Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression [*]

Pierangela Samarati[†]
Computer Science Laboratory
SRI International
Menlo Park, CA 94025, USA
samarati@csl.sri.com

Latanya Sweeney
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
sweeney@ai.mit.edu

Abstract

Today's globally networked society places great demand on the dissemination and sharing of person-specific data. Situations where aggregate statistical information was once the reporting norm now rely heavily on the transfer of microscopically detailed transaction and encounter information. This happens at a time when more and more historically public information is also electronically available. When these data are linked together, they provide an electronic shadow of a person or organization that is as identifying and personal as a fingerprint, even when the sources of the information contains no explicit identifiers, such as name and phone number. In order to protect the anonymity of individuals to whom released data refer, data holders often remove or encrypt explicit identifiers such as names, addresses and phone numbers. However, other distinctive data, which we term quasi-identifiers, *often combine uniquely and can be linked to publicly available information to re-identify individuals.*

In this paper we address the problem of releasing person-specific data while, at the same time, safeguarding the anonymity of the individuals to whom the data refer. The approach is based on the definition of k-anonymity. *A table provides k-anonymity if attempts to link explicitly identifying information to its contents ambiguously map the information to at least k entities. We illustrate how k-anonymity can be provided by using generalization and suppression techniques. We introduce the concept of* minimal *generalization, which captures the property of the release process not to distort the data more than needed to achieve k-anonymity. We illustrate possible preference policies to choose among different minimal generalizations. Finally, we present an algorithm and experimental results when an implementation of the algorithm was used to produce releases of real medical information. We also report on the quality of the released data by measuring the precision and completeness of the results for different values of k.*

[*] The work of Pierangela Samarati was supported in part DARPA/Rome Laboratory under grant F30602-96-C-0337 and by the National Science Foundation under grant ECS-94-22688. The work of Latanya Sweeney was supported by Medical Informatics Training Grant IT15 LM07092 from the National Library of Medicine.

[†] On leave from Università di Milano.

1 Introduction

In the age of the Internet and inexpensive computing power, society has developed an insatiable appetite for information, all kinds of information for many new and often exciting uses. Most actions in daily life are recorded on some computer somewhere. That information in turn is often shared, exchanged, and sold. Many people may not care that the local grocer keeps track of which items they purchase, but shared information can be quite sensitive or damaging to individuals and organizations. Improper disclosure of medical information, financial information or matters of national security can have alarming ramifications, and many abuses have been cited [2, 23]. The objective is to release information freely but to do so in a way that the identity of any individual contained in the data cannot be recognized. In this way, information can be shared freely and used for many new purposes.

Shockingly, there remains a common incorrect belief that if the data looks anonymous, it is anonymous. Data holders, including government agencies, often remove all explicit identifiers, such as name, address, and phone number, from data so that other information in the data can be shared, incorrectly believing that the identities of individuals cannot be determined. On the contrary, de-identifying data provides no guarantee of anonymity [18]. Released information often contains other data, such as birth date, gender, and ZIP code, that in combination can be linked to publicly available information to re-identify individuals. Most municipalities sell population registers that include the identities of individuals along with basic demographics; examples include local census data, voter lists, city directories, and information from motor vehicle agencies, tax assessors, and real estate agencies. For example, an electronic version of a city's voter list was purchased for twenty dollars and used to show the ease of re-identifying medical records [18]. In addition to names and addresses, the voter list included the birth dates and genders of 54,805 voters. Of these, 12% had unique birth dates, 29% were unique with respect to birth date and gender, 69% with respect to birth date and a 5-digit ZIP code, and, 97% were identifiable with just the full postal code and birth date [18]. These results reveal how uniquely identifying combinations of basic demographic attributes, such as ZIP code, date of birth, ethnicity, gender and martial status, can be.

To illustrate this problem, Figure 1 exemplifies a table of released medical data de-identified by suppressing names and Social Security Numbers (SSNs) so as not to disclose the identities of individuals to whom the data refer. However, values of other released attributes, such as {ZIP, Date Of Birth, Ethnicity, Sex, Marital Status} can also appear in some external table jointly with the individual identity, and can therefore allow it to be tracked. As illustrated in Figure 1, ZIP, Date Of Birth, and Sex can be linked to the Voter List to reveal the Name, Address, and City. Likewise, Ethnicity and Marital Status can be linked to other publicly available population registers. In the Medical Data table of Figure 1, there is only one female, born on 9/15/61 and living in the 02142 area. From the uniqueness results mentioned previously regarding an actual voter list, more than 69% of the 54,805 voters could be uniquely identified using just these attributes. This combination uniquely identifies the corresponding bulleted tuple in the released data as pertaining to "Sue J. Carlson, 1459 Main Street, Cambridge" and therefore reveals she has reported shortness of breath. (Notice the medical information is not assumed to be publicly associated with the individuals, and the desired protection is to release the medical information such that the identities of the individuals cannot be determined. However, the of the released characteristics for Sue J. Carlson leads to determine which medical data among those released are hers.) While this example demonstrated an exact match, in some cases, released information can be linked to a restrictive set of individuals to whom the released information could refer.

Several protection techniques have been developed with respect to statistical databases, such as scrambling and swapping values and adding noise to the data in such a way as to maintain an overall statistical property of the result [1, 21]. However, many new uses of data, including data mining, cost analysis and retrospective research, often need accurate information within the tuple itself. Two independently developed systems have been released which use suppression and generalization as techniques to provide disclosure con-

Medical Data Released as Anonymous

SSN	Name	Ethnicity	Date Of Birth	Sex	ZIP	Marital Status	Problem
		asian	09/27/64	female	02139	divorced	hypertension
		asian	09/30/64	female	02139	divorced	obesity
		asian	04/18/64	male	02139	married	chest pain
		asian	04/15/64	male	02139	married	obesity
		black	03/13/63	male	02138	married	hypertension
		black	03/18/63	male	02138	married	shortness of breath
		black	09/13/64	female	02141	married	shortness of breath
		black	09/07/64	female	02141	married	obesity
		white	05/14/61	male	02138	single	chest pain
		white	05/08/61	male	02138	single	obesity
		white	09/15/61	female	02142	widow	shortness of breath

Voter List

Name	Address	City	ZIP	DOB	Sex	Party	....
....	....	....	....	....	....	....	
....	....	....	....	....	....	....	
Sue J. Carlson	1459 Main St.	Cambridge	*02142*	*9/15/61*	*female*	democrat	....
....	....	....	....	....	....	....	

Figure 1: Re-identifying anonymous data by linking to external data

trol while maintaining the integrity of the values within each tuple - namely Datafly in the United States [17] and Mu-Argus [11] in Europe. However, no formal foundations or abstraction have been provided for the techniques employed by both. Further approximations made by the systems can suffer from drawbacks, such as generalizing data more than is needed, like [17], or not providing adequate protection, like [11].

In this paper we provide a formal foundation for the anonymity problem against linking and for the application of generalization and suppression towards its solution. We introduce the definition of *quasi-identifiers* as attributes that can be exploited for linking, and of *k-anonymity* as characterizing the degree of protection of data with respect to inference by linking. We show how k-anonymity can be ensured in information releases by generalizing and/or suppressing part of the data to be disclosed. Within this framework, we introduce the concepts of *generalized* table and of *minimal generalization*. Intuitively, a generalization is minimal if data are not generalized more than necessary to provide k-anonymity. Also, the definition of *preferred generalization* allows the user to select, among possible minimal generalizations, those that satisfy particular conditions, such as favoring certain attributes in the generalization process. We present an algorithm to compute a preferred minimal generalization of a given table. Finally, we discuss some experimental results derived from the application of our approach to a medical database containing information on 265 patients.

The problem we consider differs from the traditional access control [3] and from statistical database [1, 4, 8, 9, 12, 22] problems. Access control systems address the problem of controlling specific access to data with respect to rules stating whether a piece of data can or cannot be released. In our work it is not the disclosure of the specific piece of data to be protected (i.e., on which an access decision can be taken), but rather the fact that the data refers to a particular entity. Statistical database techniques address the problem of producing tabular data representing a summary of the information to be queried. Protection is enforced in such a framework by ensuring that it is not possible for users to infer original individual data from the produced summary. In our approach, instead, we allow the release of generalized person-specific data on which users can produce summaries according to their needs. The advantage with respect to precomputed release-specific statistics is an increased flexibility and availability of information for the users. This flexibility and availability has as a drawback, from the end-user stand point, a coarse granularity level of the data. This new type of declassification and release of information seems to be required more and more in today's emerging applications [18].

