An Extended Memoryless Inference Control Model: Accounting for Dependence in Table-level Controls

S. C. Hansen E. A. Unger
Department of Computing and Information Sciences
Kansas State University, Manhattan, Kansas 66506

Abstract

Memoryless inference controls are an important class of inference control methods for on-line statistical databases. Other inference controls are usually too complex to use in on-line systems. A subclass of memoryless inference controls which have been shown to require a reasonable level of computation are table level controls. Table level controls described in the literature make some assumptions about the structure of the database on which they are used. We formalize these assumptions as database restrictions, and formally prove that these table level controls provide an accurate measure of the identification risk of a query. In addition we extend these controls to account for interdependence among attributes.

1 Introduction

Protecting databases from inferential attacks on confidential information is a database security problem of great complexity[AW89]. It has been shown that it is impossible to release accurate statistical information while protecting the database against all possible releases of confidential information[AW89, DS83]. A number of efforts in the field attempt to solve the problem of preventing or reducing inferential attacks on statistical databases by restricting the release of sensitive statistics (sensitive statistics are those statistics which will allow the release of confidential data through release of the statistic in question alone[DS83]) or by altering the data and/or the statistical output[AW89]. Many of the methods thus created have been found to be too expensive to be implemented in online systems and/or ineffective.

A class of inferential control methods which tend to require only a reasonable amount of computation are the memoryless inference controls. These controls make a decision as to whether a query should be accepted or rejected without requiring a record of the results of previous queries by the same user (or group of users). Because no calculations need to be done based on previous queries by the user, much of the cost of other inferential control methods can be avoided. The approach taken in this paper is to extend the work of others in the area of memoryless inference controls [DSW84, DS83] through the development and use of semantic integrity constraints akin to the functional and multivalued dependencies.

Previous work suggests that the area of memoryless inference control can be divided into two subareas, cell level controls and table level controls[Sch75, DSW84, DS83]. The cells and tables referred to in the names of these controls are components of a lattice of m-tables, where each table has m dimensions represented by the attributes in a relational database. The "depth" of a single dimension is represented by the size of the active domain of that dimension. A database with 3 attributes has possible queries which are categorized into 8 tables; a single 3-attribute table, three 2-attribute tables, three 1 attribute tables, and finally a table which includes all of the tuples in the table. Each of these tables may contain many cells, each corresponding to a specific set of values for each of the attributes representing dimensions of the table. It is probable that some of the cells will represent combinations of attribute values which are not present in the database, thus the set size for these cells will be zero. Cells with sets of size zero will be more common in high order tables, as these tables have more possible cells for the tuple values of the database to fill.

Cell level controls are those controls which block a query if that query specifies a cell which is sensitive. Cell level controls tend to be expensive, as they require calculations of the values of more than one cell if they are to be effective[DSW84]. Table level controls operate with the intent of blocking a query if the number of cells in the table specifying less than k tuples is larger than some fraction of the database. Our work examines the possibility of extending table level memoryless inference controls in a manner that approaches a cell level

of control.

Table level memoryless inference controls have previously been divided into three types, maximum order controls, maximum relative table size controls, and minimum frequency controls[DSW84].

Maximum order controls capitalize on the fact that as more attributes are specified in a query, the table in which the tuples are represented increases in size. This increase in table size causes any given area of the table to be less dense, and the possibility of cells which represent fewer than k tuples represented increases, where k is some sensitivity level set by the DBA. To protect against the release of sensitive information, m-order tables are restricted from release if m is above some pre-set size value.

A problem with this method is that it treats all attributes alike, whether they have a domain size of 2 or 200. For a database of size 2000, a single attribute with a domain size of 2 will specify an average query set of size 1000. If, however, the domain size of the attribute was 2000, the average query set size would be 1. Thus maximum order control is probably too coarse a measurement of the probability of identification of sensitive information[DSW84].

Maximum relative table size controls take into account the domain sizes of the attributes, and restrict queries based on the size of the table they are in rather than on the number of attributes specified in the query. The table size can be calculated by multiplying the active domain sizes of all the attributes specified in the query. These domain sizes can be stored in the data dictionary, and this method of query restriction can be done relatively inexpensively. It has been shown in a test of 5 databases that the maximum relative table size control could be set to a level which would achieve an identification risk of less than 1% without being "overly restrictive"[DSW84].

Minimum-frequency controls take into account the fact that some values in the domains of attributes are present in the database significantly fewer times than other values in the same attribute. Minimum-frequency controls calculate the value of a query by multiplying the smallest relative frequency of each of the values represented in the query. The value thus calculated is compared to a preset value $1/K$. If the value is greater than $1/K$, the query is allowed. Tests on 5 databases showed that this method is not good at predicting the identification risk of a query[DSW84]. (Identification risk of a query is the number of cells which represent a single tuple in a table divided by the size of the database[DSW84].

2 Extending Maximum Relative Table Size Controls

Maximum relative table size controls are useful in predicting the identification power of a set of attributes when used in a query. To better understand why this measurement is useful, a theoretical development of these type of controls is presented.

