On-the-Fly Generalization Hierarchies for Numerical Attributes Revisited

Alina Campan, Nicholas Cooper, and Traian Marius Truta

Department of Computer Science, Northern Kentucky University,
Highland Heights, KY 41099, USA
{campana1,trutat1}@nku.edu coopern1@mymail.nku.edu

Abstract. Generalization hierarchies are frequently used in computer science, statistics, biology, bioinformatics, and other areas when less specific values are needed for data analysis. Generalization is also one of the most used disclosure control technique for anonymizing data. For numerical attributes, generalization is performed either by using existing predefined generalization hierarchies or a hierarchy-free model. Because hierarchy-free generalization is not suitable for anonymization in all possible scenarios, generalization hierarchies are of particular interest for data anonymization. Traditionally, these hierarchies were created by the data owner with help from the domain experts. But while it is feasible to construct a hierarchy of small size, the effort increases for hierarchies that have many levels. Therefore, new approaches of creating these numerical hierarchies involve their automatic/on-the-fly generation. In this paper we extend an existing method for creating on-the-fly generalization hierarchies, we present several existing information loss measures used to assess the quality of anonymized data, and we run a series of experiments that show that our new method improves over existing methods to automatically generate on-the-fly numerical generalization hierarchies.

Keywords: anonymization, k-anonymity, hierarchies for quasi-identifier numerical attributes.

1 Introduction and Motivation

Generalization hierarchies are frequently used in computer science, statistics, biology, bioinformatics, and other areas when less specific values than the original ones are needed for data analysis. The term generalization hierarchy is used in data privacy and anonymity community and more recently in data mining community. Generalization hierarchies are commonly called taxonomies (biology, bioinformatics, statistics, etc.) or concept hierarchies (data mining and data warehousing).

These hierarchies provide the foundation of roll-up and roll-down operations in a data warehousing system [13]. In data mining, generalization hierarchies are used in various data mining techniques such as characteristic rule mining [9] classification [19], association rule mining [11], and clustering [5, 7]. Other areas of computer science such as machine learning [20], data integration [34], object-oriented databases [12], and intrusion detection [18] also use generalization hierarchies. Recently,

W. Jonker and M. Petković (Eds.): SDM 2011, LNCS 6933, pp. 18–32, 2011.

generalization hierarchies received a renewed attention in the data privacy field. Statistical disclosure control community used global/local recoding (a close substitute of a generalization hierarchy) as a disclosure control technique for protecting datasets against de-identification [41]. In the data anonymity community, the seminal papers of Sweeney [35] and Samarati [33] reinforced the use of generalization as a powerful and useful technique to achieve *k*-anonymity [33, 35].

Generalization consists in replacing the actual value of an attribute with a less specific, more general value that is faithful to the original [36]. In general, generalization is based on a *domain generalization hierarchy* (*DGH*) associated to that attribute. Such a generalization hierarchy is usually provided by a domain expert based on the attribute characteristics. A second hierarchy, called *value generalization hierarchy* (*VGH*), represents all values from different domains/levels of the domain generalization hierarchy and their ancestor/descendant relationships. Fig. 1 shows two examples of DGHs and VGHs for attributes *country* and *gender*.

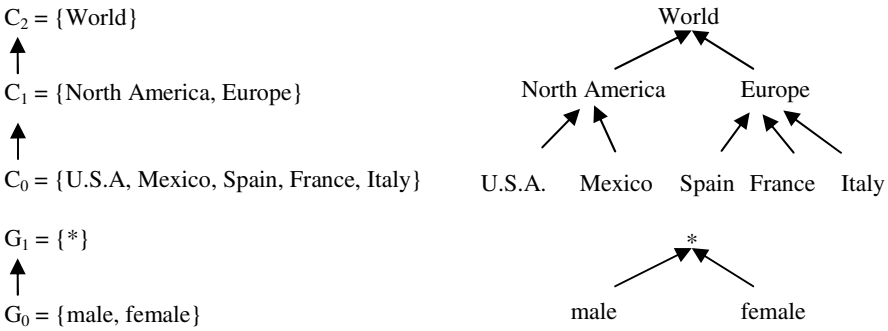

Fig. 1. DGHs and VGHs for attributes *country* and *gender*

Generalization is one of the most used disclosure control technique for anonymizing data. It is applied to microdata sets in order to avoid de-identification of individuals. *Microdata* represents a series of tuple, each tuple containing information on an individual unit such as a person or organization [41]. We call the original microdata initial microdata (IM). Due to existing regulations in various areas (such as Health Insurance Portability and Accountability Act, HIPAA [14]), IM should be released for use by a third party only after the owner of the data has masked it to limit the possibility of disclosure. We call the final microdata ready for release, the masked microdata (MM).