The remainder of this paper is organized as follows. In Section 2 we introduce basic assumptions and definitions. In Section 3 we discuss generalization to provide anonymity, and in Section 4 we continue the discussion to include suppression. In Section 5 basic preference policies for choosing among different minimal generalizations are illustrated. In Section 6 we discuss an algorithmic implementation of our approach. Section 7 reports some experimental results. Section 8 concludes the paper.

2 Assumptions and preliminary definitions

We consider the data holder's table to be a private table PT where each tuple refers to a different entity (individual, organization, and so on). From the private table PT, the data holder constructs a table which is to be an anonymous release of PT. For the sake of simplicity, we will subsequently refer to the privacy and re-identification of individuals in cases equally applicable to other entities. We assume that all explicit identifiers (e.g., names, SSNs, and addresses) are either encrypted or suppressed, and we therefore ignore them in the remainder of this paper. Borrowing the terminology from [6], we call the combination of characteristics on which linking can be enforced *quasi-identifiers*. Quasi-identifiers must therefore be protected. They are defined as follows.

Definition 2.1 (Quasi-identifier) *Let $T(A_1, \ldots, A_n)$ be a table. A quasi-identifier of T is a set of attributes $\{A_i, \ldots, A_j\} \subseteq \{A_1, \ldots, A_n\}$ whose release must be controlled.*

Given a table $T(A_1, \ldots, A_n)$, a subset of attributes $\{A_i, \ldots, A_j\} \subseteq \{A_1, \ldots, A_n\}$, and a tuple $t \in T$, $t[A_i, \ldots, A_j]$ denotes the sequence of the values of $A_i, \ldots, A_j$ in t, $T(A_i, \ldots, A_j)$ denotes the projection, maintaining duplicate tuples, of attributes $A_i, \ldots, A_j$ in T. Also, QI_T denotes the set of quasi-identifiers associated with T, and $|T|$ denotes cardinality, that is, the number of tuples in T.

Our goal is to allow the release of information in the table while ensuring the anonymity of the individuals. The anonymity constraint requires released information to indistinctly relate to at least a given number k of individuals, where k is typically set by the data holder, as stated by the following requirement.

Definition 2.2 (k-anonymity requirement) *Each release of data must be such that every combination of values of quasi-identifiers can be indistinctly matched to at least k individuals.*

Adherence to the anonymity requirement necessitates knowing how many individuals each released tuple matches. This can be done by explicitly linking the released data with externally available data. This is obviously an impossible task for the data holder. Although we can assume that the data holder knows which attributes may appear in external tables, and therefore what constitutes quasi-identifiers, the specific values of data in external knowledge cannot be assumed. The key to satisfying the k-anonymity requirement, therefore, is to *translate the requirement in terms of the released data* themselves. In order to do that, we require the following assumption to hold.

Assumption 2.1 *All attributes in table PT which are to be released and which are externally available in combination (i.e., appearing together in an external table or in possible joins between external tables)* [1] *to a data recipient are defined in a quasi-dentifier associated with PT.*

Although this is not a trivial assumption its enforcement is possible. The data holder estimates which attributes might be used to link with outside knowledge; this of course forms the basis for a quasi-identifier. While the expectation of this knowledge is somewhat reasonable for publicly available data, we recognize that there are far too many sources of semi public and private information such as pharmacy records, longitudinal

[1] A universal relation combining external tables can be imagined [20].

studies, financial records, survey responses, occupational lists, and membership lists, to account a priori for all linking possibilities [18]. Suppose the choice of attributes for a quasi-identifier is incorrect; that is, the data holder misjudges which attributes are sensitive for linking. In this case, the released data may be less anonymous than what was required, and as a result, individuals may be more easily identified. Sweeney [18] examines this risk and shows that it cannot be perfectly resolved by the data holder since the data holder cannot always know what each recipient of the data knows. [18] poses solutions that reside in policies, laws, and contracts. In the remainder of this work, we assume that proper quasi-identifiers have been recognized.

We introduce the definition of k-anonymity for a table as follows.

Definition 2.3 (k-anonymity) *Let $T(A_1, \ldots, A_n)$ be a table and QI_T be the quasi-identifiers associated with it. T is said to satisfy k-anonymity iff for each quasi-identifier $QI \in \mathsf{QI}_T$ each sequence of values in $T[QI]$ appears at least with k occurrences in $T[QI]$.*

Under Assumption 2.1, and under the hypothesis that the privately stored table contains at most one tuple for each identity to be protected (i.e., to whom a quasi-identifier refers), k-anonymity of a released table represents a sufficient condition for the satisfaction of the k-anonymity requirement. In other words, a table satisfying Definition 2.3 for a given, k satisfies the k-anonymity requirement for such a k. Consider a quasi-identifier QI; if Definition 2.3 is satisfied, each tuple in $\mathsf{PT}[QI]$ has at least k occurrences. Since the population of the private table is a subset of the population of the outside world, there will be at least k individuals in the outside world matching these values. Also, since all attributes available outside in combination are included in QI, no additional attributes can be joint to QI to reduce the cardinality of such a set. (Note also that any subset of the attributes in QI will refer to $k' > k$ individuals.) To illustrate, consider the situation exemplified in Figure 1 but assume that the released data contained two occurrences of the sequence white, 09/15/64, female, 02142, widow. Then *at least* two individuals matching such occurrences will exist in the voter list (or in the table combining the voter list with all other external tables), and it will not be possible for the data recipient to determine which of the two medical records associated with these values of the quasi-identifier belong to which of the two individuals. Since k-anonymity of 2 was provided in the release, each medical record could indistinctly belong to *at least* two individuals.

Given the assumption and definitions above, and given a private table PT to be released, we focus on the problem of producing a version of PT which satisfies k-anonymity.

3 Generalizing data

Our first approach to providing k-anonymity is based on the definition and use of generalization relationships between domains and between values that attributes can assume.

3.1 Generalization relationships

In a classical relational database system, domains are used to describe the set of values that attributes assume. For example, there might be a ZIP code domain, a *number* domain, and a *string* domain. We extend this notion of a domain to make it easier to describe how to *generalize* the values of an attribute. In the original database, where every value is as specific as possible, every attribute is in the *ground* domain. For example, 02139 is in the *ground* ZIP code domain, $\mathtt{Z}_0$. To achieve k-anonymity, we can make the ZIP code less informative. We do this by saying that there is a more general, less specific, domain that can be used to describe ZIP codes, $\mathtt{Z}_1$, in which the last digit has been replaced by a 0. There is also a mapping from $\mathtt{Z}_0$ to $\mathtt{Z}_1$, such as 02139 → 02130. This mapping between domains is stated by means of a generalization relationship, which represents a partial order $\leq_{\mathsf{D}}$ on the set $\mathtt{Dom}$ of domains, and which is required to satisfy the following conditions: *(1)* each domain D_i has at most one *direct* generalized domain, and *(2)* all

