Similarity Measures for Multidimensional Data

Eftychia Baikousi, Georgios Rogkakos, Panos Vassiliadis

[1] Dept. of Computer Science, University of Ioannina
Ioannina, 45110, Hellas
{ebaikou, grogkako, pvassil}@cs.uoi.gr

25-01-2010

Abstract. How similar are two data-cubes? Due to the great amount of data stored nowadays, it is fundamental to provide similarity measures within sets of multidimensional data. In this paper we explore various distance functions that can be used over OLAP cubes. We organize the discussed functions with respect to the properties of the dimension hierarchies that they exploit. For the purpose of discovering which distance functions are more suitable and meaningful to the users, we conducted a user study analysis. Our findings indicate that the functions that seem to fit better the user needs are characterized by the tendency to consider as closest to a point in a multidimensional space, points with the smallest shortest path with respect to the same dimension hierarchy.

Keywords: Similarity measures, OLAP.

1 Introduction

How similar are two data-cubes? To put the question a little more precisely, given two sets of points in a multidimensional hierarchical space, what is the distance between these two collections? The above research problem is generic and has several applications in domains such as multimedia information retrieval, statistical data analysis, scientific databases and digital libraries [ZADB06]. In such applications, where contemporary data lead to huge repositories of heterogeneous data stored in data warehouses, there is a need of similarity search that complements the traditional exact match search. For example, one might easily envision a context where a user of an OLAP tool is proactively informed on reports that are similar to the one she is currently browsing.

In this paper, we address the problem by (a) exhaustively organizing alternative distance functions in a taxonomy of functions and (b) experimentally assessing the effectiveness of each distance function via a user study. Our approach is structured as follows: We start (Section 2) with the formal foundations of modeling multidimensional spaces and cubes based on an existing model in the related literature [VaSk00]. Then (Section 3), we provide a taxonomy of distance functions for cubes based on a detailed study of the characteristics of dimension hierarchies, levels and members. Specifically, we organize our families of functions as follows: Initially we describe functions that can be applied between two specific values that belong in the same level of hierarchy within a given dimension and secondly we describe distance functions that can be applied between two values from different levels of hierarchy. Following, we describe distance functions that are applied between two cells of a cube and then distance functions between two OLAP cubes.

So far, related work has dealt with similar problems in different ways; however, this particular problem has not been dealt per se. Specifically, Sarawagi in [Sara99] and [Sara00] has dealt with the problem of discovering interesting patterns and differences within two instances of an OLAP cube. The DIFF and RELAX operators summarize the difference between two sub-cubes in order to discover the reason of abnormalities within the measures of two given cells. The only common factor of this work with ours is the usage of the Manhattan distance function in the procedure of discovering abnormalities. Our work addresses the problem of finding the appropriate distance function among a great variety of functions in order to compute the similarity between two given OLAP cubes. Giacometti et. al. [GMNS09] propose a recommendation system for OLAP queries by evaluating distances between multidimensional queries. This work involves the distance between queries whereas our work involves distance functions between the data of multidimensional queries. Li et.al. in [LiBM03] describe the semantic similarity between ontologies. In contrast to our work, they consider a limited set of functions whereas we have a wider range of distance functions and our work focuses on distances between data in the multidimensional space.

The main findings of our approach are due to a user study that we have conducted to assess which distance functions appear to work better for the users (Section 4). The experiment involved 15 users of various backgrounds and the *Adult* real dataset [FuWY05]. Each user was given 14 scenarios that contained a reference cube as well as a set o variant cubes, each associated with a distance function. The task of the user was to select a cube from the set of variant cubes that seemed more similar to the reference cube. The diversity of users and data types contained in the experiment was taken into consideration in order to discover which distance function is preferred depending on the user group or the type of data. The user study we conducted showed that all distance functions under test were used at least once, but there were a couple of distance functions that were most preferred among the others. In particular, the users seemed to prefer distance functions that express the similarity between two cubes based on the hierarchical shortest path or in regards to ancestor values.

2 Modeling Foundations

One of the main factors in database research is the retrieval of useful information from data that are stored under a structured collection of records. OLAP tools are based on a multidimensional view of data, where analysts may powerfully perform aggregates of data in various ways and extract useful information. In this section we provide some basic insights of the way data are stored and organized under the form of OLAP cubes.

The theoretical foundations for modeling multidimensional spaces, dimensions, hierarchies and data cubes are based on the premises of [VaSk00].

Definition 1 (dimension). A dimension D is a lattice (L, $\prec$) such that: L= (L_1, ..., L_n, *ALL*) is a finite subset of levels and $\prec$ is a partial order defined among the levels of L, such that $L_1 \prec L_i \prec ALL$ for every $1<i\leq$n. We require that the upper bound of the lattice is the level *ALL*, so that we can group all the values of the dimension into the single value 'all'. The lower bound of the lattice is called the detailed level of the dimension.

Each dimension has an associated hierarchy of levels of aggregated data. In addition, for every level L_i there is a domain of values denoted as *dom*(L_i). Therefore, for every dimension D_i the domain is denoted as $DOM(D_i)=\bigcup_{j=1}^{m} dom(L_j)$ which states that it is the union of the domains of every level of hierarchy of the specific dimension.

Definition 2 (Cube). A cube c over the schema [L_1, …L_n, M_1, …,M_m], is an expression of the form: c= (DS^0, φ, [L_1, …L_n, M_1, …M_m], [agg_1(M_1^0, …, agg_m(M_m^0)]), where DS^0 is a detailed data set over the schema S= [L_1^0, …L_n^0, M_1^0, …M_m^0], $m\leq k$, φ is a detailed selection condition, M_1^0, …M_m^0 are detailed measures, M_1, …,M_m are aggregated measures, L_i^0 and L_i are levels such that $L_i^0 \prec L_i$, $1<i\leq$n and agg_i, $1<i\leq$m are aggregated functions from the set {*sum*, *min*, *max*, *count*}.

The relationship between values of different levels of hierarchy can be achieved via ancestor functions and their inverse descendant relationships defined as follows:

$anc_{L_i}^{L_j}$ is a function that assigns a value from the domain of L_i to a value from the domain of L_j , where $L_i \prec L_j$. The relationship $desc_{L_1}^{L_2}$ is the inverse of the $anc_{L_1}^{L_2}$ function i.e., $desc_{L_1}^{L_2}(1)=\{x\in dom(L): anc_{L_1}^{L_2}(x)=1\}$.

According to the type of values that a dimension level may have we can classify the distance functions that can be applied. Thus, we categorize the dimension levels according to the values of their domain as following.

1. A dimension level is *Nominal* when its values hold the distinctness property. In other words, the values in such a dimension can be explicitly distinguished.
2. A dimension level is *Ordinal* when its values hold the distinctness property as well as the order property. The order property implies that the values of such a dimension abide by an order.
3. A dimension level is *Interval* when its values apart from the distinctness and order property also have the addition property. The addition property states that a unit of measurement exists. The difference between two values has a meaning, indicating how many values intermediate between them.
4. A dimension level is *Ratio* when its values apart from the distinctness, order and addition property also satisfy the multiplication property. The multiplication property states that differences and ratios between values have a meaning. In other words, the ratio between two values indicates their analogy difference expressed in a percentage scale.