Generalization was initially used for *categorical* attributes with *predefined* DGHs and VGHs constructed by the domain experts [36]. Generalization was next extended for *numerical* attributes either by using *predefined hierarchies* [16] or a *hierarchy-free model* [23]. While generalization of numerical attributes using predefined hierarchies is similar to the generalization for categorical attributes, the generalization of numerical attributes without generalization hierarchies is based on determining generalization intervals/bins during the anonymization process based on an optimization criterion (such as minimizing information loss). Based on how

generalization intervals are created, that hierarchy-free generalization for numerical attributes helps minimizing the information loss that occurs in the masking process, and might perform in that respect better than using a predefined hierarchy. Still, there are situations when hierarchy-free generalization is not suitable for anonymization. First, creating generalization intervals during the anonymization process does not guarantee that those intervals are disjoint (in many situations these intervals will overlap) and this will create difficulties in analyzing the resulting masked microdata. For example, the values 12, 17, 23 for the attribute *age* can be generalized to the interval [12 – 23], and the values 16, 34 for the same attribute can be generalized to [16 – 34]. The reason why the grouping is not based on the order of values (the first group in that case would be 12, 16, 17; the second group 23, 34) is because there are other attributes involved in the anonymization process and the values of those attributes will impact the creation of groups; due to their influence, the overlapping generalization intervals for the numerical attribute are preferred for a smaller overall information loss in the anonymized data. If these overlaps are not desired in the resulting masked microdata, due to the nature of the application, then the data owner should use hierarchies during the generalization process. Second, certain data anonymity models, such as constrained *k*-anonymity (which relies on boundaries imposed on the amount of generalization allowed in the anonymization process) [30] and personalized anonymity (which uses guarding nodes as boundaries for the sensitive information) [39] require pre-existing hierarchies for numerical attributes.

Based on the above considerations, we conclude that there are situations when using hierarchies for numerical attributes during the anonymization process cannot be avoided. Traditionally, the generalization hierarchies were created by the data owner with help from the domain experts. But while it is feasible to construct a hierarchy of small size, the effort increases for large hierarchies. The manual construction of a generalization hierarchy might cause problems such as erroneous classifications or omissions of concepts [17]. Usually, creating and understanding a hierarchy for categorical attributes is easier than for numerical attributes: the values of the categorical attribute are well established, discrete, have a natural hierarchical organization, while numerical attributes have many values and not very often have a natural hierarchical structure. Moreover, a domain expert will not be able to capture the data characteristics when designing a generalization hierarchy, and this will likely lead to creating masked microdata where the information loss is high. Using a "good" hierarchy in the anonymization process significantly impacts the quality of the anonymized microdata; depending on how well the hierarchy fits the distribution and grouping of the attribute's values in the microdata set. Using on-the-fly hierarchies created based on data characteristics will help in creating a better-quality masked microdata.

Automatic generation methods for creating generalization hierarchies in data mining community (usually called concept hierarchies) exist for both categorical and numerical values. There are only a few studies for categorical attributes since, as mentioned before, these hierarchies are in general easier to create by human experts [22]. For numerical hierarchies, many techniques to generate automatic hierarchies are proposed in the literature. The binning method, which partitions numbers in equal ranges or equal frequencies, is reviewed in [13]. Extensions to this method include histogram analysis and numeric clustering [10, 13]. Other approaches that try to

locate better cutting points are based on recursive binary discretization [4], minimum description length [6], entropy-based discretization [32], chi-square test [21, 26], relaxation error [5], and attribute-oriented induction [15]. All these methods focus on preprocessing data before applying data mining techniques, and they are not tailored to data anonymity. Still, we selected two such approaches introduced by Han and Fu [10] and by Chu and Chiang [5] for our experimental comparison.

A method to generate on-the-fly numerical hierarchies for anonymizing data is introduced in [3]. A hierarchical clustering agglomerative approach [37] is used to construct a hierarchy based on the distance between already created nodes in the generalization hierarchy of the target attribute.

Our research contributions in this paper are as follows.

First, we improve the existing method for creating on-the-fly hierarchies for numerical attributes introduced in [3]. Our new method will replace the agglomerative selection approach based on minimal distance between nodes with the selection of two neighbor nodes that, combined, will create the smallest possible node (in a sense that we will describe later) at that step. This improved method is presented in Section 2.

Second, we discuss how the generated hierarchies are used during anonymization and we present several existing information loss measures that assess the information lost in the generalization of numerical quasi-identifier attribute values.

Third, we perform a series of experiments on the Adult dataset [17]. We generate *k*-anonymous masked microdata sets using the on-the-fly generalization hierarchies created based on our new method, using the existing method presented in [3], using a set of predefined hierarchies, and without using hierarchies (hierarchy-free generalization). We also create anonymized datasets using hierarchies generated with two existing methods used for dynamic generation of numerical hierarchies in data mining [5, 10]. The quality of the resulting datasets is compared with respect to the information loss measures discussed in Section 3. These information loss measures' values are dependent on the hierarchies used to perform generalization and on the anonymization algorithm used. To compare the quality of generalization hierarchies, we use the same anonymization algorithm (introduced in [2]) for all our generated datasets.

The paper ends with conclusions and suggestions for future work.

2 On-the-Fly Hierarchies for Numerical Attributes

The initial microdata (IM) is described by a set of attributes that are classified into three categories: *identifier* attributes such as *Name* and *SSN* that can be used to identify a tuple; *quasi-identifier* attributes such as *ZipCode* and *Sex* that may be known by an intruder; and *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 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. To avoid this possibility of disclosure, one frequently used

solution is to further process (modify) the initial microdata through generalization and suppression [36] 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.

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 protects only against identity disclosure and it fails to protect confidential information against attribute disclosure [29, 38]. As a result, several anonymity models were introduced to increase the protection of confidential information of individuals in the released datasets. Some of the most known extensions of *k*-anonymity include *l*-diversity [29], *p*-sensitive *k*-anonymity [38], (α, *k*)-anonymity [42], *t*-closeness [25], (ε, *m*)-anonymity [24], l^+-diversity [27], and (τ, λ)-uniqueness [40].