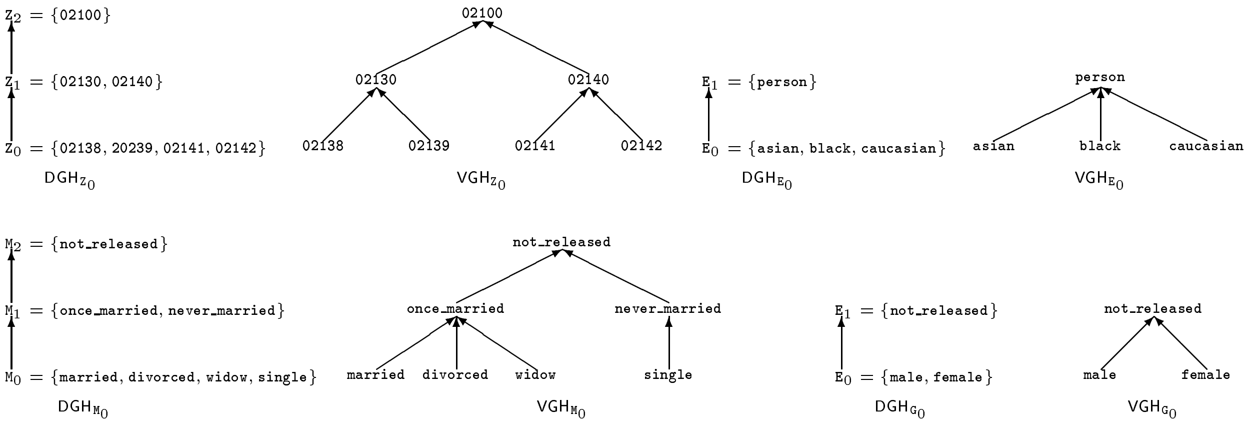

Figure 2: Examples of domain and value generalization hierarchies

maximal elements of Dom are singleton.[2] The definition of this generalization implies the existence, for each domain $D \in \mathsf{Dom}$, of a hierarchy, which we term the **domain generalization hierarchy** DGH_D. Since generalized values can be used in place of more specific ones, it is important that all domains in a hierarchy be compatible. Compatibility can be ensured by using the same storage representation form for all domains in a generalization hierarchy. A *value generalization relationship*, partial order $\leq_{\mathsf{V}}$, is also defined which associates with each value v_i in a domain D_i a *unique* value in domain D_j direct generalization of D_i. Such a relationship implies the existence, for each domain D, of a **value generalization hierarchy** VGH_D.

Example 3.1 *Figure 2 illustrates an example of domain and value generalization hierarchies for domain* $\mathtt{Z}_0$ *representing zip-codes of the Cambridge, MA, area,* $\mathtt{E}_0$ *representing ethnicities,* $\mathtt{M}_0$ *representing marital status, and* G_0 *representing gender.*

In the remainder of this paper we will often refer to a domain or value generalization hierarchy in terms of the graph representing all and only the *direct* generalization relationships between the elements in it (i.e., implied generalization relationships do not appear as arcs in the graph). We will use the term *hierarchy* interchangeably to denote either a partially ordered set or the graph representing the set and all the direct generalization relationships between its elements. We will explicitly refer to the ordered set or to the graph when it is not otherwise clear from context.

Also, since we will be dealing with sets of attributes, it is useful to visualize the generalization relationship and hierarchies in terms of tuples composed of elements of Dom or of their values. Given a tuple $DT = \langle D_1, \ldots, D_n \rangle$ such that $D_i \in \mathsf{Dom}$, $i = 1, \ldots, n$, we define the **domain generalization hierarchy** of DT as $\mathsf{DGH}_{DT} = \mathsf{DGH}_{D_1} \times \ldots \times \mathsf{DGH}_{D_n}$, assuming that the Cartesian product is ordered by imposing coordinate-wise order [7]. DGH_{DT} defines a lattice whose minimal element is DT. The generalization hierarchy of a domain tuple DT defines the different ways in which DT can be generalized. In particular, each path from DT to the unique maximal element of DGH_{DT} in the graph describing DGH_{DT} defines a possible alternative path that can be followed in the generalization process. We refer to the set of nodes in each of such paths together with the generalization relationships between them as a **generalization strategy** for DGH_{DT}. Figure 3 illustrates the domain generalization hierarchy $\mathsf{DGH}_{\mathtt{E}_0,\mathtt{Z}_0}$ where the domain generalization hierarchies of $\mathtt{E}_0$ and $\mathtt{Z}_0$ are as illustrated in Figure 2.

[2] The motivation behind condition 2 is to ensure that all values in each domain can be eventually generalized to a single value.

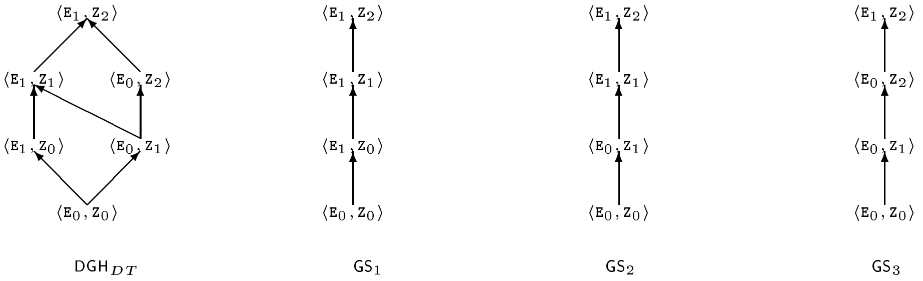

Figure 3: Domain generalization hierarchy DGH_{DT} and strategies for $DT = \langle \mathtt{E}_0, \mathtt{Z}_0 \rangle$

Eth:$\mathtt{E}_0$	**ZIP**:$\mathtt{Z}_0$
asian	02138
asian	02139
asian	02141
asian	02142
black	02138
black	02139
black	02141
black	02142
white	02138
white	02139
white	02141
white	02142

PT

Eth:$\mathtt{E}_1$	**ZIP**:$\mathtt{Z}_0$
person	02138
person	02139
person	02141
person	02142
person	02138
person	02139
person	02141
person	02142
person	02138
person	02139
person	02141
person	02142

$\mathsf{GT}_{[1,0]}$

Eth:$\mathtt{E}_1$	**ZIP**:$\mathtt{Z}_1$
person	02130
person	02130
person	02140
person	02140
person	02130
person	02130
person	02140
person	02140
person	02130
person	02130
person	02140
person	02140

$\mathsf{GT}_{[1,1]}$

Eth:$\mathtt{E}_0$	**ZIP**:$\mathtt{Z}_2$
asian	02100
asian	02100
asian	02100
asian	02100
black	02100
black	02100
black	02100
black	02100
white	02100
white	02100
white	02100
white	02100

$\mathsf{GT}_{[0,2]}$

Eth:$\mathtt{E}_0$	**ZIP**:$\mathtt{Z}_1$
asian	02130
asian	02130
asian	02140
asian	02140
black	02130
black	02130
black	02140
black	02140
white	02130
white	02130
white	02140
white	02140

$\mathsf{GT}_{[0,1]}$

Figure 4: Examples of generalized tables for PT

3.2 Generalized table and minimal generalization

Given a private table PT, our first approach to provide k-anonymity consists of generalizing the values stored in the table. Intuitively, attribute values stored in the private table can be substituted, upon release, with generalized values. Since multiple values can map to a single generalized value, generalization may decrease the number of distinct tuples, thereby possibly increasing the size of the clusters containing tuples with the same values. We perform generalization at the attribute level. Generalizing an attribute means substituting its values with corresponding values from a more general domain. Generalization at the attribute level ensures that all values of an attribute belong to the same domain. However, as a result of the generalization process, the domain of an attribute can change. In the following, $dom(A_i, T)$ denotes the domain of attribute A_i in table T. $D_i = dom(A_i, \mathsf{PT})$ denotes the domain associated with attribute A_i in the private table PT.

Definition 3.1 (Generalized Table) *Let $T_i(A_1, \ldots, A_n)$ and $T_j(A_1, \ldots, A_n)$ be two tables defined on the same set of attributes. T_j is said to be a generalization of T_i, written $T_i \preceq T_j$, iff*

1. $|T_i| = |T_j|$
2. $\forall z = 1, \ldots, n : dom(A_z, T_i) \leq_{\mathsf{D}} dom(A_z, T_j)$
3. *It is possible to define a bijective mapping between T_i and T_j that associates each tuples t_i and t_j such that $t_i[A_z] \leq_{\mathsf{V}} t_j[A_z]$.*

Definition 3.1 states that a table T_j is a generalization of a table T_i, defined on the same attributes, iff (*1*) T_i and T_j have the same number of tuples, (*2*) the domain of each attribute in T_j is equal to or a

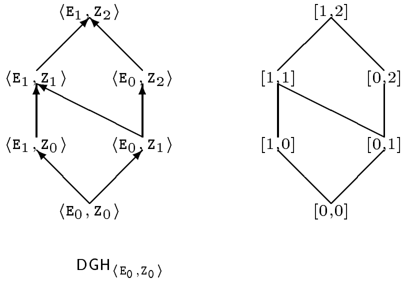

Figure 5: Hierarchy $\mathsf{DGH}_{\langle \mathtt{E}_0,\mathtt{Z}_0\rangle}$ and corresponding lattice on distance vectors

generalization of the domain of the attribute in T_i, and (*3*) each tuple t_i in T_i has a corresponding tuple t_j in T_j (and vice versa) such that the value for each attribute in t_j is equal to or a generalization of the value of the corresponding attribute in t_i.