Before this theoretical development is undertaken, we discuss a generalized view of the power of value specification for attributes in reducing the membership in a query set. To adequately handle this presentation, we first give a few definitions:

Definition 1 *Characteristic Formula, C*

A subgroup of tuples in the database is specified or characterized by a **characteristic formula** C. C is a formula consisting of clauses specifying the values of attributes which may be be connected by the logical operators *OR*, *AND*, and *NOT* (operators are written in order of increasing precedence).

Definition 2 *Query Set, Q*

The set of tuples, Q, from a database whose attribute values cause a characteristic formula C to be satisfied is called the **query set** of C for that database.

Definition 3 *Attribute Domain of an attribute* a_j, *dom*(a_j)

The domain of an attribute a_j, dom(a_j), consists of all the possible values which are semantically correct values for the that attribute in the database. To avoid the inherent problems of infinite domains, we will henceforth use the concept of active domain (Definition 4) to replace the concept of domain in all areas of our research.

Definition 4 *Active Domain of attribute* a_j *over database instance* r, *adom*(a_j, r)

The active domain of a_j over database instance $r(adom(a_j, r))$ is the set of values that exist for attribute a_j within a database instance.

Definition 5 *Domain Size of an attribute* a_j *over database instance* r, $\mid adom(a_j, \mathrm{r}) \mid$

The domain size of an attribute a_j over database instance r is the size of the active domain of a_j, $\mid adom(a_j, r) \mid$, or that is the cardinality of adom(a_j, r) The active domain may be calculated through the domain calculus query $\mid \pi_{a_j}(r) \mid$ (Note: π is the projection operator in domain calculus and relational algebra).

It is important to logically divide the attributes in a statistical database into three sets, not necessarily

disjoint. These sets are the characterizing attributes, the confidential attributes, and the attributes whose expected use is for statistical purposes, statistical attributes. For any given query these sets can be clearly identified. The thrust of this work will be to generalize these sets for all or a large proportion of the queries to be directed to the database. These attribute sets are defined below:

Definition 6 *Characterizing Attributes, $A_C(q)$ for a query*

Characterizing attributes are those attributes which are used in the characteristic formula, C, of a query. Given a query, q, the characterizing attributes $A_C(q)$ are those attributes used in the characteristic formula of the query.

Definition 7 *Characterizing Attributes,* $\mathbf{A_C(r)}$ *of a database*

For a given database the characterizing attributes $\mathbf{A_C(r)}$ are those which are known to be used in characterizing formulas or which are identified as having potential for use in characterizing formulas. The characterizing attribute set may contain attributes which are also in the statistical attribute set. As an example, in an employee statistical database which includes address-city, address-state, wage, age, veteran, department, and gender attributes, all of the attributes may potentially be used in characterizing formulas, though wage may be the least useful in that regard if the use of ranges is disallowed.

Definition 8 *Confidential Attributes,* $\mathbf{A_{CO}(r)}$, *of a database*

Confidential attributes, $\mathbf{A_{CO}(r)}$, of a database, are those attributes which have a security level above that of the anticipated user of the statistical database and thus are unavailable for reference in the query. Attributes or attribute sets which uniquely identify individuals would fall into this category. Examples would include Social Security Number or other ID numbers, along with names and street addresses.

Definition 9 *Statistical Attributes, $A_S(q)$, of a query*

Statistical attributes are those attributes from which the user may collect statistics. For a given query q, the statistical attributes $A_S(q)$ are those attributes about which statistics are requested.

Definition 10 *Statistical Attributes,* $\mathbf{A_S(r)}$, *of a database*

For a given database the statistical attributes $\mathbf{A_S(r)}$ are those known to be used or to have potential to be used to retrieve statistical information from the database. For the attributes listed in the definition of characterizing attributes, (Def. 2), wages would be an obvious member of the set of statistical attributes, but, through the use of the "count" statistic, all of the attributes might be in the set of statistical attributes, $\mathbf{A_S(r)}$, for the database. In addition to the above definitions, we will use the concept of restricting power in our discussion. The necessary definitions follow.

Definition 11 *Restricting Power of a value v_j from the domain of a single attribute a_i* ($\Re(a_i = v_j, \mathrm{r})$)

The Restricting Power of a value v_j from the domain of a single attribute a_i ($\Re(a_i = v_j, r)$), is the proportion of the number of tuples in the database containing that value for that attribute to the total number of tuples in the database.

Definition 12 *Restricting power of an attribute a_i* ($\Re(a_i, \mathrm{r})$)

The Restricting Power of an attribute a_i ($\Re(a_i, r)$), is some measure of the collection of restrictive powers for each value in the active domain of that attribute in the database instance r. (The measure used for a query on a database meeting Restrictions 1-4 is the mean, as the mean is the same as the restrictive power of each of the values in the active domain)

Definition 13 *Restricting Power of a set of values $\{v_1..v_n\}$ for an attribute set $\{a_1..a_n\}$* ($\Re(\{a_1 = v_1..a_n = v_n\}, \mathrm{r})$)

The Restricting Power of a set of values $\{v_1..v_n\}$ for an attribute set $\{a_1..a_n\}$ ($\Re(\{a_1 = v_1..a_n = v_n\}, r)$), is the proportion of the tuples in the database which contain all of those value-attribute pairs to the total number of tuples in the database.

