl_1, cl_2, \ldots, cl_v\}$ of the p-sensitive k-anonymization problem, denoted by $IL(\mathcal{S})$, is the sum of the information loss measure for all the clusters in $\mathcal{S}$:

$$IL(\mathcal{S}) = \sum_{j=1}^{v} (IL(cl_j)\ . \quad (5)$$

In order to achieve p-sensitive k-anonymity for each cluster, we need to address the p-sensitive part with uttermost attention. While the k-anonymity is satisfied for each individual cluster when its size is k or more, the p-sensitive property is not so obvious to achieve. To help us in this process we introduce two diversity measures that quantify, with respect to sensitive attributes, the diversity between a tuple and a cluster and the homogeneity of a cluster.

Let X^i, $i = 1 \ldots n$, be the tuples from IM subject to p-sensitive k-anonymization. We denote an individual tuple by $X^i = \{k_1^i, \ldots, k_q^i, s_1^i, \ldots, s_r^i\}$, where k^i s are the values for the quasi-identifier attributes and s^i s are the values for the confidential attributes.

Definition 11 (***diversity between a tuple and a cluster***): The ***diversity between a tuple X^i and a cluster cl*** w.r.t. the confidential attributes is given by:

$$Div(X^i, cl) = \sum_{i=1}^{r} (y_i' - y_i) \cdot (p - y_i) \cdot w_i \text{, where} \quad (6)$$

- y_i – is the number of distinct values for attribute S_i ($1 \le i \le r$) in cl if this number is less than p, and p otherwise.
- y_i' – is the number of distinct values for attribute S_i ($1 \le i \le r$) in $cl' = cl \cup \{X^i\}$ if this number is less than p, and p otherwise. It is easy to show that for each $i = 1 \ldots r$, y_i' is either y_i or $y_i + 1$.
- $(w_1, w_2, \ldots, w_r)$ – is a weight vector, $\sum_{l=1}^{r} w_l = 1$. The data owner can choose different criteria to define this weights vector. One possible selection of the

weight values is to initialize them as inversely proportional to the number of distinct sensitive attribute values in the microdata IM (defined as s_i values). In the experimental section we chose to use the same value for all the weights.

Definition 12 (*cluster homogeneity*): The ***homogeneity of a cluster cl*** w.r.t. the confidential attributes is given by:

$$Hom(cl) = \sum_{i=1}^{r} (p - y_i) \cdot w_i \,, \tag{7}$$

where y_i and w_i have the same meaning as in the previous definition.

Property 1: A cluster cl is p-sensitive w.r.t. all confidential attributes $S_1, S_2, \ldots, S_r$ iif $Hom(cl)=0$.

Proof: This property follows directly from the definition of cluster homogeneity.

3.2 The *EnhancedPKClustering* Algorithm

First, we introduce two total order relations that will help us present our algorithm.

Definition 13 ($\geq_h$ *relation*): Let S_i and S_j be two sensitive attributes. The following relation $S_i \geq_h S_j$ is true if and only if $maxClusters_i \leq maxClusters_j$ where $maxClusters_l$ is computed for IM with only one sensitive attribute S_l, $l = i, j$, given p and k. We use the term ***S_i is harder than or as hard as S_j to make sensitive*** for $S_i \geq_h S_j$.

Definition 14 ($\geq_d$ *relation*): Let cl_i and cl_j be two clusters. The following relation $cl_i \geq_d cl_j$ is true if and only if $Hom(cl_i) \leq Hom(cl_j)$, for a given p. We use the term ***cl_i is more diverse than or as diverse as cl_j*** for $cl_i \geq_d cl_j$.

Property 2: Let $maxClusters$ be as defined in Section 2.2. Let S_1 harder than or as hard as every other confidential attribute to make sensitive as defined in Definition 13. Let $iVal$ be the smallest value between 1 and p such that $maxClusters = \left\lfloor \frac{n - cf_{p-iVal}}{iVal} \right\rfloor$. The relation $|SEC_i^1| \leq maxClusters$ holds for all $i \geq p\text{-}iVal+1$ for which SEC_i^1 are defined.