3 Distance Functions

In this section, we organize the distance functions that can be used to measure the distance between two cubes. We build our taxonomy progressively: In section 3.1 we describe the distance functions that can be applied between two values that belong in the same level for a given dimension. In section 3.2 we describe the distance functions that can be applied for two values that are in different levels, again from a given dimension. In section 3.3 and 3.4 we provide a taxonomy for distance functions between two cells of a cube and between two OLAP cubes. Throughout all our deliberations we will refer to two reference dimensions, *Time* and *Location*. The hierarchies of these dimensions are shown in figure 1(a). In more detail, the Time dimension hierarchy consists of 5 levels. The levels of *Time* are *Day* (L_1), *Week* (L_2) and *Month* (L_2), *Year* (L_3) and *All* (L_4). The dimension *Location* consists of four levels of hierarchy which are *City* (L_1), *Country* (L_2), *Continent* (L_3) and *All* (L_4). In figure 1(b) we illustrate the lattice of the dimension *Location* at the instance level.

3.1 Distance Functions Between Two Values in the Same Level of Hierarchy

In this section we specify the distance functions we can apply over two specific values belonging in the same level of hierarchy for any dimension D. Assume a specific dimension D, its lattice of level hierarchies $L_1 \prec L_2 \prec \ldots \prec ALL$, and two specific values x and y belonging in the domain of L_i, i.e. $x, y \in dom(L_i)$. We classify the distance functions that can be applied to two values of the same level in two categories: (a) locally computable and (b) hierarchical computable distance functions.

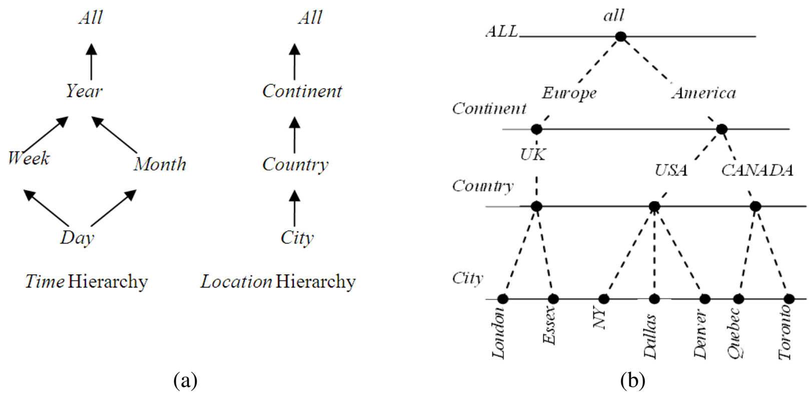


Fig. 1. (a) The hierarchy of levels for dimensions Time and Location (b) Values of the Location dimension

3.1.1 Locally Computable Distance Functions. The first category of locally computable distance function can be divided into three subcategories: (a) Distance functions with explicit assignment of values, (b) Distance function based on attribute values and (c) Distance functions based on the values of x and y.

Distance Functions with Explicit assignment of values. The functions of this category explicitly define n^2 distances for the n values of the *dom* (L_i). This requires that the n values are known and that *dom* (L_i) is a finite set. For example, assume a case where the distance between two cities is explicitly defined via a distance table.

Distance Functions based on Attribute Values. Assume a level whose instances are accompanied with a set of attributes. Then every level instance can be described as a tuple of attribute values. In this case, the distance between the two values x and y can possibly be expressed with respect to their attribute values via simple distance function applicable to the attributes' domains (e.g., simple subtraction for arithmetic values).

Distance functions based on the values x and y. In this subcategory, the distance between two values may be expressed through a function of their actual values whenever this is possible. In this subcategory one option is to make use of the simple identity function for nominal values. Thus, a value from the set {0, 1} where

$$\text{dist}(x, y) = \begin{cases} 0, \text{ if } x = y \\ 1, \text{ if } x \neq y \end{cases}$$

Another option is to make use of the Minkowski family distance functions for arithmetic values. In this section, since the distance function is applied for two specific values, all types of Minkowski distances reduce to the Manhattan distance which is $|x\text{-}y|$. As an example, consider the dimension *Time* whose levels are shown in figure 1(a). Assume two instances x and y from the level *Year*, where x= '1995' and y= '2000'. Then the distance between these two values is obviously |1995-2000| = 5. In order to normalize this distance function within the interval [0, 1], we can divide the distance value with the difference between the maximum and minimum values of the level where x and y belong in.

3.1.2 Hierarchical Computable Distance Functions The second category of hierarchical computable distance functions can be divided into three subcategories: (a) Distance functions expressed with respect to a lower level of hierarchy, (b) Distance functions expressed with respect to hierarchy path and (c) Highway distance functions.

Distance functions expressed with respect to a lower level of hierarchy. For any two values in the same level other than the detailed level L_1, the distance can be expressed with respect to an aggregation function (e.g., *count*, *max*, *min*) applied over the descendants of the two values in a lower level of hierarchy.

Assume an instance x from level L_i and $desc_{L_L}^{L_i}(x)$ be the set of its descendants, where L_L is any lower level of L_i. The result of applying an aggregation function over the set $desc_{L_L}^{L_i}(x)$ is denoted as x_{aggr}, i.e., $x_{aggr} = f_{aggr}(desc_{L_L}^{L_i}(x))$. Depending on the aggregation function applied, x_{aggr} can be the summation of the values in the set of descendants $desc_{L_L}^{L_i}(x)$, the minimum value from this set or simply the cardinality of this set. Assume two values x and y from a level L_i, x_{aggr} and y_{aggr} denote the results of applying an aggregation function f_{aggr} over the set of descendants of x and y respectively. Therefore, $x_{aggr} = f_{aggr}(desc_{L_L}^{L_i}(x))$ and $y_{aggr} = f_{aggr}(desc_{L_L}^{L_i}(y))$, where L_L could be any lower level of L_i, $x, y \in L_i$ and f_{aggr} denotes an aggregation function such as *count*, *min*, *max*, *sum or avg*. The distance between the values x and y can now be expressed according to the following formula: $dist(x, y) = g(x_{aggr}, y_{aggr})$, where the function g depend on the function f_{aggr}. For instance, if f_{aggr} returns a value of interval type such as when the function f_{aggr} is *count*, then the values x_{aggr} and y_{aggr} are of interval type and thus the function g may be the Minkowski distance. If the result of the function f_{aggr} is a specific value from the descendants, such as when f_{aggr} is *min* or *max*, then the function g may be any one from the locally computable function. In case that x and y belong in the detailed level L_1, then the $dist(x, y)$ may become any distance function from the previous category of locally computable section. The normalized form of this function, within the interval [0, 1], can be expressed as $dist(x, y) = \frac{g(x_{aggr}, y_{aggr})}{max\{g(a_{aggr}, b_{aggr})\}}$, where a and b are any possible values from the same level of hierarchy as x and y, i.e., $a, b \in L_i$.

Distance Functions expressed with respect to Hierarchy Path. The distance between two values x and y can be expressed according to the length of the path in the hierarchy that connects them. Several distance functions and combinations falling into this subcategory where described by Li, Bandar and McLean in [LiBM03]. Here, we describe the possible distance functions that can be applied between two values x and y from a hierarchy, (a) with respect to the length of the path in the hierarchy, and, (b) with respect to the depth in the hierarchy path. Assume two values x and y such that $x, y \in L$. We denote the lowest common ancestor of x and y as *lca*.