Generalization is one of the most used techniques to create a masked microdata that satisfies not only *k*-anonymity but also any of the improved anonymization models. For a fair comparison of the quality of generated masked microdata sets with various generalization hierarchies, the same anonymization model must be used. In this paper we decided to use *k*-anonymity for our comparison. While a different anonymization model may increase the information loss (due to a stronger privacy requirement, the utility is expected to drop), we expect that the information loss for various generalization hierarchies will keep for other models the relative proportion they have for *k*-anonymity.

Let *K* be the numerical quasi-identifier attribute for which we construct a generalization hierarchy. We denote by $V = \{v_1, v_2, \ldots, v_m\}$ the distinct values of *K* in the dataset $\mathcal{IM}$. Each one of these values can have one or more occurrences in $\mathcal{IM}$. If more than one numerical quasi-identifier attribute needs on-the-fly hierarchies, they are constructed individually, one attribute at a time.

The method to create on-the-fly hierarchies is described next. The construction of the hierarchy starts with a set of *m* nodes, one node for each of the *m* unique values of the attribute *K*. These nodes will become the leaves of the domain value hierarchy labeled $\mathcal{H}_K$ for the attribute *K*. Next, the hierarchy is built from leaves to root, by merging at each step two nodes that will create the smallest possible node at that step. The generalization hierarchy is completely built when all values are combined into a single node, the root of the hierarchy. The resulting hierarchy is a tree, called a dendrogram [13], which is usually not balanced, and which can have its leaves on any level under the root.

We will define next the size of a node and how two nodes are merged in our approach.

Definition 3. *(a node in the numerical hierarchy).* Each node in $\mathcal{H}_K$, leaf or internal, is characterized by two values: the *minimum* (*min*) and *maximum* (*max*) numerical values represented by the node.

For a leaf node created for the value *v*, *min* and *max* are the same value (*v*). A node will be represented as $X = [min, max]$. We will denote by *v* both a value of *K* and its associated leaf node.

Definition 4. *(size of a node).* We compute the size of a node $X = [min, max]$ as $size(X) = max - min$.

Definition 5. *(adjacent nodes).* During the construction of a hierarchy, two nodes $X_i = [min^i, max^i]$ and $X_j = [min^j, max^j]$ are called adjacent if they do not have yet any ancestors (in other words these nodes were not yet used in merging) and there is no value from *K* between the two nodes (in other words the interval $(min(max^i, max^j), max(min^i, min^j))$ does not contain any value from *K*).

Definition 6. *(merge two nodes).* Two adjacent nodes $X_i = [min^i, max^i]$ and $X_j = [min^j, max^j]$ are merged into a new node $Y = merge(X_i, X_j) = [min(min^i, min^j), max(max^i, max^j)]$.Both X_i and X_j are made descendants of *Y* when merged. The nodes X_i and X_j are selected such that the resulting node (*Y*) will have the smallest possible size at that time.

We give next the pseudocode for the generalization algorithm for constructing a numerical attribute's hierarchy.

```
Algorithm Improved On-The-Fly Hierarchy (IOTF) is
 Input: IM, attribute K
 Output: H_K
 Extract from IM the leaf nodes in H_K,
 V = {v_1, v_2, …, v_m};
 each v_i ∈ V has v_i.min = v_i.max = value v_i;
 H_K = V;
 Repeat
  Find X_i, X_j ∈ V such that
   X_i, X_j are adjacent and // see Definition 5
   ∀ X,Y∈V, size(merge(X_i, X_j)) ≤ size(merge(X, Y))
   // In other words, size(merge(X_i, X_j)) is minimized
   // Merge two adjacent nodes that create the smallest new node
   X_new = merge(X_i, X_j);
   Make X_new parent in H_K for X_i and X_j;
   V = V – {X_i, X_j} ∪ {X_new};
 Until (|V| = 1);
 The remaining node in V is the root of H_K;
End On-The-Fly Hierarchy.
```

In the above algorithm, the size of the current set of nodes, *V*, is reduced by one when two nodes are merged, and after *m*-1 iterations, only one node will remain in the set. This node becomes the root of the hierarchy. The hierarchies produced by this algorithm are shaped as binary trees and can be very deep, due to how they are

created – they can actually have a maximum of m-1 levels. In is worth noting that at every iteration, the nodes from the current set of nodes are completely disjoint. In the generated hierarchy any initial value has a unique path from its corresponding leaf to the root. This prevents one problem that exists with hierarchy-free generalization (described in Section 1). Examples of hierarchies constructed with this algorithm are presented in Section 4.

The complexity of the *NumericalHierarchy* algorithm is $O(m^2)$. This is because, in each merging step, the two nodes to be merged can only be adjacent nodes in the list of current nodes V. The nodes in V are kept sorted based on their *max* value (any value from the node can be used in this ordering since the nodes are disjoint). Consequently, finding the pair of nodes in V that when merged create the smallest node implies comparing $|V|$-1 pairs of nodes. As the size of V decreases from m to 1, the overall cost is $O(\sum_{l=1}^{m-1} l) = O(m^2)$.

3 Information Loss Measures Used in Data Anonymity

To measure the quality of masked microdata we use and adapt several well known information loss (*IL*) / data utility measures. Since our on-the-fly generalization is applicable to numerical attributes only, we present in this section these information loss measures with the assumption that all quasi-identifier attributes are numerical. We exclusively limit quasi-identifiers to homogeneous combinations of numerical attributes, with or without hierarchies, to isolate and study the impact on masked microdata quality of using different types of numerical hierarchies in the anonymization process.