Proof: From the definition of sensitive equivalence classes, the larger the value of i the smaller the cardinality of SEC's; therefore, it is enough to prove that $|SEC_{p-iVal+1}^1| \leq maxClusters$ holds.

From $maxClusters$ definition and the selection of $iVal$ we have:

$$maxClusters = \left\lfloor \frac{n - cf_{p-iVal}}{iVal} \right\rfloor < \left\lfloor \frac{n - cf_{p-iVal+1}}{iVal - 1} \right\rfloor \tag{8}$$

As S_1 is the hardest to make sensitive attribute and from definition of cumulative frequencies it follows that:

$$cf_{p-iVal+1} \geq cf_{p-iVal+1}^1 = cf_{p-iVal}^1 + |SEC_{p-iVal+1}^1| = cf_{p-iVal} + |SEC_{p-iVal+1}^1| \tag{9}$$

From (8) and (9) the following relation holds:

$$\left\lfloor \frac{n - cf_{p-iVal}}{iVal} \right\rfloor < \left\lfloor \frac{n - (cf_{p-iVal} + |\, SEC^1_{p-iVal+1} \,|)}{iVal - 1} \right\rfloor \quad \textbf{(10)}$$

Assuming $|\, SEC^1_{p-iVal+1} \,| > maxClusters \Rightarrow$

$$|\, SEC^1_{p-iVal+1} \,| > \frac{n - cf_{p-iVal}}{iVal} \quad \textbf{(11)}$$

Using relations (10) and (11) we obtain**:**

$$\left\lfloor \frac{n - cf_{p-iVal}}{iVal} \right\rfloor < \left\lfloor \left(n - \left(cf_{p-iVal} + \frac{n - cf_{p-iVal}}{iVal} \right) \right) \Big/ (iVal - 1) \right\rfloor = \left\lfloor \frac{n - cf_{p-iVal}}{iVal} \right\rfloor. \quad \textbf{(12)}$$

As a result, our assumption is false and the property $|\, SEC^1_i \,| \leq maxClusters$ holds for all $i \geq p\text{-}iVal+1$. Q.E.D.

The *EnhancedPKClustering* algorithm finds a solution for the *p*-sensitive *k*-anonymization problem for a given $\mathcal{IM}$. It considers *AVG* (or the partition cardinality) that has to be maximized as the cost measure.

This algorithm starts by enforcing the *p*-sensitive part using the properties proved for the *p*-sensitive *k*-anonymity model. The tuples from $\mathcal{IM}$ are distributed to form *p*-sensitive clusters with respect to the sensitive attributes. After *p*-sensitivity is achieved, the clusters are further processed to satisfy k-anonymity requirement as well. A more detailed description of how the algorithm proceeds follows.

In the beginning, the algorithm determines the *p*-sensitive equivalence classes, orders the attributes based on the harder to make sensitive relation, and computes the value *iValue* that divides the *p*-sensitive equivalence classes into two categories: one with less frequent values for the hardest to anonymize attribute and one with more frequent values. Now, the *QI*-clusters are created using the following steps:

- First, the tuples in the less frequent category of *p*-sensitive equivalence classes are divided into *maxClusters* clusters (*Split* function) such that each cluster will have *iValue* tuples with *iValue* distinct values within each cluster for attribute S_1 (the hardest to anonymize).
- Second, the remaining *p*-sensitive equivalence classes are used to fill the clusters such that each of them will have exactly *p* tuples with *p* distinct values for S_1.
- Third, the tuples not yet assigned to any cluster are used to add diversity for all remaining sensitive attributes until all clusters are *p*-sensitive. If no tuples are available, some of the less diverse (more homogenous) clusters are removed and their tuples are reused for the remaining clusters. At the end of this step all clusters are *p*-sensitive.
- Fourth, the tuples not yet assigned to any cluster are used to increase the size of each cluster to *k*. If no tuples are available, some of the less populated clusters are removed and their tuples are reused for the remaining clusters. At the end of this step all clusters are *p*-sensitive *k*-anonymous.