Definition 3 (lowest common ancestor). The lowest common ancestor *lca*, of two values x and y where $x, y \in L$, $lca \in L_z$ and L_z is any higher level of L, $L_z \succ L$ is a value such that:

$$\begin{aligned} lca = \{z \mid z = anc_L^{L_z}(x) \wedge\ z = anc_L^{L_z}(y)\ \ \wedge(\neg\exists z' \text{ s.t.} \\ z' = anc_L^{L_z}(x) \wedge\ z' = anc_L^{L_z}(y) \wedge L_{z'} \prec L_z\} \end{aligned} \tag{1}$$

The distance between the values x and y can be expressed with one of the following formulas:

1. $dist(x, y) = f_{path}(w_x * |path(x, lca)| + w_y * |path(y, lca)|)$

2. $dist(x, y) = f_{\mathrm{depth}}$ (|$path$ (lca, L_1) |)

The first formula indicates that the distance is a function of the weighted sum of the length of the path from the values x and y to their lowest common ancestor lca respectively. The second formula indicates that the distance of the values is expressed as a function of the length of the path of the lowest common ancestor lca from the detailed level L_1 of the hierarchy. In both formulas the functions f_{path} and f_{depth} may be any linear or exponential function such as $f(x) = e^{a*x}$, where a is real parameter. Moreover, these two functions are normalized in the interval [0, 1] by making use of the height of the hierarchy. Specifically, the first formula should be divided by $(w_x + w_y)*|path(ALL, L_1)|$ whereas the second formula is divided by $|path(ALL, L_1)|$. As an example, assume two values x = 'USA' and y = 'Canada' from the level *Country* of the hierarchy *Location* denoted in figure 1(b) where their lowest common ancestor is the value lca= 'America' from the level *Continent*. For simplicity, assume functions f_{path} and f_{depth} are the identity function and that the weighted factors w_x and w_y take the value 2, since x and y belong in the second level of hierarchy. Therefore, the functions are of the form: $f_{\mathrm{path}} = (2 * |path\ (x, lca)| + 2 * |path\ (y, lca)|)/ 4*|path(ALL, L_1)|$ and $f_{\mathrm{depth}} = |path\ (lca, L_1)|/ |path(ALL, L_1)|$ respectively. The distance between x and y occurs to be f_{path}= 2*(1+1)/4*3 =1/3 and f_{depth}= 2/3 respectively.

Highway Distance Functions. Highway distance functions can be applied in cases where the values of level L can be grouped into k groups, where each group has a representative. Then, the distance between any two representatives can be thought of as a highway. We denote with $r(x)$ and $r(y)$ the representatives of the groups where x and y belong in respectively. Therefore, the distance of two values in the level L can be expressed with the following formula:

$$dist\ (x, y) = dist\ (x, r(x)) + dist\ (r(x), r(\ y)) + dist\ (y, r(y)) \qquad (2)$$

The distance between the two representatives $r(x)$ and $r(y)$, can be computed through a function from the previous subsection of locally computable functions[1]. However, the distance between a value and its representative depends on the way the representative is selected. There are several ways of selecting the representative of a group. Some of them are:

1. Explicit assignment of the representative in each group.
2. The representative of a group is an ancestor of x. The representative could be $r(x) = anc_{L_x}^{L_u}(x)$ where L_u is any upper level of L_x. Then, in general, the distance between a value x and its representative may be computed through any distance function between two values in different levels of hierarchy (described in the following section). In the special case where the representatives $r(x)$ and $r(y)$ coincide in being the lowest common ancestor denoted as lca, the formula is simplified as: $dist\ (x, y) = dist\ (x, lca) + dist\ (y, lca)$. As an example, assume two values x = 'UK' and y = 'USA' from the level *Country* of the hierarchy *Location* denoted in figure 1(b). In addition, assume the representative $r(x)$ = 'Europe' and

[1] In case the two representatives belong in different levels their distance may be computed by applying any distance function from the following section that describes the possible distances between values in different levels of hierarchy.

the representative *r*(*y*) = 'America'. Then, the distance of the values *x* and *y* is by summing the distances *dist* ('Greece', 'Europe'), *dist* ('Europe', 'America') and *dist* ('America', 'USA').

3. The representative of a group is a descendant of *x*. The representative of a group can be selected with respect to the descendants of the group where *x* belongs in. The representative *r*(*x*) can be (a) a value from the domain of L_L (i.e., when *r*(*x*) is picked explicitly from the set $desc_{L_x}^{L_L}(x)$ or by applying an aggregation function of the form *min* or *max* over the set $desc_{L_x}^{L_L}(x)$), or, (b) an arithmetic type value (i.e., when an aggregation function of the form *sum* or *count* is applied over the set $desc_{L_x}^{L_L}(x)$). As an example, consider countries and their cities where the representative of a country can be selected among its cities based for instance on the major airport or the highest population. According to the way the representative *r*(*x*) is selected, the distance between *x* and *r*(*x*) can be (a) any function described in the following section, distances between two values in different levels of hierarchy, or, (b) any simple arithmetic function (or any other function from the distance functions based on attribute values) respectively.

3.2 Distance Functions between Two Values in Different Levels of Hierarchy

In this section we specify the distance functions we can apply over two specific values belonging in different levels of hierarchy for any dimension *D*. Assume a specific dimension *D* from the cube accompanied with its lattice of level hierarchies $L_1 \prec L_2 \prec \ldots \prec ALL$ and two specific values denoted as *x* and *y* from different levels of hierarchy, L_x and L_y respectively. In order to define the distance between these two values we have to define partial distances. For the following we denote with *dist*(*a*, *b*) the distance of any two values *a*, *b*. Without loss of generality assume $L_x \prec L_y$. In general, the distance between two values in different levels of hierarchy can be visualized through a number of various paths as shown in figure 2.

Assume the ancestor of *x* in level L_y denoted as $x_y = anc_{L_x}^{L_y}(x)$. In addition, assume a representative of *y* in the level of hierarchy L_x denoted as $y_x = f(desc_{L_x}^{L_y}(y))$. The function *f* applied over the descendants of *y* can result either to an explicitly assigned descendant or to the result of an aggregation function (e.g., min, max) over the set of descendants. We categorize the distance functions according to the paths that might be followed in (a) distance functions expressed with respect to an ancestor and (b) distance functions expressed with respect to a descendant.

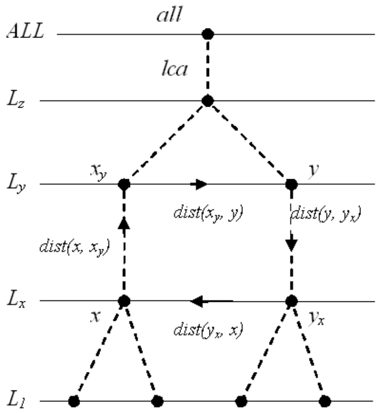


Fig. 2. Partial distances between two values in different levels of hierarchy.

3.2.1 Distance Functions Expressed with Respect to an Ancestor. In this category, the distance between values from different levels may be expressed with respect to an ancestor, via one of three possible ways.