We use the following notations in this section:

- $QI = \{K_1, K_2, \ldots, K_p\}$ – the set of p numerical quasi-identifiers for the initial microdata, IM.
- s – the number of quasi-identifier attributes for which we use hierarchy-free generalization. We agree that these attributes are the first s in the set QI ($\{K_1, K_2, \ldots, K_s\}$). Consequently, the set $\{K_{s+1}, K_{s+2}, \ldots, K_p\}$ represent the quasi-identifier attributes that are generalized using hierarchies. Note that when $s = 0$, all quasi-identifier attributes have hierarchies and when $s = p$ all attributes are generalized using hierarchy-free generalization.
- n – the number of tuples from IM.
- $cl = \{t_1, t_2, \ldots, t_q\}$ – a set of q tuples from IM.
- $\mathcal{S} = \{cl_1, cl_2, \ldots, cl_u\}$ – a complete and disjoint partition of IM (every tuple from IM belongs to exactly one cluster from the partition).
- $t_r|_{QI} = (t_r^1, t_r^2, \ldots, t_r^p)$, for all $r = 1..q$; $t_r|_{QI}$ denotes the relational projection operation of a tuple t_r on the set of attributes QI.
- $[min^k(cl), max^k(cl)] = [min(t_1^k, t_2^k, \ldots, t_q^k), max(t_1^k, t_2^k, \ldots, t_q^k)]$ for all $k = 1..p$. This interval represents the generalization interval of the cluster cl for the attribute K_k when hierarchy-free generalization is used.
- $\mathcal{H}_{Kk}$ – the generalization hierarchy of the attribute K_k.
- $root(\mathcal{H}_{Kk})$ – the root node of $\mathcal{H}_{Kk}$.
- $anc^k(cl)$ – the generalization node in $\mathcal{H}_{Kk}$ for the cluster cl. This node is the first

common ancestor for all values form the cluster *cl* with respect to the attribute K_k. This node represents the interval [$anc^k(cl).min$, $anc^k(cl).max$] (see Definition 3). We also use $size(anc^k(cl)) = anc^k(cl).max - anc^k(cl).min$ as per Definition 4.

To achieve *k*-anonymity, $\mathcal{IM}$ is partitioned into clusters of size at least *k*. Each such cluster is generalized to the corresponding *QI*-cluster using either hierarchy-free generalization or hierarchy (predefined or on-the-fly)-based generalization for each quasi-identifier attribute. This process will lead to loss of information in $\mathcal{MM}$ compared to $\mathcal{IM}$.

The first information loss measure we present in Definitions 7 and 8 was previously presented in [3] and it extends the measure previously introduced in [2] by assessing the information loss in hierarchies where leaf nodes are situated at different levels.

Definition 7. *(cluster information loss due to generalization).* The information loss caused by generalizing a cluster *cl* to the same "tuple" (these tuples form a *QI*-cluster in $\mathcal{MM}$), denoted by *IL*(*cl*), is defined as follows:

$$IL(cl) = |cl| \times \left[\sum_{k=1}^{s} \frac{max^k(cl) - min^k(cl)}{max^k(IM) - min^k(IM)} + \sum_{k=s+1}^{p} \frac{size(anc^k(cl))}{size(root(H_{Kk}))} \right]$$

Definition 8. *(normalized total information loss).* The ***normalized total information loss*** for a partition into clusters, $\mathcal{S}$, of the initial microdata set, $\mathcal{IM}$, is the sum of the information loss for all clusters in $\mathcal{S}$ divided to the number of tuples from $\mathcal{IM}$ times the number of quasi-identifier attributes. Formally:

$$NTIL(IM, S) = \frac{\sum_{j=1}^{u} IL(cl_j)}{n \cdot p},$$

The maximum value for *NTIL* is 1, and it corresponds to the case when all tuples in $\mathcal{IM}$ would have each quasi-identifier attribute generalized to the interval that covers all of its values in the set, or, respectively, generalized to the root value of its value generalization hierarchy. The minimum value (0) is obtained when $\mathcal{MM}$ is the same as $\mathcal{IM}$ (there was no generalization performed).

The next two information loss measures presented in Definitions 9 and 10 are based on Minkowski-norms on group extents and they are introduced in [8].

Definition 9. *(normalized information loss – average-extent metric).* The ***normalized information loss based on average extent metric*** for a partition into clusters, $\mathcal{S}$, of the initial microdata set, $\mathcal{IM}$, is defined as follows:

$$NIL_1(IM, S) = \left[\sum_{j=1}^{u} \left(\sum_{k=1}^{s} \frac{max^k(cl_j) - min^k(cl_j)}{max^k(IM) - min^k(IM)} + \sum_{k=s+1}^{p} \frac{size\left(anc^k(cl_j)\right)}{size\left(root(H_{Kk})\right)} \right) \right] \Big/ p \cdot u$$

NIL_1 is similar to *NTIL* except it does not take into account the size of clusters from the partition $\mathcal{S}$. The range of values for NIL_1 is [0, 1], and the boundaries are also met

for no generalization ($NIL_1 = 0$) and generalization to the root ($NIL_1 = 1$), respectively.