Definition 14 *Restricting Power of an attribute set $\{a_1..a_n\}$* ($\Re(\{a_1..a_n\}\mathrm{r})$)

The Restricting Power of an attribute set $\{a_1..a_n\}$ ($\Re(\{a_1..a_n\}, r)$), is some measure of the collection of restrictive powers of all of the value-attribute mappings which occur in the database for those attributes. (The measure used for a query on a database meeting Restrictions 1-4 is the mean, as the mean is the same as the restrictive power of each of the values in the active domain.)

To complete our set of definitions, a definition of the query set size of an attribute set is given:

Definition 15 *Query Set Size of an attribute set, $\mathcal{QSS}(A)$, w.r.t. database instance* r.

The query set size of an attribute set with respect to database instance r is some measure of the number of tuples which will be returned from the database when presented with a query in which the characteristic formula contains exactly that set of attributes. This value is calculated by multiplying some measure of the restricting power of an attribute set $\Re$(A, r) by the number of tuples in the database $\mid r \mid$. (The measure used for a query on a database meeting Restrictions 1-4 is the mean, as the mean is the same as the restrictive power of each of the values in the active domain.)

Four restrictions are now stated to allow a formal proof of memoryless inference methods previously presented in the literature (Table 1). Each restriction will be discussed, one will be lifted and two more will be added to allow a proof of a new method for memoryless inference control which is presented in this paper.

Table 1: Initial database restrictions (for characterizing attributes).

Restriction 1 : *Universal relation form of the database, U, is assumed.*

the database is represented by a single relation (file) with no null values, i.e. a universal form of the database is assumed formed with no null values.

Restriction 2 : *Equality of characterizing attribute active domain size.*

$(\forall i,j \mid a_i, a_j \in \mathbf{A_C(r)}, \mid adom(a_i, r) \mid = \mid adom(a_j, r) \mid)$.

Restriction 3 : *Uniform distribution of active domain values.*

$(\forall i,j \mid v_i, v_j \in a_k \wedge a_k \in U, \mid v_i \mid = \mid v_j \mid)$.

Restriction 4 : *All characterizing attributes are independent.*

$\forall a_i, a_j \in A_C$, *if* $\exists t_1, t_2 \mid t_1(a_i) = v_1 \wedge t_2(a_i) = v_2 \wedge t_1(a_j) = w_1 \wedge t_2(a_j) = w_2 \wedge v_1 \neq v_2 \wedge w_1 \neq w_2 \Rightarrow \exists t_3 \mid t_3(a_i) = v_1 \wedge t_3(a_j) = w_2$, *where* t_1, t_2, *and* t_3 are tuples in the database.

Restriction 1 (universal, no-nulls) is partially implied by Restrictions 2 and 3. True independence of attributes used in identifying tuples requires that each value in one domain must be associated with all values in another domain. The role of null values in this formulation is unclear at best. Null values are restricted at this juncture to simplify the logic; this problem will be addressed in later work.

Restriction 2 (domain size equality) simplifies the conceptualization of the query set restricting power of an attribute: If all attributes have the same domain size, then specifying a value for any attribute limits the cardinality of the query set to the same number no matter which attribute is chosen (assuming that Restriction 3 (uniform value distribution), also applies).

Restriction 3 (uniform value distribution) states that all domain values for a given attribute are represented equally in the database. Each value will thus be found in the same number of tuples. This is obviously an unrealistic assumption, especially when one thinks of an attribute such as city, where many more tuples in the census database will have a city of Chicago as opposed to tuples associated with the small town of Flush, Kansas.

Restriction 4 (independent attributes) simplifies calculations to determine the interaction of the restricting power of attributes. If two attributes a_i and a_j are independent (not just logically but also physically in the database), then the proportion of tuples in the database which have a value v_k for attribute a_i and a value v_l for attribute a_j can be calculated by multiplying the proportion of tuples which have the value v_k for attribute a_i by the proportion of tuples which that the value v_l for attribute a_j.

3 Description of a database meeting restrictions 1-4

Restrictions 1-4 applied to a database, r, guarantees that the number of tuples ($\mid r \mid$) present in the database with any given value of an attribute is equal for all values. Specifying a single value for a single attribute a_i in a query on such a restricted database reduces the number of tuples in the query set to $\frac{1}{\mid adom(a_i,r) \mid} \times \mid r \mid$. If single values for two attributes a_i and a_j are specified in a query, the size of the query set is reduced to $\frac{1}{\mid adom(a_i,r) \mid \times \mid adom(a_j,r) \mid} \times \mid r \mid$. The first factor in each of these calculations represents the restricting power of an attribute and attribute set and a proof of them is given in Properties 1 and 2.