1. The distance function is expressed in regards to x_{y}. In this case the distance is expressed as a summation of $dist(x, x_{\mathrm{y}})$ and $dist(x_{\mathrm{y}}, y)$, as shown in figure 2. Formally this is expressed as:

   $$dist(x, y) = dist(x, x_{\mathrm{y}}) + dist(x_{\mathrm{y}}, y) = dist(x, anc_{L_{\mathrm{x}}}^{L_{\mathrm{y}}}(x)) + dist(anc_{L_{\mathrm{x}}}^{L_{\mathrm{y}}}(x), y)$$

   where $dist(anc_{L_{\mathrm{x}}}^{L_{\mathrm{y}}}(x), y)$ is a function from the section *distance between two values from the same level of hierarchy*. As for $dist(x, x_{\mathrm{y}})$, this may be expressed through a function from the path family or the percentage family functions. In the special case where y is the ancestor of x, thus the values x_{y} and y coincide, the distance is then simplified as $dist(x, y) = dist(x, x_{\mathrm{y}})$. Since $dist(x, x_{\mathrm{y}})$ and $dist(x_{\mathrm{y}}, y)$ are within the interval [0, 1], the normalized form of dist(x, y) is simply by dividing it with 2. As an example, assume the dimension *Location* where its levels of hierarchy are seen in figure 1(a). The values of this dimension are seen in figure 1(b). Assume the two values x = 'USA' and y = 'Europe' and the ancestor x_{y} of x is $anc_{Country}^{Continent}(x) = \text{'America'}$. Assume $dist(x, x_{\mathrm{y}})$ is computed via a function from the percentage family functions. $dist(x_{\mathrm{y}}, y)$ is computed through the first formula from the path family functions where the weighted factors w_{x} and w_{y} are set to 1. Consequently, the distance between x and y becomes *dist*(*'USA'*, *'Europe'*)= ($dist(x, x_{\mathrm{y}})$ + $dist(x_{\mathrm{y}}, y)$)/2 = (*dist*(*'USA'*, *'America'*) + *dist*(*'America'*, *'Europe'*))/2 = (1/2 + 2/3)/2 = 7/12.
2. The distance function is expressed with respect to the lowest common ancestor of x and y. One option is to express this distance via a function from the path family. Specifically, the distance may be expressed as:

1. $dist(x, y) = f_{\text{path}}\left(\dfrac{w_{\text{x}} *|\, path\,(x, lca)| + w_y *|\, path\,(y, lca)|}{(w_{\text{x}} + w_{\text{y}})*|\, path(ALL, L_1)|}\right)$
2. $dist(x, y) = f_{\text{depth}}\left(\dfrac{|\, path\,(\, lca, L_1)|}{|\, path(ALL, L_1)|}\right)$

Another option is to express the distance function as:

$dist(x, y) = (dist(x, lca) + dist(lca, y))/2 = (dist(x, anc_{L_{\text{x}}}^{L_{\text{z}}}(x)) + dist(anc_{L_{\text{y}}}^{L_{\text{z}}}(y), y))/2$

where L_{z} denotes the level of hierarchy that *lca* belongs in and the denominator is set to 2 for normalization reasons. In this case, the distance from a value and the lowest common ancestor *lca*, thus $dist(x, lca)$ and $dist(lca, y)$, can be expressed by using a function from the percentage family. As an example, assume again the dimension *Location* whose values over the lattice are shown in figure 1(b). Assume the values *x* = 'NY' and *y*= 'Canada', their lowest common ancestor is *lca* = 'America'. In addition assume $dist(x, y)$ is calculated from the linear expression of the first formula of the hierarchy path, where the weighted factors w_{x} and w_{y} are set to 1. Then the distance is expressed as:

$dist(\text{'NY'}, \text{'Canada'}) =$

$\dfrac{|\, path(\text{'NY'}, \text{'America'})| + |\, path(\text{'America'}, \text{'Canada'})|}{2*|\, path(ALL, L_1)|} = \dfrac{2+1}{2*3} = 0.5$

3. Percentage family functions. According to this subcategory, the distance between two values *x* and *y*, where *y* is an ancestor of *x*, may be expressed according to a percentage of occurrences over the values of the hierarchy. In other words, the similarity of two values is expressed as the similarity of the number of descendants this two values have. Assume the lattice of level hierarchies be denoted as $L_1 \prec \ldots \prec L_{\text{L}} \prec L_{\text{x}} \prec L_{\text{y}} \prec$ *All* where L_1 denotes the most detailed level. The distance of a value *x* in a level L_x in regards to its ancestor *y* in level L_{y} may be calculated according to one of the following functions:

$$dist(x, y) = \frac{|\, desc_{L_{\text{i}}}^{L_{\text{x}}}(x)|}{|\, desc_{L_{\text{i}}}^{L_{\text{y}}}(y)|}\text{, where } L_{\text{i}} \text{ is one of the levels } L_{\text{x}}, L_{\text{L}} \text{ and } L_1 \qquad (3)$$

The above formula expresses the distance between a value *x* and one of its ancestors *y* as a percentage via three ways. In case L_{i} is L_{x}, then the distance is expressed as a percentage in regards to the occurrences of all the other values from L_{x} whose ancestor is *y*. In case L_{i} is L_{L}, the distance is expressed as a percentage of occurrences of the descendants of *x* in a lower level of hierarchy L_{L} in regards to the descendants of *y* in the same lower level L_{L}. In case the lower level is the detailed level L_1, then the distance is expressed as a percentage of occurrences of the descendants of *x* in L_1 in regards to the descendants of *y* in L_1. As an example assume the dimension *Location* where its lattice can be visualized in figure 1(a) and the values of this dimension visualized in figure 1(b). Assume the values *x* = 'USA' and

y= ‘America’. Then, in regards to the above formula the distance between these two values can be computed as:

i. $dist(\text{'USA'},\text{'America'}) = \frac{1}{|\,desc_{Country}^{Continent}(\text{'America'})\,|} = \frac{1}{2}$ where L_i is chosen to be the level L_x, i.e., $L_{country}$

ii. $dist(\text{'USA'},\text{'America'}) = \frac{|\,desc_{City}^{Country}(\text{'USA'})\,|}{|\,desc_{City}^{Continent}(\text{'America'})\,|} = \frac{3}{5}$ where L_i is chosen to be the detailed level L_1, i.e., L_{city}

As for the third case, in this example it coincides with the second since the lower and detailed level, i.e. *City*, are identical.

3.2.2 Distance Functions Expressed with Respect to Descendants. In this category the distance between two values in different levels of hierarchy may be expressed in regards to their descendants according to the following subcategories.

1. The distance function is expressed in regards to a representative from the descendants of y. The distance between x and y can be expressed by adding the distances $dist(y, y_x)$ and $dist(y_x, x)$ as shown in figure 2 This can be defined through the formula:

$$dist(x, y) = \frac{dist(y, y_x) + dist(y_x, x)}{2} = \frac{dist(y, f(desc_{L_x}^{L_y}(y))) + dist(f(desc_{L_x}^{L_y}(y)), x)}{2}$$

where the function f returns a descendant from the set of descendants $desc_{L_x}^{L_y}(y)$ and again the denominator is set to 2 for normalization reasons. The distance between the value x and y_x may be computed through a function from the section *distance between two values from the same level of hierarchy*. The distance $dist(y, y_x)$ can be calculated via a function from the path or percentage family functions. In the special case where x is a descendant of y then the above formula is simplified as: $dist(x, y) = dist(y, y_x)$. As an example assume two values from the hierarchy *Location*, x = ‘USA’ and y = ‘Europe’, where the descendant of y is selected as $f(desc_{L_x}^{L_y}(y)) = \text{'UK'}$. Assume the distance between y and its descendant y_x is computed through the formula $dist(y_x, y) = \frac{|\,desc_{L_x}^{L_x}(y_x)\,|}{|\,desc_{L_x}^{L_y}(y)\,|}$ from the percentage family functions. The distance between x and the descendant of y is computed through the first formula from the path family functions with w_x and w_y set to 1. Consequently, the distance between x and y becomes $dist(\text{'USA'},\text{'Europe'}) =$

$$\frac{dist(y, y_x) + dist(y_x, x)}{2} = \frac{dist(\text{'Europe'},\text{'UK'}) + dist(\text{'UK'},\text{'USA'})}{2} = \frac{1/1 + 4/6}{2} = \frac{5}{6}$$

.