Along all the steps, when a choice is to be made, one or more optimization criteria are used (diversity between a tuple and a cluster, and increase in information loss).

Algorithm ***EnhancedPKClustering*** is
Input IM – initial microdata;
p, k – as in p-sensitive k-anonymity;
Output $S = \{cl_1, cl_2, \ldots, cl_v\}$ – a solution for the p-sensitive k-anonymization problem for IM;

Reorder S_1, S_2, …, S_r such that $S_i \geq_h S_j$, i, $j = 1..v$, $i > j$;

$maxClusters = \min_{i=1,p} \left\lfloor \frac{n - cf_{p-i}}{i} \right\rfloor$;

$iValue = \min \left\{ i \mid maxClusters = \left\lfloor \frac{n - cf_{p-i}}{i} \right\rfloor, i = 1..p \right\}$;

for i = 1 to $maxClusters$ do $cl_i = \emptyset$;
$S = \{cl_1, cl_2, \ldots, cl_{maxClusters}\}$;
$U = \{pSEC^1_{p-iValue+1}, pSEC^1_{p-iValue+2}, \ldots, pSEC^1_p\}$;
// Based on Condition 2, the tuples in U can be allocated to
// $maxClusters$ clusters, each having $iValue$ different values for S_1
Split (U, S, E);
for $j = p - iValue$ down to 1 {
  auxSEC = $pSEC^1_j$; aux$S$ = S;
  while (aux$S \neq \emptyset$) {
    ($tuple$, cl) = BestMatch($auxSEC$, aux$S$); // maximize diversity
    cl = cl $\cup$ {tuple};
    auxSEC = auxSEC – {tuple};
    aux$S$ = aux$S$ – {cl};
  } // end while
} // end for.
// Now p-sensitive property holds w.r.t. S_1

// T contains leftover tuples from $pSEC$'s plus tuples from E.
Let T be the set of tuples not assigned yet to any cluster from S.
Reorder clusters from S, such that $cl_i \geq_d cl_j$, $i, j = 1..maxClusters$, $i>j$;
h = 1;
while (Hom(cl_h) == 0) h= h + 1;
//cl_h the first cluster without p-sensitivity
aux = $maxClusters$;
while (h ≤ aux) {
  while (h ≤ aux) && (T ≠ ∅) {
    ($tuple$, cl_h) = BestMatch(T, {cl_h});
    $cl_h = cl_h \cup \{tuple\}$; $T = T - \{tuple\}$;
    if (Hom(cl_h) == 0) h = h + 1;
  }
  if (T == ∅) && ($h \leq aux$) {
    $T = cl_{aux}$;
    $aux = aux - 1$; // redistribute T
  }
}