The name restricting power comes from the fact that as they represent the proportion of the query set size to the original database size for a query on 1 or 2 attributes. The size of the query set can be thought of as a restriction on the original database size.

The database in Table 2 meets Restrictions 1-4 from Table 1. Using this database, assume that queries written using Eyecolor and Town as characterizing attributes. Restriction 1 (universal, no-nulls) is trivially satisfied. Eyecolor and town both have a domain size

Property 1 *Restricting power of a single attribute,* $\Re(\{a_i\}, \mathrm{r})$.

Given a database instance r*(R) meeting restrictions 1-4,* $\Re(\{a_i\}, \mathrm{r})$*, based on the measure of proportion, is* $\frac{1}{|adom(a_i, \mathrm{r})|}$*, where* $a_i \in R$.

Proof

1. The restricting power of an attribute a_i on some database instance r equals some measure of the set $\forall m \mid v_m \in adom(a_i, r) \frac{|\sigma_{a_i = v_m}(r)|}{|r|}$. (Note: σ is the relational algebra selection operator)
   Definition 11.
2. $\forall a_k \in R, and \forall v_i, v_j \in a_k, \mid \{t_m | t_m(a_k) = v_i\} \mid = \mid \{t_n | t_n(a_k) = v_j\} \mid$. (All values v_i, v_j for a single attribute a_k are equally represented in the database.)
   Restriction 3.
   Let n be the number of records containing value v_j for attribute a_i, where $v_j \in adom(a_i, r)$
3. $n = \frac{|r|}{|adom(a_i, r)|}$
   2, Restriction 1, definition of a database with no null values.
4. The proportion of records with a given value for a single attribute a_i is $\frac{n}{n \times |adom(a_i, r)|}$.
   2, 3, definition of proportion
5. The proportion of records with a given value for a single attribute a_i is $\frac{1}{|adom(a_i, r)|}$.
   4, reduction of a fraction.
6. The restricting power of a single attribute $\Re(\{a_i\}, r)$, based on the measure of proportion, equals $\frac{1}{|adom(a_i, r)|}$.
   1, 2, 5, substitution.

□

Property 2 *Restricting power of a set of two attributes,* $\Re(\{a_i, a_j\}, \mathrm{r}), i \neq j$.

Given a database instance r*(R) meeting restrictions 1-4,* $\Re(\{a_i, a_j\}, \mathrm{r})$*, based on the measure of proportion, is* $\frac{1}{|adom(a_i, \mathrm{r})|} \times \frac{1}{|adom(a_j, \mathrm{r})|}$*, where* $a_i, a_j \in R$.

Proof

1. The restricting power of a single attribute a_i is $\frac{1}{|adom(a_i, r)|}$.
   Property 1.
2. The attributes a_i and a_j are independent.
   Restriction 4.
3. The proportion of records containing both a value $v_k \in adom(a_i, r)$ for a_i and a value $v_l \in adom(a_j, r)$ for a_j is $\frac{1}{|adom(a_i, r)|} \times \frac{1}{|adom(a_j, r)|}$.
   2, definition of independence.
4. $\Re(\{a_i, a_j\}, r) = \frac{1}{|adom(a_i, r)|} \times \frac{1}{|adom(a_j, r)|}$.
   3, definition of restricting power.

□

Tuple no.	SSN	Eyecolor	Town	Age
1	1	Blue	Ann Arbor	22
2	2	Blue	Manhattan	44
3	3	Green	Ann Arbor	33
4	4	Green	Manhattan	55
5	5	Blue	Ann Arbor	21
6	6	Blue	Manhattan	45
7	7	Green	Ann Arbor	34
8	8	Green	Manhattan	56

Table 2: Sample database meeting restrictions 1-4.

of 2, thus the query meets Restriction 2 (domain size equality). Each value in the domains Eyecolor and Town is represented 4 times in the database, thus the database meets Restriction 3 (uniform value distribution). The Eyecolor and Town attributes are independent, as a projection without removal of duplicates is a multiple of the Cartesian product of Eyecolor and Town. The database thus satisfies Restriction 4 (independent attributes).

No tuple in this database can be uniquely identified by specifying a value for both Eyecolor and Town. However, if a database is formed from the first four or last four tuples of the database in Table 2 (either set of which still meets all four restrictions), specification of both of the mentioned attributes will **always** identify a single tuple. Similarly, the specification of Eyecolor and Town will always identify two tuples in the database in Table 2. In fact, the only databases which satisfy all four restrictions are databases for which all queries over the same attributes produce query sets with the same size.

Property 2, which gives a method for calculating the restricting power of two attributes in a database conforming to Restrictions 1-4, can easily be extended to hold for a set of n attributes to develop Property 3.