2. The distance function is expressed with respect to the detailed level. Assume $x_1 = f_{\mathrm{aggr}}(desc_{L_1}^{L_x}(x))$ is the value returned by applying an aggregation function over the set of descendants of x from the most detailed level L_1. Similarly, assume $y_1 = f_{aggr}(\, desc_{L_1}^{L_y}(\, y\,))$ is the result of an aggregation function applied over the set of descendants of y from the detailed level L_1. Then the normalized form of the distance between x and y can be formally expressed as $dist(x, y) = \frac{dist(x, x_1) + dist(x_1, y_1) + dist(y, y_1)}{3}$. The way of computing the individual distances of this formula depend upon the kind of the aggregation function that is applied over the set of descendants. There are two kinds of results that f_{aggr} may return. Firstly, in case f_{aggr} returns an arithmetic type value, such as when f_{aggr} is *count* or *sum* then distances between x_1 and y_1 may be computed by making use of the Minkowski distance. In such case the distance between a value x and x_1 may be computed by making use of a distance from the percentage family functions. Secondly, in case f_{aggr} returns a value from the set of descendants, such as when f_{aggr} is *min* or *max*, then the distance between x_1 and y_1 may be calculated from the distance function of the section *distance between two values from the same level of hierarchy*. As for the distance between a value and its descendant, this may be computed according to a function from the path or percentage families.

3.2.3 Highway Distance Functions. Assume that every level of hierarchy L is grouped into k groups and every group has its own representative r_k. Then, the distance between the values x and y can be expressed as $dist\ (x, y) = dist\ (x, r(x)) + dist\ (r(x), r(y)) + dist\ (y, r(y))$ where $r(x)$ and $r(y)$ denote the representatives of the groups that x and y belong into respectively. Similarly to the highway distance between two values in the same level, the individual distances of this formula depend upon the way the representative is selected. The possible ways that the individual distances may be computed are equivalent to the ones described in section 3.1 concerning highway distances between two values of the same level of hierarchy.

3.3 Distance functions between two cells of an OLAP cube.

In this section we describe the distance functions that can possibly be applied in order to measure the distance between two cells from a cube. Assume an OLAP cube C defined over the detailed schema $C= [L_1^0, L_2^0, \ldots, L_n^0, M_1^0, M_2^0, \ldots, M_m^0]$, where L_i^0 is a detailed level and M_i^0 is a detailed measure. In addition assume two cells from this cube, $c_1 = (l_1^1, l_2^1, \ldots, l_n^1, m_1^1, m_2^1, \ldots, m_m^1)$ and $c_2 = (l_1^2, l_2^2, \ldots, l_n^2, m_1^2, m_2^2, \ldots, m_m^2)$, where $l_i^1, l_i^2 \in dom(L_i^0)$ and m_i^1, m_i^2 denote the values of the corresponding measure M_i^0 . The distance between two cells c_1 and c_2 can be expressed in regards to a) their level coordinates and b) their measure values. Therefore, the distance between two cells c_1 and c_2 can be expressed as a synthesis of the partial distances $d_i(l_i^1, l_i^2)$ between levels and/or $d_i(m_i^1, m_i^2)$ between measures. The distance between two cells can be expressed according to the coordinates of each cell, thus the levels, and/or by taking into account the distances between the instance values of the cells. In other words, $dist(c_1, c_2) = f\,(d_i(L_i^1, L_i^2), d_i(M_i^1, M_i^2))$. The function f can possibly be (a) a weighted sum, (b) Minkowski distance, (c) min, (d) proportion of common coordinates.

3.3.1 Distance functions between two cells of a cube expressed as a weighted sum.

In this category the distance between two cells c_1, c_2 where $c_1, c_2 \in C$ can be expressed through the formula $f: \frac{\sum_{i=1}^{n} w_i d_i(l_i^1, l_i^2)}{\sum_{i=1}^{n} w_i} + \frac{\sum_{i=1}^{m} w_i' d_i(m_i^1, m_i^2)}{\sum_{i=1}^{m} w_i'}$, where w_i and w_i' are parameters that assign a weight for the level L_i and the measure M_i respectively, $d_i(l_i^1, l_i^2)$ denotes the partial distance between two values of the detailed level L_i^0 from dimension D_i and $d_i(m_i^1, m_i^2)$ denotes the partial distance between two instances of the measure M_i^0. Regarding the distance $d_i(l_i^1, l_i^2)$, this is expressed through the various formulas from the section 3.1 which describes the possible distance functions between two values from the same level of hierarchy over a dimension. The distance $d_i(m_i^1, m_i^2)$ between two instances of a measure can be calculated through the Minkowski family distance when m_i^1, m_i^2 are of arithmetic type, or through the simple identity function in case m_i^1, m_i^2 are of character type. The above formula is a general expression of the distance between two cells. Simplifications of this can be applied. For instance, the distance of two cells can be calculated only in regards to the coordinates that define each cell and without taking into consideration the measure values of each cell, i.e., by omitting from the above formula the second fraction. Moreover, in case the partial distances are normalized in the interval [0, 1] then, f expresses the overall distance between two cells normalized in the same interval [0, 1].

3.3.2 Distance functions between two cells of a cube expressed in regards to the Minkowski family distances.

In this section we describe the possible distance functions between two cells from a cube by making use of the Minkowski family distances. In general the Minkowski distance is defined from the formula $L_p[(x_1,\ldots,x_n),(y_1,\ldots,y_n)] = \sqrt[p]{\sum_{i=1}^{n} d_i(x_i, y_i)^p}$ where $d_i(x_i, y_i)$ denotes the distance between the two coordinates x_i and y_i of two given points x and y. Assume two cells $c_1 = (l_1^1, l_2^1, \ldots, l_n^1, m_1^1, m_2^1, \ldots, m_m^1)$ and $c_2 = (l_1^2, l_2^2, \ldots, l_n^2, m_1^2, m_2^2, \ldots, m_m^2)$, where $l_i^1, l_i^2 \in dom(L_i^0)$ and m_i^1, m_i^2 denote the values of the corresponding measure M_i^0. The Minkowski distance can be applied in this category, by substituting point coordinates x_i and y_i with cell coordinates, thus l_i^1 and l_i^2. In general, in the Minkowski family distances the partial distances are defined as $d_i(x_i, y_i)=|x_i - y_i|$. When applying the Minkowski distance over cell coordinates, then the partial distances $d_i(l_i^1, l_i^2)$ can be expressed as the distance between two values from the same level of hierarchy as described in section 3.1.