Example 3.2 *Consider the table* PT *illustrated in Figure 4 and the domain and value generalization hierarchies for* $\mathtt{E}_0$ *and* $\mathtt{Z}_0$ *illustrated in Figure 2. The remaining four tables in Figure 4 are all possible generalized tables for* PT*, but the topmost one generalizes each tuple to* $\langle \mathtt{person}, \mathtt{02100}\rangle$*. For the clarity of the example, each table reports the domain for each attribute in the table. With respect to k-anonymity,* $\mathsf{GT}_{[0,1]}$ *satisfies k-anonymity for $k = 1, 2$;* $\mathsf{GT}_{[1,0]}$ *satisfies k-anonymity for $k = 1, 2, 3$;* $\mathsf{GT}_{[0,2]}$ *satisfies k-anonymity for $k = 1, \ldots, 4$, and* $\mathsf{GT}_{[1,1]}$ *satisfies k-anonymity for $k = 1, \ldots, 6$.*

Given a table, different possible generalizations exist. Not all generalizations, however, can be considered equally satisfactory. For instance, the trivial generalization bringing each attribute to the highest possible level of generalization, thus collapsing all tuples in T to the same list of values, provides k-anonymity at the price of a strong generalization of the data. Such extreme generalization is not needed if a more specific table (i.e., containing more specific values) exists which satisfies k-anonymity. This concept is captured by the definition of k-minimal generalization. To introduce it we first introduce the notion of distance vector.

Definition 3.2 (Distance vector) *Let $T_i(A_1, \ldots, A_n)$ and $T_j(A_1, \ldots, A_n)$ be two tables such that $T_i \preceq T_j$. The distance vector of T_j from T_i is the vector $\mathsf{DV}_{i,j} = [d_1, \ldots, d_n]$ where each d_z is the length of the* unique *path between $D =$dom(A_z, T_i) and dom(A_z, T_j) in the domain generalization hierarchy DGH_D.*

Example 3.3 *Consider table* PT *and its generalized tables illustrated in Figure 4. The distance vectors between* PT *and its different generalizations are the vectors appearing as a subscript of each table.*

Given two distance vectors $\mathsf{DV} = [d_1, \ldots, d_n]$ and $\mathsf{DV}' = [d'_1, \ldots, d'_n]$, $\mathsf{DV} \leq \mathsf{DV}'$ iff $d_i \leq d'_i$ for all $i = 1, \ldots, n$; $\mathsf{DV} < \mathsf{DV}'$ iff $\mathsf{DV} \leq \mathsf{DV}'$ and $\mathsf{DV} \neq \mathsf{DV}'$. A generalization hierarchy for a domain tuple can be seen as a hierarchy (lattice) on the corresponding distance vectors. For instance, Figure 5 illustrates the lattice representing the $\leq$ relationship between the distance vectors corresponding to the possible generalization of $\langle \mathtt{E}_0, \mathtt{Z}_0\rangle$.

We can now introduce the definition of k-minimal generalization.

Definition 3.3 (k-minimal generalization) *Let T_i and T_j be two tables such that $T_i \preceq T_j$. T_j is said to be a k-*minimal *generalization of T_i iff*

1. *T_j satisfies k-anonymity*

Ethn	DOB	Sex	ZIP	Status
asian	09/27/64	female	02139	divorced
asian	09/30/64	female	02139	divorced
asian	04/18/64	male	02139	married
asian	04/15/64	male	02139	married
black	03/13/63	male	02138	married
black	03/18/63	male	02138	married
black	09/13/64	female	02141	married
black	09/07/64	female	02141	married
white	05/14/61	male	02138	single
white	05/08/61	male	02138	single
white	09/15/61	female	02142	widow

PT

Ethn	DOB	Sex	ZIP	Status
asian	64	not_rel	02100	not_rel
asian	64	not_rel	02100	not_rel
asian	64	not_rel	02100	not_rel
asian	64	not_rel	02100	not_rel
black	63	not_rel	02100	not_rel
black	63	not_rel	02100	not_rel
black	64	not_rel	02100	not_rel
black	64	not_rel	02100	not_rel
white	61	not_rel	02100	not_rel
white	61	not_rel	02100	not_rel
white	61	not_rel	02100	not_rel

$\mathsf{GT}_{[0,2,1,2,2]}$

Ethn	DOB	Sex	ZIP	Status
pers	[60-65]	female	02130	been
pers	[60-65]	female	02130	been
pers	[60-65]	male	02130	been
pers	[60-65]	male	02130	been
pers	[60-65]	male	02130	been
pers	[60-65]	male	02130	been
pers	[60-65]	female	02140	been
pers	[60-65]	female	02140	been
pers	[60-65]	male	02130	never
pers	[60-65]	male	02130	never
pers	[60-65]	female	02140	been

$\mathsf{GT}_{[1,3,0,1,1]}$

Figure 6: An example of table PT and its minimal generalizations

2. $\nexists T_z : T_i \leq T_z, T_z$ *satisfies* k*-anonymity, and* $DV_{i,z} < DV_{i,j}$.

Intuitively, a generalization T_j is minimal iff there does not exist another generalization T_z satisfying k-anonymity which is dominated by T_j in the domain generalization hierarchy of $\langle D_1, \ldots, D_n \rangle$ (or, equivalently, in the corresponding lattice of distance vectors). If this were the case T_j would itself be a generalization for T_z. Note also that a table can be a minimal generalization of itself if the table already achieved k-anonymity.

Example 3.4 *Consider table* PT *and its generalized tables illustrated in Figure 4. Assume* $QI = (\texttt{Eth}, \texttt{ZIP})$ *to be a quasi-identifier. It is easy to see that for* $k = 2$ *there exist two* k*-minimal generalizations, which are* $\mathsf{GT}_{[1,0]}$ *and* $\mathsf{GT}_{[0,1]}$*. Table* $\mathsf{GT}_{[0,2]}$*, which satisfies the anonymity requirements, is not minimal since it is a generalization of* $\mathsf{GT}_{[0,1]}$*. Analogously* $\mathsf{GT}_{[1,1]}$ *cannot be minimal, being a generalization of both* $\mathsf{GT}_{[1,0]}$ *and* $\mathsf{GT}_{[0,1]}$*. There are also only two* k*-minimal generalized tables for k=3, which are* $\mathsf{GT}_{[1,0]}$ *and* $\mathsf{GT}_{[0,2]}$*.*

Note that since k-anonymity requires the existence of k-occurrences for each sequence of values only for quasi-identifiers, for every minimal generalization T_j, $DV_{i,j}[d_z] = 0$ for all attributes A_z which do not belong to any quasi-identifier.