4 Relaxing Restriction 2

It is relatively easy to take domain size into account when calculating the restricting power of a set of attributes. To accomplish this adjustment, we remove Restriction 2 (domain size equality) from our set of restrictions, and accept Property 3 without further change. This can be done since each domain size is already represented separately even though the domain sizes were assumed to be equal up to this point.

Once we note that Property 3 does not demand that the domain sizes all be equal, we see that the method

Property 3 *Calculating the restricting power of a set of n attributes* $\Re(\{a_1, a_2, ..., a_n\}, r)$.

Given a database instance $r(R)$ *meeting restrictions 1-4,* $\Re(\{a_1, ..., a_n\}, r)$, *based on the measure of proportion, is the product:* $\frac{1}{|adom(a_1,r)|} \times \frac{1}{|adom(a_2,r)|} \times ... \times \frac{1}{|adom(a_n,r)|}$, *where* $a_1, a_2, ..., a_n \in R$.

Proof

1. The restricting power of a single attribute a_i is $\frac{1}{|adom(a_i,r)|}$.
   Property 1.
2. The attributes a_1, a_2,... a_n are independent.
   Restriction 4.
3. The proportion of records containing a value $v_{1k_1} \in adom(a_1, r)$ for a_1, a value $v_{2k_2} \in adom(a_2, r)$ for a_j, ..., and a value $v_{nk_n} \in adom(a_n, r)$ is: $\frac{1}{|adom(a_1,r)|} \times \frac{1}{|adom(a_2,r)|} \times ... \times \frac{1}{|adom(a_n,r)|}$.
   2, definition of independence.
4. $\Re(\{a_1, a_2, ..., a_n\}, r) = \frac{1}{|adom(a_1,r)|} \times \frac{1}{|adom(a_2,r)|} \times ... \times \frac{1}{|adom(a_n,r)|}$.
   3, definition of restricting power.

□

of calculating the restricting power of a set of attributes presented in Property 3 is basically equivalent to the maximum relative table size controls already noted in the literature[DS83].

It may be useful at this point to give some concrete examples of the calculation of restricting power for an actual database. Using the database represented in Table 2, we attempt to calculate the restricting power of a single attribute, Eyecolor. The domain size of Eyecolor is 2 (Blue, Green). Thus according to Property 1, the restricting power of the attribute Eyecolor is equal to $1/|\ adom(Eyecolor, r)\ |$, which in this example is 1/2. This means that if we specify a characteristic formula with a value given for the attribute Eyecolor, we will have a query set with 1/2 of the tuples, and a cursory examination of the database shows this to be true. Similarly, the restricting power of the attribute Town is also 1/2.

Property 3 is used to calculate the restricting power of both attributes Town and Eyecolor. The restricting power of the pair Town and Eyecolor is thus $1/2 \times 1/2$ or 1/4. Again, an examination of the database shows that choosing values for both attributes returns a query set of 2 tuples, which is 1/4 of the tuples in the database.

It is possible for the restricting power of a set of attributes to be greater than necessary to identify a single tuple in the database. If we take as an example the set of attributes Eyecolor, Town, Age, we obtain single attribute restricting powers of 1/2, 1/2, and 1/8, respectively for the database in Table 2. Using Property 1.3 to obtain the restricting power of the set, we get $1/2 \times 1/2 \times 1/8 = 1/32$, which would allow identification of a single tuple in a database of 32 tuples holding to Restrictions 1,3 and 4. The database in Table 2 holds to Restrictions 1,3 and 4 yet has only 8 tuples. In a case like this, we find that a subset of the set of attributes in the characteristic formula would have been sufficient to identify a single tuple.

Tuple No.	...	Eyecolor	Haircolor	...
1	...	Green	Blonde	...
2	...	Green	Black	...
3	...	Green	Red	...
4	...	Green	Brown	...
5	...	Brown	Blonde	...
6	...	Brown	Black	...
7	...	Brown	Red	...
8	...	Brown	Brown	...
9	...	Blue	Blonde	...
10	...	Blue	Black	...
11	...	Blue	Red	...
12	...	Blue	Brown	...
13	...	Gray	Blonde	...
14	...	Gray	Black	...
15	...	Gray	Red	...
16	...	Gray	Brown	...

Table 3: Sample database meeting restrictions 1-4.

5 Relaxing restriction 4

We now extend our analysis of the restricting power of attribute sets by removing Restriction 4 from the databases. We have not found evidence in the published literature that this type of analysis has been previously done.

A database is presented which meets all restrictions (1-4) (Table 3), and another database is presented which meets all restrictions **except** Restriction 4 (independent attributes) (Table 4). These are used to motivate and illustrate the relaxation of restriction 4.

The database in Table 3 meets all 4 restrictions. Restriction 1 (universal, no-nulls) is trivially met. Restriction 2 (domain size equality) is met since both attributes have an active domain size of 4. Restriction 3 (uniform value distribution) is met because all domain values for an attribute are equally represented in the database. Restriction 4 (independent attributes) is met because a projection of the attributes Eyecolor and Haircolor is a multiple of the power set of the attribute domains.