// p-sensitivity property holds for all clusters.
// the set T (possible empty) must be spread.
Reorder S based on the number of tuples in each cluster($|cl_i| \geq |cl_j|$, $i, j = 1..aux$, $i > j$;)
u = 1;

```
while (|cl_u| ≥ k) u = u + 1;
// cl_i with i > u are not k-anonymous.
v = min(aux, u + ⌊(|T| + |cl_{u+1}| + ... + |cl_aux|) / k⌋);
if (v < aux) T = T ∪ {t ∈ cl_i | i = v + 1 ,.., aux};
for i = 1 to totalClusters do {
  while (|cl_i| < k) {
    Find a tuple such that IL(cl_i ∪ {tuple}) = min{IL(cl_i∪{t})| t ∈ T);
    cl_i = cl_i ∪ {tuple};
    T = T - {tuple};
  }
} // p-sensitive k-anonymity is achieved

for every t ∈ T do {   // extra tuples left in T are distributed
  Find cl such that IL(cl ∪ {t}) - IL(cl)
     = min(IL(cl_i∪{t}) - IL(cl_i)| i = 1,..,v);
  cl = cl ∪ {t};
}
End EnhancedPKClustering;

Function Split(U, S, E)
  U = {pSEC^1_{p-iValue+1}, ..., pSEC^1_p} = {SEC^1_{p-iValue+1}, . . , SEC^1_p, SEC^1_{p+1}, . . , SEC^1_{s_1}};
  i = 1;
  for j = s_1 down to p - iValue + 1 do {
    auxSEC = SEC^1_j;
    // tuples are assigned to clusters in a circular way; any two tuples
    // from the same auxSEC will belong to distinct clusters. (Prop. 2)
    while (auxSEC ≠ ∅) {
      (t, cl_i) = BestMatch(auxSEC, {cl_i});
      auxSEC = auxSEC - {t};
      cl_i = cl_i ∪ {t};
      i = i + 1;
      if (i > |S|) then
        if (|cl_1| < iValue) then i = 1
        else {
           // each cluster has iValue tuples
           E = all tuples in U not assigned; return; }
    }
  }
End Split;

Function BestMatch(auxSEC, auxS)
  Find the set Pairs of all pairs (t_i, cl_j) such that Div(t_i,cl_j) =
    max{Div(t,cl) | (t,cl) ∈ auxSEC × auxS};   // maximize diversity
  Return any pair (t,cl) ∈ Pairs such that IL(cl ∪ {t})-IL(cl) =
    min{IL(cl_j ∪ {t_i})-IL(cl_j)| (t_i,cl_j) ∈ Pairs};  // minimize IL
End BestMatch;
```

Informally, we state that the complexity of the *EnhancedPKClustering* algorithm is $O(n^2)$. A complete complexity analysis of the algorithm will be presented in the full version of the paper.

4 Preliminary results

In this section we report the experiments we have conducted to compare, for the *p*-sensitive *k*-anonymity model, the performance of *EnhancedPKClustering* algorithm against: an adapted version of *Incognito* algorithm [11] and the *GreedyPKClustering* algorithm [6]. We intend to extend our experiments and perform comparative tests with other algorithms proposed to enforce models equivalent with *p*-sensitive *k*-anonymity (*l*-diversity, (α, *k*)-anonymity, and *t*-closeness). However, we think that an algorithm based on global recoding will produce weaker results (in terms of any cost measure) compared to a local recoding algorithm (such as *EnhancedPKClustering* or *GreedyPKClustering*), and this without connection to a specific anonymity model.

All three algorithms have been implemented in Java, and tests were executed on a dual CPU machine running Windows 2003 Server with 3.00 GHz and 1 GB of RAM.

A set of experiments has been conducted for an IM consisting in 10000 tuples randomly selected from the *Adult* dataset from the UC Irvine Machine Learning Repository [15]. In all the experiments, we considered *age, workclass, marital-status, race*, *sex*, and *native-country* as the set of quasi-identifier attributes; and *education_num*, *education*, and *occupation* as the set of confidential attributes. Microdata *p*-sensitive *k*-anonymity was enforced in respect to the quasi-identifier consisting of all 6 quasi-identifier attributes and all 3 confidential attributes. Although many values of *k* and *p* were considered, due to space limitations, we present in this paper only a small subset of the results.

Fig. 3 shows comparatively the AVG and DM values of the three algorithms, *EnhancedPKClustering*, *GreedyPKClustering* and *Incognito*, produced for $k = 20$ and different *p* values. As expected, the results for the first two algorithms clearly outperform Incognito results. We notice that *EnhancedPKClustering* is able to improve the performances of the *GreedyPKClustering* algorithm in cases where solving the *p*-sensitivity part takes prevalence over creating clusters of size *k*.