4 Suppressing data

In Section 3 we discussed how, given a private table PT, a generalized table can be produced which releases a more general version of the data in PT and which satisfies a k-anonymity constraint. Generalization has the advantage of allowing release of all the single tuples in the table, although in a more general form. Here, we illustrate a *complementary* approach to providing k-anonymity, which is *suppression*. Suppressing means to remove data from the table so that they are not released and as a disclosure control technique is not new [5, 21]. We apply suppression at the tuple level, that is, a tuple can be suppressed only in its entirety. Suppression is used to "moderate" the generalization process when a limited number of outliers (that is,

Ethn	DOB	Sex	ZIP	Status
asian	09/27/64	female	02139	divorced
asian	09/30/64	female	02139	divorced
asian	04/18/64	male	02139	married
asian	04/15/64	male	02139	married
black	03/13/63	male	02138	married
black	03/18/63	male	02138	married
black	09/13/64	female	02141	married
black	09/07/64	female	02141	married
white	05/14/61	male	02138	single
white	05/08/61	male	02138	single

PT

Ethn	DOB	Sex	ZIP	Status
asian	64	female	02139	divorced
asian	64	female	02139	divorced
asian	64	male	02139	married
asian	64	male	02139	married
black	63	male	02138	married
black	63	male	02138	married
black	64	female	02141	married
black	64	female	02141	married
white	61	male	02138	single
white	61	male	02138	single

$\mathsf{GT}_{[0,2,0,0,0]}$

Figure 7: An example of table PT and its minimal generalization

Eth:E_0	ZIP:Z_0
asian	02138
asian	02138
asian	02142
asian	02142
black	**02138**
black	**02141**
black	**02142**
white	**02138**

PT

Eth:E_1	ZIP:Z_0
person	02138
person	02138
person	02142
person	02142
person	02138
person	**02141**
person	02142
person	02138

$\mathsf{GT}_{[1,0]}$

Eth:E_0	ZIP:Z_1
asian	02130
asian	02130
asian	02140
asian	02140
black	**02130**
black	02140
black	02140
white	**02130**

$\mathsf{GT}_{[0,1]}$

Eth:E_0	ZIP:Z_2
asian	02100
asian	02100
asian	02100
asian	02100
black	02100
black	02100
black	02100
white	**02100**

$\mathsf{GT}_{[0,2]}$

Eth:E_1	ZIP:Z_1
person	02130
person	02130
person	02140
person	02140
person	02130
person	02140
person	02140
person	02130

$\mathsf{GT}_{[1,1]}$

Figure 8: Examples of generalized tables for PT

tuples with less that k occurrences) would force a great amount of generalization. To clarify, consider the table illustrated in Figure 1, whose projection on the considered quasi-identifier is illustrated in Figure 6 and suppose k-anonymity with $k = 2$ is to be provided. Attribute `Date of Birth` has a domain date with the following generalizations: from the specific date (`mm/dd/yy`) to the month (`mm/yy`) to the year (`yy`) to a 5-year interval (e.g., `[60-64]`) to a 10-year interval (e.g., `[60,69]`) to a 25-year interval and so on.[3] It is easy to see that the presence of the last tuple in the table necessitates, for this requirement to be satisfied, two steps of generalization on `Date of Birth`, one step of generalization on `Zip Code`, one step of generalization on `Marital Status`, and either one further step on `Sex`, `Zip Code`, and `Marital Status`, or, alternatively, on `Ethnicity` and `Date of Birth`. The two possible minimal generalizations are as illustrated in Figure 6. In practice, in both cases almost all the attributes must be generalized. It can be easily seen, at the same time, that had this last tuple not been present k-anonymity could have been simply achieved by two steps of generalization on attribute `Date of Birth`, as illustrated in Figure 7. Suppressing the tuple would in this case permit enforcement of less generalization.

In illustrating how suppression interplays with generalization to provide k-anonymity, we begin by restating the definition of *generalized table* as follows.

Definition 4.1 (Generalized Table - with suppression) *Let $T_i(A_1, \ldots, A_n)$ and $T_j(A_1, \ldots, A_n)$ be two tables defined on the same set of attributes. T_j is said to be a generalization of T_i, written $T_i \preceq T_j$, iff*

1. $|T_j| \leq |T_i|$
2. $\forall z = 1, \ldots, n : dom(A_z, T_i) \leq_{\mathsf{D}} dom(A_z, T_j)$
3. *It is possible to define an injective mapping between T_i and T_j that associates tuples $t_i \in T_i$ and $t_j \in T_j$ such that $t_i[A_z] \leq_{\mathsf{V}} t_j[A_z]$.*

[3] Note that although generalization may seem to change the format of the data, compatibility can be assured by using the same representation form. For instance, the month can be represented always as a specific day. This is actually the trick that we used in our application of generalization.

Eth:E_0	**ZIP**:Z_0
asian	02138
asian	02138
asian	02142
asian	02142
black	02138
black	02141
black	02142
white	02138

PT

Eth:E_1	**ZIP**:Z_0
person	02138
person	02138
person	02142
person	02142
person	02138
person	02142
person	02138

$\mathsf{GT}_{[1,0]}$

Eth:E_0	**ZIP**:Z_1
asian	02130
asian	02130
asian	02140
asian	02140
black	02140
black	02140

$\mathsf{GT}_{[0,1]}$

Eth:E_0	**ZIP**:Z_2
asian	02100
asian	02100
asian	02100
asian	02100
black	02100
black	02100
black	02100

$\mathsf{GT}_{[0,2]}$

Eth:E_1	**ZIP**:Z_1
person	02130
person	02130
person	02140
person	02140
person	02130
person	02140
person	02140
person	02130

$\mathsf{GT}_{[1,1]}$

Figure 9: Examples of generalized tables for PT

The definition above differs from Definition 3.1 since it allows tuples appearing in T_i not to have any corresponding generalized tuple in T_j. Intuitively, tuples in T_i not having any correspondent in T_j are tuples which have been suppressed.

Definition 4.1 allows any amount of suppression in a generalized table. Obviously, we are not interested in tables that suppress more tuples than necessary to achieve k-anonymity at a given level of generalization. This is captured by the following definition.

Definition 4.2 (Minimal required suppression) *Let T_i be a table and T_j a generalization of T_i satisfying k-anonymity. T_j is said to enforce* minimal required suppression *iff $\nexists T_z$ such that $T_i \leq T_z, \mathsf{DV}_{i,z} = \mathsf{DV}_{i,j}, |T_j| < |T_z|$ and T_z satisfies k-anonymity.*

Example 4.1 *Consider the table* PT *and its generalizations illustrated in Figure 8. The tuples written in bold face and marked with double lines in each table are the tuples that must be suppressed to achieve k-anonymity of 2. Suppression of a subset of them would not reach the required anonymity. Suppression of any superset would be unnecessary (not satisfying minimal required suppression).*

Allowing tuples to be suppressed typically affords more tables per level of generalization. It is trivial to prove, however, that for each possible distance vector, the generalized table satisfying a k-anonymity constraint by enforcing minimal suppression is unique. This table is obtained by first applying the generalization described by the distance vector and then removing *all and only* the tuples that appear with fewer than k occurrences.

In the remainder of this paper we assume the condition stated in Definition 4.2 to be satisfied, that is, all generalizations that we consider enforce minimal required suppression. Hence, in the following, within the context of a k-anonymity constraint, when referring to the generalization at a given distance vector we will intend the *unique* generalization for that distance vector which satisfies the k-anonymity constraint enforcing minimal required suppression. To illustrate, consider the table PT in Figure 8; with respect to k-anonymity with k=2, we would refer to its generalizations as illustrated in Figure 9. (Note that for sake of clarity, we have left an empty row to correspond to each removed tuple.)

Generalization and suppression are two different approaches to obtaining, from a given table, a table which satisfies k-anonymity. It is trivial to note that the two approaches produce the best results when jointly applied. For instance, we have already noticed how, with respect to the table in Figure 1, generalization alone is unsatisfactory (see Figure 6). Suppression alone, on the other side, would require suppression of all tuples in the table. Joint application of the two techniques allows, instead, the release of a table like the one in Figure 7. The question is therefore whether it is better to generalize, at the cost of less precision in the data, or to suppress, at the cost of completeness. From observations of real-life applications and requirements [16], we assume the following. We consider an *acceptable suppression* threshold MaxSup, as specified, stating the maximum number of suppressed tuples that is considered acceptable. Within this acceptable threshold, suppression is considered preferable to generalization (in other words, it is better to suppress more tuples than to enforce more generalization). The reason for this is that suppression affects single tuples whereas generalization modifies all values associated with an attribute, thus affecting all tuples in the table. Tables which enforce suppression beyond MaxSup are considered unacceptable.

Given these assumptions, we can now restate the definition of k-minimal generalization taking suppression into consideration.