So far, the distance between two cells is described only in regards to their level coordinates. However, the distance between two cells can also be expressed by taking into consideration the instance values of the cells, thus their measure values. The Minkowski family distances can be applied, as well, in regards to the partial distances

$d_i(m_i^1, m_i^2)$. Therefore, the distance between two cells can be expressed by adding the equivalent two formulas. Depending on the value of *p* the Minkowski distances over two cells are defined as:

- $L_1 = \sum_{i=1}^{n} d_i(l_i^1, l_i^2) + \sum_{i=1}^{m} d_i(m_i^1, m_i^2)$, 1-norm distance
- $L_2 = \sqrt{\sum_{i=1}^{n} (d_i(l_i^1, l_i^2))^2} + \sqrt{\sum_{i=1}^{m} (d_i(m_i^1, m_i^2))^2}$, 2-norm distance
- $L_p = \sqrt[p]{\sum_{i=1}^{n} (d_i(l_i^1, l_i^2))^p} + \sqrt[p]{\sum_{i=1}^{m} (d_i(m_i^1, m_i^2))^p}$, p-norm distance
- $L_\infty = \lim_{p \to \infty} \left(\sqrt[p]{\sum_{i=1}^{n} (d_i(l_i^1, l_i^2))^p} \right) + \lim_{p \to \infty} \left(\sqrt[p]{\sum_{i=1}^{m} (d_i(m_i^1, m_i^2))^p} \right) =$
  $max(d_1(l_1^1, l_1^2), d_2(l_2^1, l_2^2), \ldots, d_n(l_n^1, l_n^2))$
  $+ max(d_1(m_1^1, m_1^2), d_2(m_2^1, m_2^2), \ldots, d_m(m_m^1, m_m^2))$, infinity norm distance or Chebyshev distance.

3.3.3 Distance functions between two cells of a cube expressed as the minimum partial distance. In this category the distance between two cells $c_1 = (l_1^1, l_2^1, \ldots, l_n^1, m_1^1, m_2^1, \ldots, m_m^1)$ and $c_2 = (l_1^2, l_2^2, \ldots, l_n^2, m_1^2, m_2^2, \ldots, m_m^2)$ can be expressed as: $\min_{d_i}\{d_i(l_i^1, l_i^2)\} + \min_{d_i}\{d_i(m_i^1, m_i^2)\} = \min\{d_1(l_1^1, l_1^2), d_2(l_2^1, l_2^2), \ldots, d_n(l_n^1, l_n^2)\}$ $+ \min\{d_1(m_1^1, m_1^2), d_2(m_2^1, m_2^2), \ldots, d_m(m_m^1, m_m^2)\}$. Therefore, the distance between two points is expressed as the minimum distance of their level coordinates plus the minimum distance of their measure values.

3.3.4 Distance functions between two cells of a cube expressed as a proportion of common coordinates. In this category the distance between two cells can be expressed as a proportion of their common values of their level coordinates and their measure values. Therefore, the distance between two cells $c_1 = (l_1^1, l_2^1, \ldots, l_n^1, m_1^1, m_2^1, \ldots, m_m^1)$ and $c_2 = (l_1^2, l_2^2, \ldots, l_n^2, m_1^2, m_2^2, \ldots, m_m^2)$ can be expressed through the formula *f*: $\frac{\#(l_i^1 = l_i^2 \forall i \in \{1,2,\ldots,n\})}{n} + \frac{\#(m_i^1 = m_i^2 \forall i \in \{1,2,\ldots,m\})}{m}$. The above formula states the distance between two cells as a summation of two fractions. The first fraction is the number of level values that are same for both cells, divided by the number of all level values that describe a cell. The second fraction expresses the number of measures that have the same value for both cells divided by the number of all possible measures in a cell.

3.4 Distance functions between two OLAP cubes

Assume two OLAP cubes *C* and $C^{'}$ defined through the same detailed schema [L_1^0, L_2^0, …, L_n^0, M_1^0, M_2^0, …,M_m^0], where L_i^0 is a detailed level and M_i^0 is a detailed measure. In addition assume that cube *C* consists of *l* cells of the form $c = (l_1, l_2, \ldots, l_n, m_1, m_2, \ldots, m_m)$ and cube *C*’ consists of *k* cells of the form $c' = (l_1^{'}, l_2^{'}, \ldots, l_n^{'}, m_1^{'}, m_2^{'}, \ldots, m_m^{'})$, where l_i, $l_i^{'} \in dom(L_i^0)$ and m_i, $m_i^{'}$ denote the values of the corresponding measure M_i^0 . In general the two cubes can be of different cardinality, i.e., $l \neq k$. Assume *dist*(*c*, *c’*) where $c \in C$ and $c' \in C'$ denotes the distance between two specific cells according to the various categories of section 3.3. The distance between the two cubes can be expressed as a synthesis of the partial distances *dist*(*c*, *c’*). In other words *dist*(*C*, *C*’)= *f* (*dist*(*c*, *c’*)) is a function of the partial distances *dist*(*c*, *c’*). The function *f* can possibly belong to one of the following families: (a) a weighted sum, (b) Minkowski distance, (c) closest relative, (d) Hausdorff distance, and, (e) Jaccard’s coefficient.

3.4.1 Distance functions between two cubes expressed as a weighted sum. In this category the distance between two cubes can possibly be expressed as a weighted sum over the distances between each cell from one cube to every cell from the other cube. Therefore, the distance can be expressed through the formula:

$$f : \frac{\sum_{\mathrm{i=1}}^{\mathrm{l}} \sum_{\mathrm{j=1}}^{\mathrm{k}} w_{ij}\, dist(c,c')}{\sum_{\mathrm{i=1}}^{\mathrm{l}} \sum_{\mathrm{j=1}}^{\mathrm{k}} w_{\mathrm{ij}}}$$

, where *dist*(*c*, *c’*)is the distance between a cell from cube *C* to a cell from cube *C*’ and w_{ij} denotes the weight factors assigned to each distance.

3.4.2 Distance functions between two cubes expressed through Minkowski family distances. The distance between two cubes *C* and *C*’ can be expressed by making use of a distance function from the Minkowski family. The distance between *C* and *C*’ by applying the Minkowski family distances, depending on the values of the parameter *p*, are defined as:

- $L_1 = \sum_{\mathrm{i=1}}^{\mathrm{l}} \sum_{\mathrm{j=1}}^{\mathrm{k}} dist(c,c')$, 1-norm distance
- $L_2 = \sqrt{\sum_{\mathrm{i=1}}^{\mathrm{l}} \sum_{\mathrm{j=1}}^{\mathrm{k}} dist(c,c')^2}$, 2-norm distance
- $L_{\mathrm{p}} = \sqrt[\mathrm{p}]{\sum_{\mathrm{i=1}}^{\mathrm{l}} \sum_{\mathrm{j=1}}^{\mathrm{k}} dist(c,c')^{\mathrm{p}}}$, p-norm distance

- $L_\infty = \lim_{p \to \infty} \left(\sqrt[p]{\sum_{i=1}^{l} \sum_{j=1}^{k} dist(c,c')^p} \right) =$ $\max\{dist(c_1,c'_1), dist(c_1,c'_2), ..., dist(c_1,c'_k), ..., dist(c_l,c'_1), dist(c_l,c'_2), ..., dist(c_l,c'_k)\}$, infinity norm distance or Chebyshev distance.

3.4.3 Distance functions between two cubes expressed in regards to the closest relative. In this category the distance between two cubes *C* and *C*' is expressed as the summation of distances between every cell of a cube with the most similar cell of the other cube through the formula: $dist(C,C') = \frac{\sum_{i=1}^{k} \min_{j}\{dist(c,c')\}}{k}$. Another option is to express the distance as the infimum of the distances between any two of the cubes' respective cells. Therefore the distance between *C* and *C*' is expressed as: $dist(C,C') = \inf\{dist(c,c') \mid c \in C, c' \in C'\}$, where *dist*(*c*, *c'*) is the distance between a cell from cube *C* to a cell from cube *C*'. In case the two cubes are disjoint i.e., $C \cap C' = \emptyset$, then *dist*(*C*, *C'*) is a positive number, whereas if the two cubes have common cells i.e., $C \cap C' \neq \emptyset$, then *dist*(*C*, *C'*) is zero.