Definition 10. *(normalized information loss – maximum-extent metric).* The ***normalized information loss based on maximum extent metric*** for a partition into clusters, $\mathcal{S}$, of the initial microdata set, $\mathcal{IM}$, is defined as follows:

$$NIL_{\infty}(IM,S) = \left[\sum_{k=1}^{s} \max_{j=1,u}\left(\frac{max^k(cl_j) - min^k(cl_j)}{max^k(IM) - min^k(IM)}\right) + \sum_{k=s+1}^{p} \max_{j=1,u}\left(\frac{size\left(anc^k(cl_j)\right)}{size\left(root(H_{Kk})\right)}\right)\right] \Big/ p$$

NIL_{∞} is considering the maximum information loss per attribute between all clusters which is averaged for all quasi-identifier attributes and normalized to [0, 1]. While the value 0 is also obtained when there is no generalization, the value 1 can be obtained more easily, for instance it is enough if only a cluster is generalized to the root (or maximum interval, for hierarchy-free generalization) for all attributes. This value can also be obtained if for any quasi-identifier attribute, there is a cluster that generalizes that attribute to the root.

The last two information loss measures we present in Definitions 11 and 12 are based on *discernability metric* (*DM*) [1] and *average cluster size metric* (*AVG*) [23]. These measures are not normalized to [0, 1].

Definition 11. *(discernability metric).* The ***discernability metric (DM)*** assigns to each tuple from $\mathcal{IM}$ a penalty that is determined by the size of the cluster containing that tuple:

$$DM(IM,S) = \sum_{j=1}^{u} \left(|cl_j|\right)^2$$

Definition 12. *(average cluster size metric).* The ***average cluster size metric (AVG)*** is defined as follows:

$$NAVG(IM,S) = \frac{n}{u \cdot k}$$

4 Experimental Results

For our experiments, we selected the anonymization algorithm called *greedy k-member clustering* presented in [2]. This algorithm works by creating clusters of tuples from $\mathcal{IM}$, of size k or more. These clusters will be then generalized to the same tuple, forming a *QI*-cluster in the $\mathcal{MM}$. The clusters are created one at a time, starting from a seed tuple and absorbing one new tuple at a time, until the cluster has k tuples. The new tuple selection criterion is based on an objective function. The objective function in our case is the *NTIL* function; therefore, a new tuple is added to a cluster labeled *cl* if it produces a local minimum increase of *IL*(*cl*) (see Definition 7).

To assess the performance of the new proposed on-the-fly generalization method of hierarchies for numerical attributes, we used the Adult dataset [31]. This dataset is the de-facto benchmark for many data anonymization problems and it consists of 45,422 tuples. Since we want to compare the generalization hierarchies for numerical

attributes we will restrict our experiments to the following 3 numerical quasi-identifier attributes: *age*, *education_num*, *hours_per_week*. As we already mentioned before, we considered all of the quasi-identifiers to be numerical, as to avoid the categorical ones to impact in any way the anonymization process and the quality of the masked microdata.

We performed experiments with six settings for the above mentioned quasi-identifier attribute set:

- Each attribute had a generalization hierarchy dynamically created with the method introduced in this paper. We refer to it as *IOTF* (improved on-the-fly) method.
- Each attribute had a generalization hierarchy dynamically created with the related method introduced in [3]. We call this method *OTF* (on-the-fly) method.
- Each attribute had *predefined* hierarchies. These hierarchies are the same as in [3].
- Each attributes did not have hierarchies (i.e. hierarchy-free generalization).
- Each attribute had a generalization hierarchy dynamically created with a method used in data mining for concept hierarchies introduced in [10]. In the algorithm to generate hierarchies from [10] we use a threshold value of 4 and a fan-out value of 5. We call this method *Han* based on the first author name.
- Each attribute had a generalization hierarchy dynamically created with a method used in data mining for concept hierarchies introduced in [5]. In the algorithm to generate hierarchies from [5] we use a threshold value of 2. We use the first author's name, *Chu*, to refer to this method.

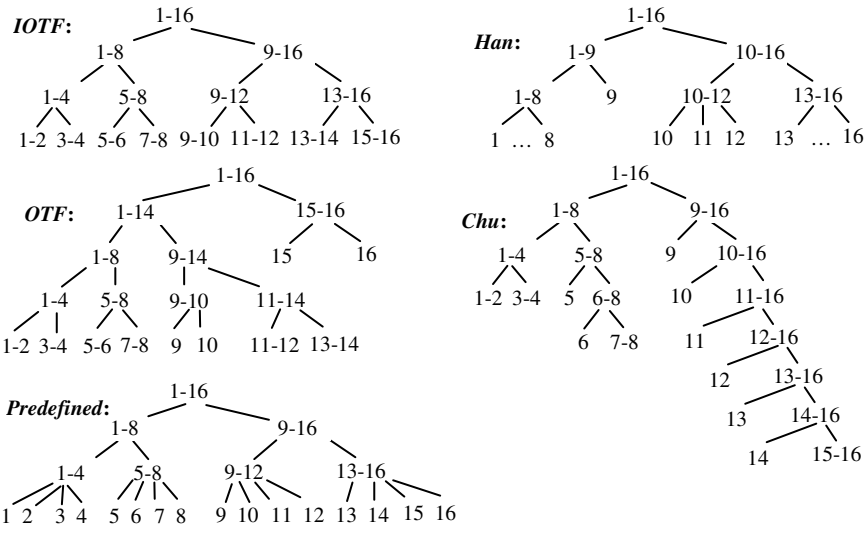

Fig. 2. VGHs for attribute *education_num* generated using *IOTF*, *OTF*, *Predefined*, *Han*, and *Chu* methods

We present in Fig. 2 the generated value generalization hierarchies for the attribute *education_num* (we selected this attribute as it has the smallest number of distinct values; similar hierarchies were generated for *age* and *hours_per_week* attributes using all five methods). In Fig. 2, due to space limitation some of the single value leafs are not shown.