Definition 4.3 (k-minimal generalization - with suppression) *Let T_i and T_j be two tables such that $T_i \leq T_j$ and let MaxSup be the specified threshold of acceptable suppression. T_j is said to be a k-*minimal *generalization of a table T_i iff*

1. *T_j satisfies k-anonymity*
2. *$|T_i| - |T_j| \leq \mathsf{MaxSup}$*
3. *$\nexists T_z : T_i \leq T_z, T_z$ satisfies conditions 1 and 2, and $DV_{i,z} < DV_{i,j}$.*

Intuitively, generalization T_j is k-minimal iff it satisfies k-anonymity, it does not enforce more suppression than it is allowed, and there does not exist another generalization satisfying these conditions with a distance vector smaller than that of T_j, nor does there exist another table with the same level of generalization satisfying these conditions with less suppression.

Example 4.2 *Consider the private table PT illustrated in Figure 9 and suppose k-anonymity with $k = 2$ is required. The possible generalizations (but the topmost one collapsing every tuple to $\langle \texttt{person}, \texttt{02100} \rangle$) are illustrated in Figure 9. Depending on the acceptable suppression threshold, the following generalizations are considered minimal:*
$\mathsf{MaxSup} = 0$: $\mathsf{GT}_{[1,1]}$ *($\mathsf{GT}_{[1,0]}$, $\mathsf{GT}_{[0,1]}$, or $\mathsf{GT}_{[0,2]}$ suppress more tuple than it is allowed, $\mathsf{GT}_{[1,2]}$ is not minimal because of $\mathsf{GT}_{[1,1]}$);*
$\mathsf{MaxSup} = 1$: $\mathsf{GT}_{[1,0]}$ *and* $\mathsf{GT}_{[0,2]}$ *($\mathsf{GT}_{[0,1]}$ suppresses more tuple than it is allowed, $\mathtt{GT}_{[1,1]}$ is not minimal because of $\mathsf{GT}_{[1,0]}$ and $\mathsf{GT}_{[1,2]}$ is not minimal because of $\mathtt{GT}_{[1,0]}$ and $\mathsf{GT}_{[0,2]}$);*
$\mathsf{MaxSup} \geq 2$: $\mathsf{GT}_{[1,0]}$ *and* $\mathsf{GT}_{[0,1]}$ *($\mathsf{GT}_{[0,2]}$ is not minimal because of $\mathtt{GT}_{[0,1]}$, $\mathtt{GT}_{[1,1]}$ and $\mathsf{GT}_{[1,2]}$ are not minimal because of $\mathsf{GT}_{[1,0]}$ and $\mathsf{GT}_{[0,1]}$).*

5 Preferences

It is clear from Section 4 that there may be more than one minimal generalization for a given table, suppression threshold and k-anonymity constraint. This is completely legitimate since the definition of "minimal" only captures the concept that the least amount of generalization and suppression necessary to achieve k-anonymity is enforced. However, multiple solutions may exist which satisfy this condition. Which of the solutions is to be preferred depends on subjective measures and preferences of the data recipient. For instance, depending on the use of the released data, it may be preferable to generalize some attributes instead of others. We outline here some simple preference policies that can be applied in choosing a preferred minimal generalization. To do that, we first introduce two distance measures defined between tables: *absolute* and *relative* distance. Let $T_i(A_1, \ldots, A_n)$ be a table and $T_j(A_1, \ldots, A_n)$ be one of its generalizations with distance vector $\mathsf{DV}_{i,j} = [d_1, \ldots, d_n]$. The *absolute distance* of T_j from T_i, written $\mathsf{Absdist}_{i,j}$, is the sum of the distances for each attribute. Formally, $\mathsf{Absdist}_{i,j} = \sum_{i=1}^{n} d_i$. The relative distance of T_j from T_i, written $\mathsf{Reldist}_{i,j}$, is the sum of the "relative" distance for each attribute, where the relative distance of each attribute is obtained by dividing the distance over the total height of the hierarchy. Formally, $\mathsf{Reldist}_{i,j} = \sum_{z=1}^{n} \frac{d_z}{h_z}$, where h_z is the height of the domain generalization hierarchy of $dom(A_z, T_i)$.

Given those distance measures we can outline the following basic preference policies:

Minimum absolute distance prefers the generalization(s) that has a smaller absolute distance, that is, with a smaller total number of generalization steps (regardless of the hierarchies on which they have been taken).

Minimum relative distance prefers the generalization(s) that has a smaller relative distance, that is, that minimizes the total number of relative steps, that is, considered with respect to the height of the hierarchy on which they are taken.

Maximum distribution prefers the generalization(s) that contains the greatest number of distinct tuples.

Minimum suppression prefers the generalization(s) that suppresses less, that is, that contains the greater number of tuples.

Example 5.1 *Consider Example 4.2. Suppose* $\mathsf{MaxSup} = 1$*. Minimal generalizations are* $\mathsf{GT}_{[1,0]}$ *and* $\mathsf{GT}_{[0,2]}$*. Under minimum absolute distance,* $\mathsf{GT}_{[1,0]}$ *is preferred. Under minimum relative distance, maximum distribution, and minimum suppression policies, the two generalizations are equally preferable. Suppose* $\mathsf{MaxSup} = 2$*. Minimal generalizations are* $\mathsf{GT}_{[1,0]}$ *and* $\mathsf{GT}_{[0,1]}$*. Under the minimum absolute distance policy, the two generalizations are equally preferable. Under the minimum suppression policy,* $\mathsf{GT}_{[1,0]}$ *is preferred. Under the minimum relative distance and the maximum distribution policies,* $\mathsf{GT}_{[0,1]}$ *is preferred.*

The list above is obviously not complete and there remain additional preference policies that could be applied; the best one to use, of course, depends on the specific use for the released data. Examination of an exhaustive set of possible policies is outside the scope of this paper. The choice of a specific preference policy is done by the requester at the time of access [18]. Different preference policies can be applied to different quasi-identifiers in the same released data.

6 Computing a preferred generalization

We have defined the concept of preferred k-minimal generalization corresponding to a given private table. Here, we illustrate an approach to computing such a generalization. Before discussing the algorithm we make some observations clarifying the problem of finding a minimal generalization and its complexity. We use the term *outlier* to refer to a tuple with fewer than k occurrences, where k is the anonymity constraint required.

First of all, given that the k-anonymity property is required only for attributes in quasi-identifiers, we consider the generalization of each specific quasi-identifier within table PT independently. Instead of considering the whole table PT to be generalized, we consider its projection $\mathsf{PT}[QI]$, keeping duplicates, on the attributes of a quasi-identifier QI. The generalized table PT is obtained by enforcing generalization for each quasi-identifier $QI \in \mathsf{QI}_{\mathsf{PT}}$. The correctness of the combination of the generalizations independently produced for each quasi-identifier is ensured by the fact that the definition of a generalized table requires correspondence of values across whole tuples and by the fact that the quasi-identifiers of a table are disjoint. [4]

In Section 3 we illustrated the concepts of a generalization hierarchy and strategies for a domain tuple. Given a quasi-identifier $QI = (A_1, \ldots, A_n)$, the corresponding domain hierarchy on $DT = \langle D_1, \ldots, D_n \rangle$ pictures all the possible generalizations and their relationships. Each path (strategy) in it defines a different way in which generalization can be applied. With respect to a strategy, we could define the concept of local minimal generalization as the generalization that is minimal with respect to the set of generalizations in the strategy (intuitively the first found in the path from the bottom element DT to the top element). Each k-minimal generalization is locally minimal with respect to some strategy, as stated by the following theorem.

[4] This last constraint can be removed provided that generalization of non-disjoint quasi-identifiers be executed serially.

Theorem 6.1 *Let $T(A_1,\ldots,A_n) = \mathsf{PT}[QI]$ be the table to be generalized and let $DT = \langle D_1,\ldots,D_n\rangle$ be the tuple where $D_z = dom(A_z,T)$, $z = 1,\ldots,n$, be a table to be generalized. Every k-minimal generalization of T_i is a local minimal generalization for some strategy of DGH_{DT}.*

PROOF.*(sketch)* By contradiction. Suppose T_j is k-minimal but is not locally minimal with respect to any strategy. Then, there exists a strategy containing T_j such that there exists another generalization T_z dominated by T_j in this strategy which satisfies k-anonymity by suppressing no more tuples than what is allowed. Hence, T_z satisfies conditions 1 and 2 of Definition 4.3. Moreover, since T_z is dominated by T_j, $DV_{i,z} < DV_{i,j}$. Hence, T_j cannot be minimal, which contradicts the assumption. □

Since strategies are not disjoint, the converse is not necessarily true, that is, a local minimal generalization with respect to a strategy may not correspond to a k-minimal generalization.