3.4.4 Distance functions between two cubes expressed by Hausdorff distance. In this category the distance between two cubes can be expressed by making use of the *Hausdorff* distance [HuKR93]. The Hausdorff distance between two cubes can be defined as *H*(*C*, *C*') = max (*h*(*C*, *C*'), *h*(*C*', *C*)) where h(C, C') = $\max_{c \in C}\{\min_{c' \in C'}\{dist(c,c')\}\}$ and *dist* (*c*, *c'*) is the distance between two cells *c* and *c*' from the cubes *C* and *C*' respectively. The function *h*(*C*, *C*') is called the *directed* Hausdorff distance from *C* to *C*' and the distance measured is the maximum distance of a cube *C* to the *"nearest"* cell of the other cube *C*'. The Hausdorff distance is the maximum of *h*(*C*, *C*') and *h*(*C*', *C*), thus it measures the distance of a cell $c \in C$ that is the *"farthest"* from any cell *c*' of the cube *C*' and vice versa. In other words, the Hausdorff distance expresses the degree of mismatch between *C* and *C*'.

3.4.5 Distance functions between two cubes expressed by Jaccard's Coefficient. In this category the distance between two cubes can be expressed in regards to the *Jaccard's coefficient* [ZADB06]. The Jaccard's coefficient is defined as: $dist(C,C') = 1 - \frac{|C \cap C'|}{|C \cup C'|}$. The distance is based on the ratio between the cardinalities of intersection and union of the cubes *C* and *C*'. In addition, based on the Jaccard's coefficient the distance between two cubes can be expressed by applying the Dice's coefficient. For two cubes *C* and *C*' the Dice's coefficient is defined as: $dist(C,C') = \frac{2|C \cap C'|}{|C| + |C'|}$. This formula expresses the similarity between two cubes as the ratio between the cardinality of intersection and the summation of cardinalities of the two cubes.

4 Selecting an Appropriate Distance Function for Multidimensional Data

The choice of the distance function that can be applied depends upon the user needs as well as the type of values that each hierarchy contains. As for the type of values, these may be one of the following: nominal, ordinal and interval, which were described earlier. We describe the appropriateness of the distance functions in regards to (a) the type of values and (b) user preferences.

4.1 Selecting Distance Functions According to the Type of Values

In this section, we summarize (shown in table 1 and table 2) the possible distance functions as well as their appropriateness usage depending on the type of values that a level of hierarchy contains. Specifically, table 1 shows the distance functions that can be applied when computing the distance between two values from the same level of hierarchy and table 2 shows the distance functions for two values from different levels of hierarchy. In both tables, each family of distance functions is labeled with a Y (Yes) or N (No) showing the suitability of the family function in regards to the type of values. In case some family functions are expressed in regards to other family functions, then the name of the later family function is tagged in the former family function.

Table 1. Summary of distance functions between two values in the same level of hierarchy.

3.1 **Same level**					Nom	Ord	Int
3.1.1	Locally	*Explicit*			**Y**	**Y**	**Y**
		Attribute based			**Y**	**Y**	**Y**
		Function of values	*Minkowski*		**Y**	**Y**	**Y**
			identity		**N**	**N**	**Y**
3.1.2	Hierarchical	*Wrt lower level*					
			g(x1, y1)	**d(x, x1)**			
		Sum	Function of values	Attribute based			
		Max	Locally	Different level			
				Wrt hierarchy path			
				Percentage family function			
		Wrt hierarchy path			**Y**	**Y**	**Y**
		Highway					
			dist(x, r(x))	**dist(r(x), r(y))**			
		explicit			**Y**	**Y**	**Y**
		ancestor	Different level	Locally			
			Wrt hierarchy path	Different level			
			Percentage family function				
		descendant					
		Sum	Attribute based	Function of values			
		Max	Different level	Locally			
			Wrt hierarchy path				
			Percentage family function				

Table 2. Summary of distance functions between two values in different levels of hierarchy.

3.2 **Different level**					Nom	Ord	Int
3.2.1	Wrt ancestor	*Ancestor of x*					
			Dist(x, x_y)	**Dist(y, x_y)**			
			Wrt hierarchy path	Same level			
			Percentage family function				
		Common ancestor z					
			Dist(x, z)				
			Percentage family function				
		Percentage family function			**Y**	**Y**	**Y**
3.2.2	Wrt descendant	*Descendant of y*					
			Dist(y, y_x)	**Dist(x, y_x)**			
			Wrt hierarchy path	Same level			
			Percentage family function				
		Wrt detail level					
			Dist(x, x1)	**Dist(x1, y1)**			
		Sum	Percentage family function	Minkowski			
		Max	Wrt hierarchy path	Same level			
			Percentage family function				
3.2.3	HighWay						

In general, for interval type values all possible distance functions may be applied, whereas for nominal and ordinal type values the pure mathematical distance functions such as the Minkowksi distance cannot be applied. For nominal type values it is straightforward that their instances cannot provide an order, whereas for ordinal and interval type values there is an intuitive order among them. However, in a lattice if a level of hierarchy is of type nominal and an upper level is of type ordinal or interval, then the lower level nominal type values may provide an order if these are expressed in regards to the ancestors of the upper level.

4.2 Selecting Distance Functions According to the User Preferences

In this section we describe a user study that we conducted for the purpose of discovering which distance functions seem to be more suitable for user needs. The experiment involved 15 out of which 10 are graduate students in Computer Science and 5 that are of other backgrounds. In the rest of the paper we refer to the set of users with computer science background as *Users_cs*, the set of users with other background as *Users_non* and when thinking of all users independently of their background we denote the set as *Users_all*.

For the needs of this experiment, we made use of the "Adult" real data set taking into consideration the dimension hierarchies as described in [FuWY05]. This dataset contains the fact table *Adult* and 8 dimension tables. The type of values as well as the number of tuples and the number of the dimension levels for each table are shown in Table 3.

Table 3. Adult dataset tables

Table	Value Type	# Tuples	# Dim. Levels
Adult fact		30418	-
Age Dim.	Numeric	72	5
Education Dim.	Categorical	16	5
Gender Dim.	Categorical	2	2
Marital Status Dim.	Categorical	7	4
Native Country Dim.	Categorical	41	4
Occupation Dim.	Categorical	14	3
Race Dim.	Categorical	5	3
Work Class Dim.	Categorical	7	4

Each user was given 14 different case scenarios, from which the 2 last were a reordering of 2 previous ones. This was done in order to discover whether the users were stable in their selection. The purpose of the experiment is to assess which distance function between two values is best in regards to the user preferences. Therefore, each case scenario contained a reference cube and a set of cubes, which we call *variant* cubes, that occurred by slightly altering the reference cube. For each case scenario the generation of these variant cubes was performed as follows. First a random cube was selected as a reference cube to be compared with the others. Within the 14 scenarios we included different kinds of cubes in regards to the value types as well as the different levels of granularity. Secondly, for each reference cube, the variant cubes were generated (a) by altering the granularity level for one dimension, or, (b) by altering the value range of the reference cube. For instance, assume that a reference cube contains the dimension levels Age_level_1, $Education\_level_2$ under the age interval [*17, 21*]. According to the first type of modification, a variant cube could be generated by changing the dimension level to Age_level_2 or Age_level_0, or similarly changing the level of the Education Dimension. According to the second type of modification, another variant cube could be generated by changing the age interval to [*22, 26*] or to [*17, 26*]. Among all possible variations of the reference cube we chose the set of variant cubes such that each of them was the closest to the reference cube given a specific distance function. In order to observe which distance function is preferred by users depending on the type of data the cubes contained, we have distinguished the 14 scenarios into three sets. The first set consists of cubes that contain only arithmetic type values (these were 5 cubes). The second set consists of cubes containing only categorical type values (these were 2 cubes). The third set consists of cubes that contained a combination of both categorical and arithmetic type values (these were 7 cubes). All the scenarios used in the experiment can be seen in the Appendix.