Fig. 4 left shows comparatively the DM and AVG values obtained by *EnhancedPKClustering* algorithm divided by the same values computed using *GreedyPKClustering* algorithm. We notice that for $p = 2$ and 4 there is no improvement. In these cases both algorithms were able to find the optimal solution in terms of DM and AVG values. As soon as the *p*-sensitive part is hard to achieve, the *EnhancedPKClustering* algorithm performs better. Fig. 4 right shows the time required to generate the masked microdata by all three algorithms. Since *Incognito* uses global recording and our domain generalization hierarchies for this dataset have a low height, the running time is very fast. The *GreedyPKClustering* is faster than the new algorithm for small values of p, but when it is mode difficult to create p-sensitivity within each cluster the *EnhancedPKClustering* has a slight advantage. Based on these results, it is worth noting that a combination of GreedyPKClustering (for low values of *p*, in our experiment 2 and 4) and *EnhancedPKClustering* (for high values of *p*, in our experiment 6, 8, and 10) would be desirable in order to improve both running time and the selected cost measure (AVG or DM).

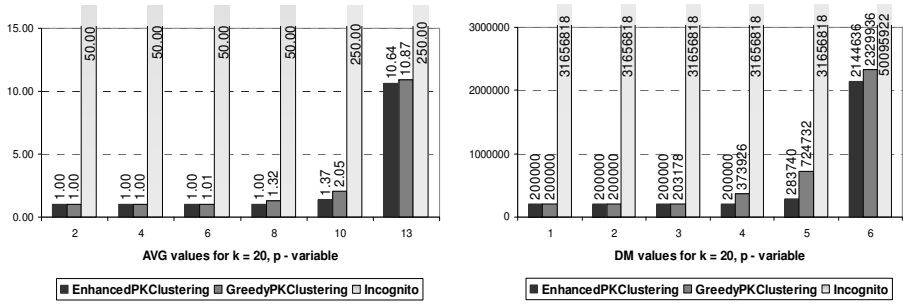

Fig. 3. AVG and DM for *EnhancedPKClustering*, *GreedyPKClustering*, and *Incognito*.

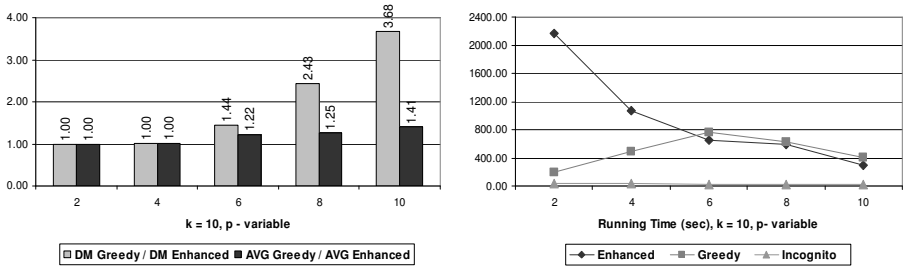

Fig. 4. Comparison between *EnhancedPKClustering* and *GreedyPKClustering* in terms DM and AVG values and the running time of all three algorithms..

5 Conclusions and future work

In this paper, a new algorithm to generate masked microdata with *p*-sensitive *k*-anonymity property was introduced. The algorithm uses several properties of the *p*-sensitive *k*-anonymity model in order to efficiently create the masked microdata that satisfy the privacy requirement. Our experiments have shown that the proposed algorithm improves both AVG and DM cost measures over existing algorithms. As our algorithm is based on local recoding (cluster-level generalization) and accepts multiple sensitive attributes, it leads to better results than the *Incognito* algorithm, but it also outperforms the local recoding based *GreedyPKClustering* algorithm. Two diversity measures that help characterize this similarity of sensitive attributes values within each cluster are also introduced.

We believe that the *EnhancedPKClustering* algorithm could be used for enforcing (*α, k*)-anonymity, *l*-diversity, or the new introduced *t*-closeness on microdata as well.