From Theorem 6.1, following each generalization strategy from the domain tuple to the maximal element of the hierarchy would then reveal all the local minimal generalizations from which the k-minimal generalizations can be selected and an eventual preferred generalization chosen. (The consideration of preferences implies that we cannot stop the search at the first generalization found that is known to be k-minimal.) However, this process is much too costly because of the high number of strategies which should be followed. It can be proved that the number of different strategies for a domain tuple $DT = \langle D_1,\ldots,D_n\rangle$ is $\frac{(h_1+\ldots+h_n)!}{h_1!\ldots h_n!}$, where each h_i is the length of the path from D_i to the top domain in DGH_{D_i}.

In the implementation of our approach we have realized an algorithm that computes a preferred generalization without needing to follow all the strategies and computing the generalizations. The algorithm makes use of the concept of distance vector between tuples. Let T be a table and $x,y \in T$ two tuples such that $x = \langle v'_1,\ldots,v'_n\rangle$ and $y = \langle v''_1,\ldots,v''_n\rangle$ where each v'_i, v''_i is a value in domain D_i. The **distance vector** between x and y is the vector $\mathsf{V}_{x,y} = [d_1,\ldots,d_n]$ where d_i is the length of the paths from v'_i and v''_i to their closest common ancestor in the value generalization hierarchy VGH_{D_i}. For instance, with reference to the PT illustrated in Figure 4, the distance between $\langle$`asian,02139`$\rangle$ and $\langle$`black,02139`$\rangle$ is [1,0]. Intuitively, the distance between two tuples x and y in table T_i is the distance vector between T_i and the table T_j, with $T_i \leq T_j$ where the domains of the attribute in T_j are the most specific domains for which x and y generalize to the same tuple t.

The following theorem states the relationship between distance vectors between tuples in a table and a minimal generalization for the table.

Theorem 6.2 *Let $T_i(A_1,\ldots,A_n) = \mathsf{PT}[QI]$ and T_j be two tables such that $T_i \leq T_j$. If T_j is k-minimal then $\mathsf{DV}_{i,j} = V_{x,y}$ for some tuples x,y in T_i such that either x or y has a number of occurrences smaller than k.*

PROOF.*(sketch)* By contradiction. Suppose that a k-minimal generalization T_j exists such that $\mathsf{DV}_{i,j}$ does not satisfy the condition above. Let $\mathsf{DV}_{i,j} = [d_1,\ldots,d_n]$. Consider a strategy containing a generalization with that distance vector (there will be more than one of such strategies, and which one is considered is not important). Consider the different generalization steps executed according to the strategy, from the bottom going up, arriving at the generalization corresponding to T_j. Since no outlier is at exact distance $[d_1,\ldots,d_n]$ from any tuple, no outlier is merged with any tuple at the last step of generalization considered. Then the generalization directly below T_j in the strategy satisfies the same k-anonymity constraint as T_i with the same amount of suppression. Also, by definition of strategy, $\mathsf{DV}_{i,z} < \mathsf{DV}_{i,j}$. Then, by Definition 4.3, T_j cannot be minimal, which contradicts the assumption. □

According to Theorem 6.2 the distance vector of a minimal generalization falls within the set of the vectors between the outliers and other tuples in the table. This property is exploited by the generalization algorithm to reduce the number of generalizations to be considered.

The algorithm works as follows. Let $\mathsf{PT}[QI]$ be the projection of PT over quasi-identifier QI. First, all distinct tuples in $\mathsf{PT}[QI]$ are determined together with the number of their occurrences. Then, the distance vectors between each outlier and every tuple in the table is computed. Then, a DAG with, as nodes, all distance vectors found is constructed. There is an arc from each vector to all the smallest vector dominating it in the set. Intuitively, the DAG corresponds to a "summary" of the strategies to be considered (not all strategies may be represented, and not all generalizations of a strategy may be present). Each path in the DAG is then followed from the bottom up until a minimal local generalization is found. The algorithm determines if a generalization is locally minimal simply by controlling how the occurrences of the tuples would combine (on the basis of the distance table constructed at the beginning), without actually performing the generalization. When a local generalization is found, another path is followed. As paths may be not disjoint, the algorithm keeps track of generalizations that have been considered so as to stop on a path when it runs into another path on which a local minimum has already been found. Once all possible paths have been examined, the evaluation of the distance vectors allows the determination of the generalizations, among those found, which are k-minimal. Among them, a preferred generalization to be computed is then determined on the basis of the distance vectors and of how the occurrences of tuples would combine.

The characteristics that reduce the computation cost are therefore that *(1)* the computation of the distance vectors between tuples greatly reduces the number of generalizations to be considered; *(2)* generalizations are not actually computed but foreseen by looking at how the occurrences of the tuples would combine; *(3)* the fact that the algorithm keeps track of evaluated generalizations allows it to stop evaluation on a path whenever it crosses a path already evaluated.

The correctness of the algorithm descends directly from Theorems 6.1 and 6.2.

The necessary and sufficient condition for a table T to satisfy k-anonymity is that the cardinality of the table be at least k, and only in this case, therefore, is the algorithm applied. This is stated by the following theorem.

Theorem 6.3 *Let T be a table,* $\mathsf{MaxSup} \leq |T|$ *be the* acceptable suppression *threshold, and k be a natural number. If $|T| \geq k$, then there exists at least a k-minimal generalization for T. If $|T| < k$ there are no non-empty k-minimal generalizations for T.*

Proof.*(sketch)* Suppose $|T| \geq k$. Consider the generalization generalizing each tuple to the topmost possible domain. Since maximal elements of Dom are singleton, all values of an attribute collapse to the same value. Hence, the generalization will contain $|T|$ occurrences of the same tuple. Since $|T| \geq k$, it satisfies k-anonymity. Suppose $|T| < k$, no generalization can satisfy k-anonymity, which can be reached only by suppressing all the tuples in T. □

7 Application of the approach: some experimental results

We constructed a computer program that produces tables adhering to k-minimal generalizations given specific thresholds of suppression. The program was written in C++, using ODBC to interface with an SQL server, which in turn accessed a medical database. Our goal was to model an actual release and to measure the quality of the released data. Most states have legislative mandates to collect medical data from hospitals, so we collapsed the original medical database into a single table consistent with the format and primary attributes the National Association of Health Data Organizations recommends that state agencies collect [14]. Each tuple represents one patient, and each patient is unique. The data contained medical records for 265 patients. Figure 10 itemizes the attributes used; the table is considered de-identified because it contains no explicit identifying information such as name or address. As discussed earlier, ZIP code, date of birth, and gender can be linked to population registers that are publicly available in order to re-identify patients [18]. Therefore, the quasi-identifier QI {`ZIP`, `birth date`, `gender`, `ethnicity`} was considered. Each tuple within QI was found to be unique.

The top table in Figure 10 is a sample of the original data, and the lower table illustrates a k-minimal generalization of that table given a threshold of suppression. The ZIP field has been generalized to the

Attribute	# distinct values	min frequency	max frequency	median frequency	comments
ZIP	66	1	24	2	
Birth year	23	1	31	10	23 yr range
Gender	2	96	169	132	
Ethnicity	4	6	211	24	

Table 1: Distribution of values in the table considered in the experiment

first 3 digits, and date of birth to the year. The tuple with the unusual ZIP code of 32030-1057 has been suppressed. The recipient of the data is informed of the levels of generalizations and how many tuples were suppressed. (Note: The default value for month is January and for day is the 1st when dates are generalized. This is done for practical considerations that preserve the data type originally assigned to the attribute (see Section 3).)

Table 1 itemizes the basic distribution of values within the attributes. ZIP codes were stored in the full 9-digit form, with a generalization hierarchy replacing rightmost digits with 0, of 10 levels. Birth dates were generalized first to the month, then 1-year, 5-year, 10-year, 20-year, and 100-year periods. A two-level hierarchy was considered for gender and ethnicity (see Figure 2). The product of the number of possible domains for each attribute gives the total number of possible generalizations, which is 280.