In each setting, we anonymized the microdata set using the same algorithm ([2]), for all possible k values in the range 2 - 20. It is worth noting that in all experiments we either use hierarchies ($s = 0$) or a hierarchy-free approach ($s = p$) for all three quasi-identifier attributes (see Section 3 for definitions of s and p). For each experiment we computed all measures presented in Section 3: *NTIL*, NIL_1, NIL_∞, *DM*, and *AVG*.

Fig. 3 presents comparatively the normalized total information loss (*NTIL*) and normalized information loss based on average-extent metric (NIL_1) for all six cases, for the even values of k we considered in our experiments ($k = 2, .., 20$). It can be seen that the *IOTF* method of generating on-the-fly hierarchies outperform the other four methods based on generated or predefined hierarchies (*OTF*, *Predefined*, *Han*, and *Chu*) and as expected it does not perform as well as hierarchy-free generalization. However, as presented in Section 1, hierarchy-free generalization is not applicable in all anonymization scenarios. Out of the five generated or predefined hierarchy methods, *Han* and *Predefined* perform the worse because they do not use binary hierarchies, and therefore the generalization will create larger intervals faster than in the other methods. *Chu* and *OTF* methods produce results that are close to *IOTF*, however *IOTF* performed better with respect to *NTIL* and NIL_1 in all scenarios. The reason why *Chu* method performs reasonably well is because it uses a top down-approach in which intervals are split based on a measure (called relaxation error) that considers the value frequencies and the distance between values [5].

Fig. 4 presents comparatively the discernability metric (*DM*) and average cluster size (*AVG*) for all six cases, for even values of k considered in our experiments ($k = 2, .., 20$). The results are similar with the ones for *NTIL* and NIL_1 measures. Han and Predefined methods perform worse than the other methods, and as expected, hierarchy-free generalization performs the best. However in this case, there is almost a tie between the other three methods. For discernability metric values, out of 18 experiments ($k = 2, 3, \ldots, 20$), *IOTF* outperformed *OTF* and *Chu* 7 times, while *OTF* and *Chu* had the best result 6 times each. For average cluster size metric, *IOTF* had the best result 9 times, *OTF* 5 times, and *Chu* also 5 times. The reason why the proposed algorithm is not a clear winner for these two measures is because they do not consider the size of the created clusters, but only their number. As described in Section 2, the *IOTF* algorithm minimizes the size of newly created intervals, and this will contribute to smaller size clusters but not necessarily to fewer clusters.

We did not include a depiction with the results for NIL_∞ because, when using hierarchies, in almost all cases one cluster was generalized to the entire range for each attribute, and therefore the NIL_∞ measure is almost all the time 1. The only three cases (out of 95, five hierarchy-based methods and 19 values of k) when NIL_∞ was not equal to 1 are: ($k = 5$, *IOTF*), ($k = 6$, *IOTF*), and ($k = 3$, *Chu*). The reason why this measure is almost all the time 1 is the chosen anonymization algorithm. *K-member-clustering* [2] is a greedy algorithm that at the end will create large clusters (large not as number of members, but large with respect to our definition of size), and since NIL_∞ considers

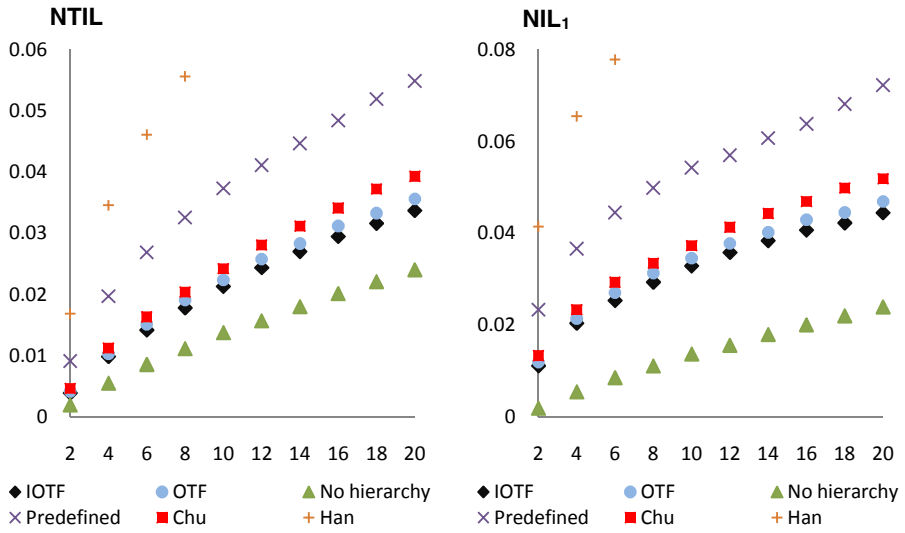

Fig. 3. *NTIL* and NIL_1 for k = 2, 4, .., 20 (even values) using five types of generalization hierarchies (*IOTF*, *OTF*, *Predefined*, *Chu*, and *Han*) and hierarchy-free generalization