Tuple No.	...	Eyecolor	Haircolor	...
1	...	Green	Blonde	...
2	...	Green	Blonde	...
3	...	Green	Red	...
4	...	Green	Red	...
5	...	Brown	Black	...
6	...	Brown	Black	...
7	...	Brown	Brown	...
8	...	Brown	Brown	...
9	...	Blue	Blonde	...
10	...	Blue	Blonde	...
11	...	Blue	Brown	...
12	...	Blue	Brown	...
13	...	Gray	Black	...
14	...	Gray	Black	...
15	...	Gray	Red	...
16	...	Gray	Red	...

Table 4: Sample database with Restriction 2 lifted.

The database in Table 4 meets all restrictions except Restriction 4 (independent attributes). Restriction 1 (universal, no-nulls) is met because the database is a single relation without nulls. Restriction 2 (domain size equality) is met since both attributes have an active domain size 4 (each attribute has the same domain as the database in Table 3). Restriction 3 (uniform value distribution) is met because all domain values for an attribute are equally represented in the database. Restriction 4 (independent attributes) is **not** met, as each Eyecolor is only associated with 2 of the 4 Haircolors, and, conversely, each Haircolor is only associated with 2 of the Eyecolors.

Comparisons between these two databases will be made as the changes in properties which are necessitated through lifting Restriction 4 (independent attributes) are discussed.

Suppose there is a connection between attributes - say given Haircolor only 2 out of 4 Eyecolors are possible (selection of Haircolor reduces the domain of Eyecolor by $\frac{1}{2}$). Further suppose that there are 4 Haircolors, and each color is represented in 4 tuples in a database instance. Selection of the attribute Haircolor alone thus gives a $\mathcal{QSS}$(Haircolor, r) of 4.

If Haircolor and Eyecolor are independent then each Haircolor will be associated with each Eyecolor in 1 tuple (red hair/blue eyes - 1 tuples, etc.), thus 4 Haircolors $\times$ 4 Eyecolors $\times$ 1 tuples per combination = 16 tuples in the database. In this case selection of both Haircolor and Eyecolor will yield a $\mathcal{QSS}$ of 1 (calculated by using Property 2 to calculate the restricting power: $\frac{1}{4} \times \frac{1}{4} = \frac{1}{16}$). Thus QSS(Haircolor, Eyecolor) = $\frac{1}{16} \times \mid r \mid = \frac{1}{16} \times 16 = 1$.

If Haircolor and Eyecolor are dependent, and each Haircolor is associated with 2 Eyecolors, then the previous method of calculating QSS will be incorrect. In the database in Table 4, the $\mathcal{QSS}$ (Haircolor, Eyecolor) is 2, not 1. The reason for this lesser amount of $\mathcal{QSS}$ reduction is the dependence between the two attributes. When the reducing power of the second attribute is applied it is no longer operating on a set of tuples with the same properties as that set (the database) on which the reducing power was calculated. The query set resulting from selection of the first attribute has a smaller active domain size for the second attribute than was present in the original database. To correctly calculate the $\mathcal{QSS}$ first Property 2 must be altered to account for the dependence. This is accomplished by calculating the new active domain size for attributes 2 in the query set upon which the second attribute works. This can be calculated by noting the reduction effect that the first attribute has on the active domain of the second attribute ($\frac{2}{4}$ in this case) as:
$\Re(\{Haircolor, Eyecolor\}, r) = \frac{1}{4} \times (\frac{1}{4} \times \frac{1}{\frac{2}{4}}) = \frac{1}{4} \times \frac{1}{2} = \frac{1}{8}$
thus: $\mathcal{QSS}$({ Haircolor, Eyecolor }) = $\frac{1}{8} \times 16 = 2$.

The relatively decreased restricting power of a set of attributes upon the database from Table 4 when compared to the restricting power of the same set of attributes upon the database from Table 3 is the effect caused by the dependencies among the attributes. A key is another example of this effect. In this case, adding attributes to the key (thereby making a superkey) does not increase the restricting power of the key. This method of increasing the accuracy of the restricting power calculation for a query on a database which does not meet Restriction 4 is formalized in Property 4. Property 4 does not explicitly use calculations to adjust the active domain size of attributes when other attributes have already been selected. Instead, the database is recursively handed to each attribute in turn, the restricting power of each attribute being calculated based on the changed database the attribute receives. The recursion for calculating the database to be used runs out when a single attribute is left. For this single attribute the restricting power is calculated using Property 1 and is based on the original database. It is this changed database which is handed the next attribute to roll back the recursion until the restricting power of the final attribute has been determined.

It should be noted that some queries which returned a positive $\mathcal{QSS}$ when the attributes were independent will now return a $\mathcal{QSS}$ of zero. Some of these query sets are of size zero because they represent a combination of attribute values which does not exist in the real world (e.g. the combination of the town of New York and the State Wyoming). By using dependencies to lessen the effect of the reducing power of dependent attributes we are effectively decreasing the logical size of the tables used in the **Maximum relative table size controls** proposed in the work by Denning, Schlorer,

Property 4 *Alternate notation for the restriction power of a set of n attributes* $\Re(\{a_1, ..., a_n\}, \mathrm{r})$.