The program constructed a clique where each node was a tuple and the edges were weighted by distance vectors between adjacent tuples. Reading these vectors from the clique, the program generated a set of generalizations to consider. There were 141 generalizations read from the clique, discarding 139 or 50%. For our tests, we used values of k to be 3, 6, 9, ..., 30 and a maximum suppression threshold of 10% or 27 tuples.

Figure 11 shows the relationship between suppression and generalization within the program in a practical and realistic application. We measure the loss of data quality due to suppression as the ratio of the number of tuples suppressed divided by the total number of tuples in the original data. We define the inverse measure of "completeness", to determine how much of the data remains, computed as one minus the loss due to suppression. Generalization also reduces the quality of the data since generalized values are less precise. We measure the loss due to generalization as the ratio of the level of generalization divided by the total height of the generalization hierarchy. We term "precision" as the amount of specificity remaining in the data, computed as one minus the loss due to generalization.

In Charts (A) and (B) of Figure 11 we compare the data quality loss as the k-anonymity requirement increases. Losses are reported for both generalization and suppression for each attribute as if it were solely responsible for achieving the k-anonymity requirement. By doing so, we characterize the distribution and nature of values found in these attributes. Given the distribution of males (96) and females (169) in the data, the gender attribute itself can achieve these values of k so we see no loss due to generalization or suppression. On the other hand, there were 258 of 265 distinct birth dates. Clearly, date of birth and ZIP code are the most discriminating values, so it is not surprising that they must be generalized more than other attributes. The flat lines on these curves indicate values being somewhat clustered

Charts (C) and (D) of Figure 11 report completeness and precision measurements for the 44 minimal generalizations found. Basically, generalizations that satisfy smaller values of k appear further to the right in chart (C), and those generalizations that achieve larger values of k are leftmost. This results from the observation that the larger the value for k, the more generalization may be required, resulting, of course, in a loss of precision. It is also not surprising that completeness remains above 0.90 because our suppression threshold during these tests was 10%. Though not shown in the charts, it can easily be understood that raising the suppression threshold typically improves precision since more values can be suppressed to achieve k. Clearly, generalization is expensive to the quality of the data since it is performed across the entire attribute; every tuple is affected. On the other hand, it remains semantically more useful to have a value

Figure 10: Example of current release practice and minimally generalized equivalent

Figure 11: Experimental results based on 265 medical records

present, even if it is a less precise one, than not having any value at all, as is the result of suppression.

From these experiments it is clear that the techniques of generalization and suppression can be used in practical applications. Of course, protecting against linking involves a loss of data quality in the attributes that comprise the quasi-identifier, though we have shown that the loss is not severe. These techniques are clearly most effective when the primary attributes required by the recipient are not the same as the quasi-identifier that can be used for linking. In the sample medical data shown earlier, researchers, computer scientists, health economists and others value the information that is not included in the quasi-identifier in order to develop diagnostic tools, perform retrospective research, and assess hospital costs [18].

8 Conclusions

We have presented an approach to disclosing entity-specific information such that the released table cannot be reliably linked to external tables. The anonymity requirement is expressed by specifying a quasi-identifier and a minimum number k of duplicates of each released tuple with respect to the attributes of the quasi-identifier. The anonymity requirement is achieved by generalizing, and possibly suppressing, information upon release. We have given the notion of minimal generalization capturing the property that information is not generalized more than it is needed to achieve the anonymity requirement. We have discussed possible preference policies to choose between different minimal generalizations and an algorithm to compute a preferred minimal generalization. Finally, we have illustrated the results of some experiments from the application of our approach to the release of a medical database containing information regarding 265 patients.

This work represents only a first step toward the definition of a complete framework for information disclosure control. Many problems are still open. From a modeling point of view, the definition of quasi-identifiers and of an appropriate size of k must be addressed. The quality of generalized data is best when the attributes most important to the recipient do not belong to any quasi-identifier. For public-use files this may be acceptable, but determining the quality and usefulness in other settings must be further researched. From the technical point of view, future work should include the investigation of an efficient algorithm [15] to enforce the proposed techniques and the consideration of specific queries, of multiple releases over time, and of data updating, which may allow inference attacks [10, 13].

Acknowledgments

We thank Steve Dawson, at SRI, for discussions and support; Rema Padman at CMU for discussions on metrics; and, Dr. Lee Mann of Inova Health Systems, Lexical Technology, Inc., and Dr. Fred Chu for making medical data available to validate our approaches. We also thank Sylvia Barrett and Henry Leitner of Harvard University for their support.

References

[1] N.R. Adam and J.C. Wortman. Security-control methods for statistical databases: A comparative study. *ACM Computing Surveys*, 21:515–556, 1989.

[2] Ross Anderson. A security policy model for clinical information systems. In *Proc. of the 1996 IEEE Symposium on Security and Privacy*, pages 30–43, Oakland, CA, May 1996.

[3] Silvana Castano, Maria Grazia Fugini, Giancarlo Martella, and Pierangela Samarati. *Database Security.* Addison Wesley, 1995.

[4] P.C. Chu. Cell suppression methodology: The importance of suppressing marginal totals. *IEEE Trans. on Knowledge Data Systems*, 4(9):513–523, July/August 1997.

[5] L.H. Cox. Suppression methodology in statistical disclosure analysis. In *ASA Proceedings of Social Statistics Section*, pages 750–755, 199.

[6] Tore Dalenius. Finding a needle in a haystack - or identifying anonymous census record. *Journal of Official Statistics*, 2(3):329–336, 1986.

[7] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order.* Cambridge University Press, 1990.

[8] Dorothy E. Denning. *Cryptography and Data Security.* Addison-Wesley, 1982.

[9] Dan Gusfield. A little knowledge goes a long way: Faster detection of compromised data in 2-D tables. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 86–94, Oakland, CA, May 1990.

[10] J. Hale and S. Shenoi. Catalytic inference analysis: Detecting inference threats due to knowledge discovery. In *Proc. of the 1997 IEEE Symposium on Security and Privacy*, pages 188–199, Oakland, CA, May 1997.

[11] A. Hundepool and L. Willenborg. μ- and τ-Argus: Software for statistical disclosure control. In *Third International Seminar on Statistical Confidentiality*, Bled, 1996.

[12] Ram Kumar. Ensuring data security in interrelated tabular data. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 96–105, Oakland, CA, May 1994.

[13] Teresa Lunt. Aggregation and inference: Facts and fallacies. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 102–109, Oakland, CA, May 1989.

[14] National Association of Health Data Organizations, Falls Church. *A Guide to State-Level Ambulatory Care Data Collection Activities*, October 1996.

[15] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when dislosing information. In *Proc. of the ACM SIGACT-SIGMOD-SIGART 1998 Symposium on Principles of Database Systems (PODS98)*, Seattle, USA, June 1998.

[16] Latanya Sweeney. Computational disclosure control for medical microdata. In *Record Linkage Workshop Bureau of the Census*, Washington, DC, 1997.

[17] Latanya Sweeney. Guaranteeing anonymity when sharing medical data, the Datafly system. In *Proc. Journal of the American Medical Informatics Association*, Washington, DC: Hanley & Belfus, Inc., 1997.

[18] Latanya Sweeney. Weaving technology and policy together to maintain confidentiality. *Journal of Law, Medicine, & Ethics*, 25(2–3):98–110, 1997.

[19] Rein Turn. Information privacy issues for the 1990s. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 394–400, Oakland, CA, May 1990.

[20] Jeffrey D. Ullman. *Principles of Databases and Knowledge-Base Systems*, volume I. Computer Science Press, 1989.

[21] L. Willenborg and T. De Waal. *Statistical disclosure control in practice.* New York: Springer-Verlag, 1996.

[22] L. Willenborg and T. De Waal. *Statistical Disclosure Control in Practice.* Springer-Verlag, 1996.

[23] Beverly Woodward. The computer-based patient record confidentiality. *The New England Journal of Medicine*, 333(21):1419–1422, 1995.