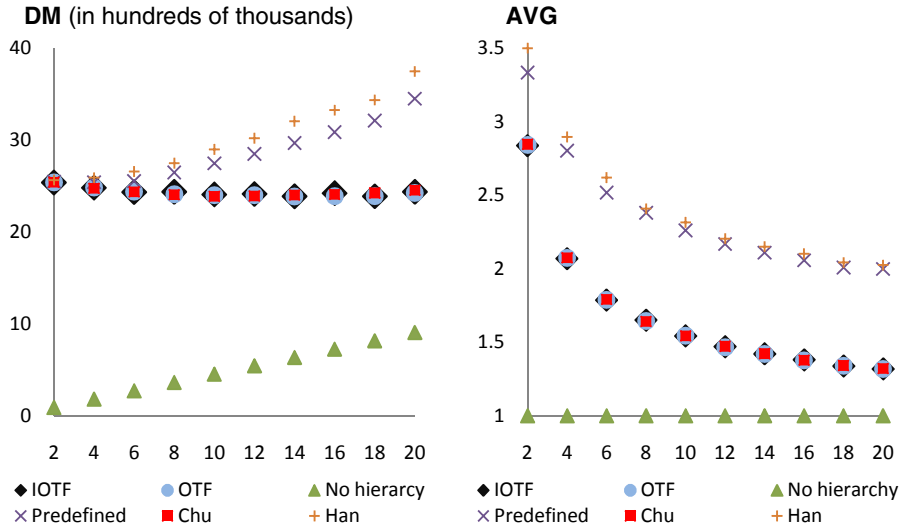

Fig. 4. *DM* and *AVG* for k = 2, 4, .., 20 (even values) using five types of generalization hierarchies (*IOTF*, *OTF*, *Predefined*, *Chu*, and *Han*) and hierarchy-free generalization

the maximum size intervals between those clusters, the results will always be 1 or close to 1. For the same reason, NIL_∞ measure was close to 1 (but not equal) for all 19 cases of hierarchy-based generalization.

5 Conclusions and Future Work

We introduced in this paper a new method for dynamically creating hierarchies for numerical quasi-identifier attributes. The resulting hierarchies represent a valid alternative to predefined hierarchies, and their usage generally results in good quality masked microdata, with reasonable information loss. Our new method clearly outperforms existing approaches to generate on-the-fly numerical hierarchies with respect to two information loss measures, normalized total information loss (*NTIL*) and normalized information loss based on average-extent metric (NIL_1). The proposed method had similar or slightly better results for the other three information loss measures, namely normalized information loss based on maximum-extent metric (NIL_∞), discernability metric (*DM*), and average cluster size (*AVG*), when compared with two other methods to create on-the fly hierarchies (*OTF* and *Chu*). On-the-fly hierarchies can be easily produced when hierarchies are necessary, instead of forcing the user to artificially develop ones that might not reflect the properties of the data, therefore negatively impacting the quality of the masked microdata.

As future work, we intend to investigate other anonymization algorithms that generate *k*-anonymous or *l*-diverse [29] masked microdata with respect to how well they perform using on-the-fly generalization hierarchies generated with the proposed method.

References

1. Bayardo, R.J., Agrawal, R.: Data Privacy through Optimal k-Anonymization. In: Proceedings of the IEEE International Conference of Data Engineering (ICDE), pp. 217–228 (2005)
2. Byun, J.W., Kamra, A., Bertino, E., Li, N.: Efficient k-Anonymity using Clustering Techniques. CERIAS Technical Report 2006-10 (2006)
3. Campan, A., Cooper, N.: On-the-Fly Hierarchies for Numerical Attributes in Data Anonymization. In: Jonker, W., Petković, M. (eds.) SDM 2010. LNCS, vol. 6358, pp. 13–25. Springer, Heidelberg (2010)
4. Catlett, J.: On Changing Continuous Attributes into Ordered Discrete Attributes. In: Kodratoff, Y. (ed.) EWSL 1991. LNCS, vol. 482, pp. 164–177. Springer, Heidelberg (1991)
5. Chu, W.W., Chiang, K.: Abstraction of High Level Concepts from Numerical Values in Databases. In: Proceedings of the Knowledge Discovery in Data Mining Workshop (KDD 1994), pp. 133–144 (1994)
6. Fayyad, U.M., Irani, K.B.: Multi-interval Discretization of Continuous-valued Attributes for Classification Learning. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 1022–1027 (1993)
7. Fischer, D.: Improving Inference through Conceptual Clustering. In: Proceedings of the National Conference on Artificial Intelligence (AAAI 1987), vol. 2, pp. 461–465 (1987)
8. Ghinita, G., Karras, P., Kalinis, K.: A Framework for Efficient Data Anonymization under Privacy and Accuracy Constraints. ACM transactions on database Systems 34(2) (2009)
9. Han, J., Cai, Y., Cercone, N.: Data-Driven Discovery of Quantitative Rules in Relational Databases. IEEE Transactions on Knowledge and Data Engineering 5(1), 29–40 (1993)