Given a database instance r*(R) meeting restrictions 1-4, an alternate notation for the restriction power of a set of attributes* $\{a_1, ..., a_n\}$, $\Re(\{a_1, ..., a_n\}, \mathrm{r})$, *based on the measure of proportion, is:*

$$\frac{|\sigma_{a_n}(\sigma_{a_1,...a_{n-1}}(\mathrm{r}))|}{|\sigma_{a_1,...a_{n-1}}(\mathrm{r})|} \times \Re(\{a_1, ..., a_{n-1}\}, \mathrm{r})$$

Proof

1. $\Re(\{a_1, ..., a_n\}, r) = \frac{|\sigma_{a_1,...,a_n}(r)|}{|r|}$
   Definition 4.
2. $\sigma_{a_1,...,a_n}(r) = \sigma_{a_n}(\sigma_{a_1,...,a_{n-1}}(r))$
   Property of the σ
   (select) operator.
3. $\frac{|\sigma_{a_1,...,a_n}(r)|}{|r|} = \frac{|\sigma_{a_n}(\sigma_{a_1,...,a_{n-1}}(r))|}{|r|}$
   1, 2, substitution
4. $\frac{|\sigma_{a_n}(\sigma_{a_1,...,a_{n-1}}(r))|}{|r|} =$
   $\frac{|\sigma_{a_n}(\sigma_{a_1,...,a_{n-1}}(r))|}{|r|} \times \frac{|\sigma_{a_1,...,a_{n-1}}(r)|}{|\sigma_{a_1,...,a_{n-1}}(r)|}$
   multiplicative identity
5. $\frac{|\sigma_{a_n}(\sigma_{a_1,...,a_{n-1}}(r))|}{|r|} =$
   $\frac{|\sigma_{a_n}(\sigma_{a_1,...,a_{n-1}}(r))|}{|\sigma_{a_1,...,a_{n-1}}(r)|} \times \frac{|\sigma_{a_1,...,a_{n-1}}(r)|}{|r|}$
   commutative property of multiplication
6. $\Re(\{a_1, ..., a_n\}, r) =$
   $\frac{|\sigma_{a_n}(\sigma_{a_1,...,a_{n-1}}(r))|}{|\sigma_{a_1,...,a_{n-1}}(r)|} \times \Re(\{a_1, ..., a_{n-1}\}, r)$
   1, 5, substitution

□

and Wehrle[DS83]. Property 4 uses these logical table sizes in its calculation of restricting power.

A problem is inherent in the method of calculating restricting power presented in Property 4: the calculation is dependent on sets of tuples which are only known at calculation time. This problem can be avoided, however, if we place two additional restrictions on the database. These restrictions concern databases which have had Restriction 3 lifted, though databases for which Restriction 3 holds also hold to these additional restrictions. Restrictions 5 and 6 are given in Table 5.

These additional restrictions allow the use of stored information to calculate the various restricting powers used in Property 4. This information would be pulled from the database, and would be stored in the form of dependencies. We call these dependencies Domain Reduction Dependencies (DRD) and they provide a measurement of the dependency between sets of attributes. If databases are close to satisfying Restrictions 1, 3, 5 and 6, the dependencies should be useful for calculating a reasonable approximation of the restricting power. A full definition of Domain Reduction Dependencies and presentation of their properties will be the subject of a future paper.

Table 5: Two Additional Database Restrictions.

Restriction 5 : *All values from an attribute* a_i*'s active domain are associated with the same number of values from the active domain of attribute* a_j*'s active domain, where* a_i *and* a_j *are both characterizing attributes* $(a_i, a_j \in A_C)$ *in the database.*

Let R be a relation scheme, and r(R) be an instance of that database. Further let A_C be the set of characterizing attributes of relation scheme R. Then, $\forall i, j$ such that $a_i, a_j \in \mathbf{A_C(r)}$, and $\forall k, m$ such that $v_k, v_m \in a_i$, $| \pi_{a_j}(\sigma_{a_i=v_k}(r)) | = | \pi_{a_j}(\sigma_{a_i=v_m}(r)) |$.

Restriction 6 : *All query sets from database* r *have attributes with uniform representation of all values in the active domain within the query set.*

$\forall$ query sets $q_s(a_1...a_n)$ on r, restriction 2 holds.

The means of calculating the restricting properties of groups of attributes from queries on databases meeting Restrictions 1, 3, 5 and 6 is presented in Property 5.

6 Conclusion

In this paper further development of a memoryless inference control method is provided. In addition, a formalization of the restrictions on a database instance is given. These restrictions must hold true if the inference control technique is to be proven to provide an accurate measure of the identification power of a query set. We present formal definitions of the restriction power of single attributes and attribute sets. Several methods of calculation of the restriction power of a set of attributes are developed and shown to hold for database instances which meet sets of restrictions also given. As a part of these calculation methods, previous work is extended by presenting a method for calculating the restricting power of a query while taking into account dependencies among the characterizing attributes of the query.