Acknowledgments. This work was supported by the Kentucky NSF EPSCoR Program under grant "*p*-Sensitive *k*-Anonymity Property for Microdata".

References

1. Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D., and Zhu A.: Anonymizing Tables. In Proceedings of the ICDT (2005) 246 – 258
2. Agrawal, R., Kiernan, J., Srikant, R., and Xu, Y.: Hippocratic Databases. In Proceedings of the VLDB (2002) 143-154
3. Bayardo, R.J, Agrawal, R.: Data Privacy through Optimal k-Anonymization. In Proceedings of the IEEE ICDE (2005) 217 – 228
4. Byun, J.W., Kamra, A., Bertino, E, Li, N.: Efficient k-Anonymity using Clustering Technique. CERIAS Tech Report 2006-10 (2006)
5. Campan, A., Truta, T.M.: Extended P-Sensitive K-Anonymity, Studia Universitatis Babes-Bolyai Informatica, Vol. 51, No. 2 (2006) 19 – 30
6. Campan, A., Truta, T.M., Miller, J., Sinca, R. A: Clustering Approach for Achieving Data Privacy, In Proceedings of the International Data Mining Conference (2007)
7. HIPAA.: Health Insurance Portability and Accountability Act. Available online at http://www.hhs.gov/ocr/hipaa (2002)
8. ICD9.: International Classification of Diseases. Available online at http://icd9cm.chrisendres.com/index.php
9. Iyengar, V.: Transforming Data to Satisfy Privacy Constraints. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2002) 279 – 288
10. Lambert, D.: Measures of Disclosure Risk and Harm. Journal of Official Statistics, Vol. 9 (1993) 313 – 331
11. LeFevre, K., DeWitt, D., and Ramakrishnan, R.: Incognito: Efficient Full-Domain K-Anonymity. In Proceedings of the ACM SIGMOD, (2005) 49 – 60
12. LeFevre, K., DeWitt, D., and Ramakrishnan, R.: Mondrian Multidimensional K-Anonymity. In Proceedings of the IEEE ICDE (2006) 25
13. Li, N., Li T., Venkatasubramanian, S.: T-Closeness: Privacy Beyond k-Anonymity and l-Diversity, In Proceedings of the IEEE ICDE (2007)
14. Machanavajjhala, A., Gehrke, J., Kifer, D.: L-Diversity: Privacy beyond K-Anonymity. In Proceedings of the IEEE ICDE (2006) 24
15. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases. online at www.ics.uci.edu/~mlearn/MLRepository.html, UC Irvine, (1998)
16. Samarati, P.: Protecting Respondents Identities in Microdata Release. IEEE Transactions on Knowledge and Data Engineering, Vol. 13, No. 6 (2001) 1010 – 1027
17. Sweeney, L.: k-Anonymity: A Model for Protecting Privacy. International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems, Vol. 10, No. 5 (2002) 557 – 570
18. Sweeney, L.: Achieving k-Anonymity Privacy Protection Using Generalization and Suppression. International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems, Vol. 10, No. 5 (2002) 571 – 588
19. Truta, T.M., Bindu, V.: Privacy Protection: P-Sensitive K-Anonymity Property. In Proceedings of the Workshop on Privacy Data Management, In Conjunction with IEEE ICDE (2006) 94
20. Truta, T.M., Campan, A.: K-Anonymization Incremental Maintenance and Optimization Techniques. In Proceedings of the ACM SAC (2007) 380 – 387
21. Winkler, W.: Matching and Record Linkage. In Business Survey Methods, Wiley (1995)
22. Wong, R.C-W., Li, J., Fu, A. W-C., Wang, K.: (α, k)-Anonymity: An Enhanced k-Anonymity Model for Privacy-Preserving Data Publishing. In Proceedings of the ACM KDD (2006) 754 – 759
23. Xiao, X., Tao, Y.: Personalized Privacy Preservation. In Proceedings of the ACM SIGMOD (2006) 229 – 240