10. Han, J., Fu, Y.: Dynamic Generation and Refinement of Concept Hierarchies for Knowledge Discovery in Databases. In: Proceedings of the Knowledge Discovery in Data Mining Workshop (KDD 1994), pp. 157–168 (1994)
11. Han, J., Fu, Y.: Discovery of Multiple-level Association Rules from Large Databases. In: Proceedings of the Very Large Database Conference (VLDB 1995), pp. 420–431 (1995)
12. Han, J., Nishio, S., Kawano, H., Wang, W.: Generalization-based Data Mining in Object-oriented Databases using an Object Cube Model. Data and Knowledge Engineering 25, 55–97 (1998)
13. Han, J., Kamber, M.: Data Mining - Concepts and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2006)
14. HIPAA: Health Insurance Portability and Accountability Act (2002), http://www.hhs.gov/ocr/hipaa
15. Hsu, C.: Extending Attribute-oriented Induction Algorithm for Major Values and Numeric Values. Expert Systems with Applications 27, 187–202 (2004)
16. Iyengar, V.: Transforming Data to Satisfy Privacy Constraints. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 279–288 (2002)
17. Jing, Y., Croft, W.B.: An Association Thesaurus for Information Retrieval. Technical Report #94-17. University of Massachusetts at Amherst, Amherst, MA (1994)
18. Julisch, K.: Clustering Intrusion Detection Alarms to Support Root Cause Analysis. ACM Transactions on Information and System Security (TISSEC) 6(4), 443–471 (2003)
19. Kamber, M., Winstone, L., Gong, W., Cheng, S., Han, J.: Generalization and Decision Tree Induction: Efficient Classification in Data Mining. In: Proceedings of the International Workshop on Research Issues on Data Engineering (RIDE 1997), pp. 111–120 (1997)
20. Kaufman, K.A., Michalski, R.S.: A Method for Reasoning with Structured and Continuous Attributes in the INLEN-2 Multistrategy Knowledge Discovery System. In: Proceedings of the Knowledge Discovery in Data Mining Conference (KDD 1996), pp. 232–237 (1996)
21. Kerber, R.: Chimerge: Discretization of Numeric Attributes. In: Proceedings of the International Conference on Artificial Intelligence (AAAI 1992). MIT Press, Cambridge (1992)
22. Lee, S., Huh, S.Y., McNiel, R.D.: Automatic Generation of Concept Hierarchies using Wordnet. Expert Systems with Applications Journal 35(3), 1132–1144 (2008)
23. LeFevre, K., DeWitt, D., Ramakrishnan, R.: Mondrian Multidimensional *K*-Anonymity. In: Proceedings of the IEEE International Conference of Data Engineering, Atlanta, Georgia (2006)
24. Li, J., Tao, Y., Xiao, X.: Preservation of Proximity Privacy in Publishing Numerical Sensitive Data. In: Proceedings of the ACM SIGMOD, pp. 473–486 (2008)
25. Li, N., Li, T., Venkatasubramanian, S.: *T*-Closeness: Privacy Beyond *k*-Anonymity and *l*-Diversity. In: Proceedings of the 23rd International Conference on Data Engineering (IEEE ICDE 2007), pp. 106–115 (2007)
26. Liu, H., Setiono, R.: Feature Selection via Discretization. IEEE Transactions on Knowledge and Data Engineering 9(4), 642–645 (1997)
27. Liu, J.Q., Wang, K.: On Optimal Anonymization for *l*+-Diversity. In: Proceedings of the International Conference on Data Engineering, IEEE ICDE 2010 (2010)
28. Lunacek, M., Whitley, D., Ray, I.: A Crossover Operator for the *k*-Anonymity Problem. In: Proceedings of the GECCO Conference, pp. 1713–1720 (2006)

29. Machanavajjhala, A., Gehrke, J., Kifer, D.: *L*-Diversity: Privacy beyond *K*-Anonymity. In: Proceedings of the International Conference on Data Engineering (IEEE ICDE 2006), p. 24 (2006)
30. Miller, J., Campan, A., Truta, T.M.: Constrained *K*-Anonymity: Privacy with Generalization Boundaries. In: Proceedings of the Workshop on Practical Preserving Data Mining, with SIAM SDM 2008 (2008)
31. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases. UC Irvine (1998), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
32. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, Los Altos (1993)
33. Samarati, P.: Protecting Respondents Identities in Microdata Release. IEEE Transactions on Knowledge and Data Engineering 13(6), 1010–1027 (2001)
34. Schultz, S., Romacker, M., Faggioli, G., Hahn, U.: From Knowledge Import to Knowledge Finishing: Automatic Acquisition and Semi-Automatic Refinement of Medical Knowledge. In: Proceedings of the Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (1999)
35. Sweeney, L.: *k*-Anonymity: A Model for Protecting Privacy. International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems 10(5), 557–570 (2002)
36. Sweeney, L.: Achieving *k*-Anonymity Privacy Protection Using Generalization and Suppression. International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems 10(5), 571–588 (2002)
37. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison Wesley, Reading (2005)
38. Truta, T.M., Bindu, V.: Privacy Protection: *P*-Sensitive *K*-Anonymity Property. In: Proceedings of the Workshop on Privacy Data Management, with ICDE 2006, p. 94 (2006)
39. Wang, P.: Personalized Anonymity Algorithm Using Clustering Techniques. Journal of Computational Information Systems 7(3), 924–931 (2011)
40. Wei, Q., Lu, Y., Lou, Q.: (τ, λ)-Uniqueness: Anonymity Management for Data Publication. In: Proceedings of the IEEE International Conference on Computer and Information Science (2008)
41. Willemborg, L., Waal, T.: Elements of Statistical Disclosure Control. Springer, Heidelberg (2001)
42. 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 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD 2006), pp. 754–759 (2006)