Property 5 *The restrictive power of a set of attributes can be calculated by multiplying the restrictive power of each single attribute by some measure of the dependence of the values in that attribute on the attributes which have already been applied to the database to make the selection.*

Given a database instance r*(R) meeting restrictions 1, 3, 5 and 6,* $\Re(\{a_1,...,a_n\},\mathrm{r})$*, is* $\frac{1}{|adom(a_n,\mathrm{r})|} \times \frac{1}{\alpha} \times \Re(\{a_1...a_{n-1}\},\mathrm{r})$*, where* α *is the dependence between the attributes in the group* $\{a_1,...,a_{n-1}\}$ *and the attribute* a_n*. This dependence is measured as the proportion* $\frac{adomsize(a_n,\sigma_{a_1,\ ,a_{n-1}}(\mathrm{r}))}{adomsize(a_n,\mathrm{r})}$.

Proof

1. $\Re(\{a_1...a_n\},r) = \frac{|\sigma_{a_n(\sigma_{a_1,\ a_{n-1}}(r))}|}{|\sigma_{a_1,\ a_{n-1}}(r)|} \times \Re(\{a_1,...,a_{n-1}\},r)$
   Property 4.
2. $\forall s = \sigma_{a_1,...,a_h}(r)$, $\forall a_k \in R$, and $\forall v_i, v_j \in a_k$, $|\{t_m \in s | t_m(a_k) = v_i\}| = |\{t_n \in s | t_n(a_k) = v_j\}|$
   All values v_i, v_j for a single attribute a_k are equally represented in all elementary subsets of the database (subsets returned as query sets where each query specifies no more than a single value for each attribute specified in the query).
   Restrictions 3, 5, and 6.

Let n be the number of records containing value v_j for attribute a_i, where $v_j \in adom(a_i, s)$, where s is a query set of the database r such that no more than a single value is specified for any attribute specified in the query.

definition

3. $n = \frac{|s|}{|adom(a_i,s)|}$
   1, Restriction 1, definition of a database with no null values.
4. $n = \frac{1}{|adom(a_i,s)|} \times |s|$
   multiplicative identity, commutative property of multiplication
5. $|\sigma_{a_j}(s)| = \frac{1}{|adom(a_i,s)|} \times |s|$
   definition of n, definition of selection operation on relational databases, substitution
6. $|\sigma_{a_j}(\sigma_{a_1,..,a_{n-1}}(r))| = \frac{1}{|adom(a_i,(\sigma_{a_1,\ ,a_{n-1}}(r))|} \times |\sigma_{a_1,..,a_{n-1}}(r)|$
   definition of s*, substitution*
7. $\frac{|\sigma_{a_j}(\sigma_{a_1,...,a_{n-1}}(r))|}{|\sigma_{a_1,..,a_{n-1}}(r)|} = \frac{1}{|adom(a_i,(\sigma_{a_1,\ ,a_{n-1}}(r))|}$
   multiplying both sides of an equality by a common factor
8. $\frac{|\sigma_{a_n(\sigma_{a_1,\ a_{n-1}}(r))}|}{|\sigma_{a_1,\ a_{n-1}}(r)|} = \frac{1}{adom(a_n,\sigma_{a_1\ a_{n-1}}(r))} \times \frac{1}{\frac{adom(a_n,r)}{adom(a_n,r)}}$
   multiplicative identity
9. $\frac{|\sigma_{a_n(\sigma_{a_1,\ a_{n-1}}(r))}|}{|\sigma_{a_1,\ a_{n-1}}(r)|} = \frac{1}{adom(a_n,r)} \times \frac{1}{\frac{adom(a_n,\sigma_{a_1\ a_{n-1}}(r))}{adom(a_n,r)}}$
   commutative property of multiplication
10. $\Re(\{a_1...a_n\},r) = \frac{1}{adom(a_n,r)} \times \frac{1}{\frac{adom(a_n,\sigma_{a_1\ a_{n-1}}(r))}{adom(a_n,r)}} \times \Re(\{a_1,...,a_{n-1}\},r)$
    1, 5, substitution.

□

References

[AW89] Nabil R. Adam and J.C. Wortmann. Security-control methods for statistical databases: A comparative study. *ACM Computing Surveys*, 21(4):515–556, December 1989.

[DS83] D.E. Denning and J. Schlorer. Inference controls for statistical databases. *Computer*, 16(7):69–82, July 1983.

[DSW84] D.E. Denning, J. Schlorer, and E. Wehrle. Memoryless inference controls for statistical databases. Technical report, Computer Science Department, Purdue Univ., 1984.

[Sch75] J. Schlorer. Identification and retrieval of personal records from a statistical data bank. *Methods of Information in Medicine*, 14(1):7–13, January 1975.