Table 4. Notation of distance functions used in the experiment

Family	**Abbr.**	**Distance function name**
Same Level	δ_{M}	Manhattan
	$\delta_{\mathrm{Low,c}}$	With respect to a lower level of hierarchy where f_{aggr} =count
	$\delta_{\mathrm{Low,m}}$	With respect to a lower level of hierarchy where f_{aggr} = max
Hierarchical Path	$\delta_{\mathrm{LCA,P}}$	Lowest common ancestor through f_{path}
	$\delta_{\mathrm{LCA,D}}$	Lowest common ancestor through f_{depth}
Different Levels	$\delta_{\%}$	Applying percentage function
	δ_{Anc}	With respect to an ancestor x_{y}
	δ_{Desc}	With respect to a descendant y_{x}
Highway	$\delta_{\mathrm{H,Desc}}$	Highway, selecting the representative from a descendant
	$\delta_{\mathrm{H,Anc}}$	Highway, selecting the representative from an ancestor

In each case scenario, the users were asked to select which of the variant cubes seemed more similar to the reference cube based only on their personal criteria. The various distance functions used are shown in Table 4. The first column of Table 4 shows the family in which each distance function belongs in according to the previous section (Section 3). The second column assigns an abbreviated name for each function. The distance functions that were tested are all from the category of distance functions between two values. To compute the distance between two cubes, the first formula from the family of *Closest Relative* distances was used (see section 3.4.3). The distance function between two cells of cubes was set to be the weighted sum of the partial distances of the two values, one from each cell, with all weights set to 1 (see the generalized form in section 3.3.1).

The analysis of the collected data provides several findings. The first finding concerns the *top three most preferred distance functions* measured over the detailed data for all scenarios and all users. It is remarkable that the top three distance functions for each of the user groups were the same and with the same ordering. Specifically, the top three distance functions in descending order are the $\delta_{\mathrm{LCA,P}}$, the δ_{Anc} and the $\delta_{\mathrm{H,Desc}}$. The specific frequencies for each one of the top three distance function in each group of users is shown in Table 5.

Table 5. Top three most frequent distance functions for each user group.

	Users_all	***Users_cs***	***Users_non***
$\boldsymbol{\delta}_{\mathbf{LCA,P}}$	40.47%	38.57%	44.28%
$\boldsymbol{\delta}_{\mathbf{Anc}}$	18.09%	20%	14.28%
$\boldsymbol{\delta}_{\mathbf{H,Desc}}$	9.52%	10.71%	7.14%

The second finding concerns *the most preferred function by users depending on the type of data the cubes contained*. Table 6 summarizes the result of the most frequent distance function for each set of scenarios and each set of users. We can observe that for the *categorical* type of cubes, all types of users mainly prefer the $\delta_{\mathrm{LCA,P}}$ distance function, whereas for the two other sets (i.e., the *arithmetic* and the *arithmetic & categorical*) the functions that users mainly prefer are the $\delta_{\mathrm{LCA,P}}$ and δ_{Anc} function. The fact that more than one distance functions appear as winners in the cells of Table 6 is due to ties when calculating the frequency of occurrence for each function.

Table 6. Most frequent distance function for each set of scenarios.

	Users_all	*Users_cs*	*Users_non*
Arithmetic	δ_{Anc}	$\delta_{\mathrm{LCA,D}}$, $\delta_{\mathrm{H,Desc}}$, δ_{Anc}	$\delta_{\mathrm{LCA,D}}$
Categorical	$\delta_{\mathrm{LCA,D}}$	$\delta_{\mathrm{LCA,D}}$	$\delta_{\mathrm{LCA,D}}$
Arithmetic & Categorical	δ_{Anc}	δ_{Anc}	$\delta_{\mathrm{LCA,D}}$, δ_{Anc}

The third finding concerns the *winner distance function per scenario*. For every scenario, we take into account the 15 occurrences by all users and see which distance function is the most frequent. We call this function the winner function of the scenario. The winner function of the scenario can be seen in Table 7. The most frequent winner function was $\delta_{\mathrm{LCA,P}}$ for every user group. Specifically, the percentages were 35.71% for the *Users_all* group, 35,71% for the *Users_cs* group and 57.14% for the *Users_non* group. Similarly, the winner function for each user is the $\delta_{\mathrm{LCA,P}}$ function which occurred for 14 out of the 15 users. There was only one user from the *Users_cs* group whose most frequent function was the function $\delta_{\mathrm{LCA,D}}$.

The fourth finding of the user study concerns of the *diversity and spread* of user choices. There are two major findings: (a) All functions were at some point picked by some user and (b) there are certain functions that appeared as user choices for all the users of a user group. Specifically, functions $\delta_{\mathrm{LCA,P}}$, $\delta_{\mathrm{H,Desc}}$ and δ_{Anc} were selected at least once by users of group *Users_cs*. Similarly, functions $\delta_{\mathrm{LCA,P}}$, $\delta_{\mathrm{Low,m}}$ and δ_{Anc} were selected at least once by *Users_non*.

The fifth finding concerned the *most preferred family of functions*. Table 8 depicts the absolute number of appearances of each distance function family per user group. The most preferred family of distances is the *Hierarchy Path* family, which also contains the top one most preferred distance function $\delta_{\mathrm{LCA,P}}$. Moreover, we observe that the ranking of the distance function families was exactly the same for each user group.

Table 7. Most frequent distance function for each scenario per user group.

	Users_all	**Users_cs**	**Users_non**
Cube1	$\delta_{\text{H,Desc}}$	$\delta_{\text{H,Desc}}$	$\delta_{\%}$
Cube2	δ_{Anc}	δ_{Anc}	$\delta_{\text{LCA,P}}$
Cube3	δ_{Anc}	δ_{Anc}	$\delta_{\text{LCA,P}}$
Cube4	$\delta_{\text{LCA,D}}$	$\delta_{\text{LCA,D}}$	$\delta_{\text{H,Desc}}$
Cube5	δ_{Anc}	δ_{Anc}	δ_{Anc}
Cube6	$\delta_{\text{LCA,P}}$	$\delta_{\text{LCA,P}}$	$\delta_{\text{LCA,P}}$
Cube7	$\delta_{\text{LCA,P}}$	$\delta_{\text{LCA,P}}$	δ_{Anc}
Cube8	$\delta_{\text{H,Desc}}$	$\delta_{\text{H,Desc}}$	$\delta_{\text{LCA,P}}$
Cube9	$\delta_{\text{LCA,P}}$	$\delta_{\text{LCA,P}}$	$\delta_{\text{LCA,P}}$
Cube10	$\delta_{\text{LCA,P}}$	$\delta_{\text{LCA,P}}$	$\delta_{\text{LCA,P}}$
Cube11	$\delta_{\%}$	$\delta_{\%}$	$\delta_{\text{LCA,D}}$
Cube12	$\delta_{\text{Low,m}}$	$\delta_{\text{Low,m}}$	$\delta_{\text{Low,m}}$
Cube13	δ_{Anc}	δ_{Anc}	$\delta_{\text{LCA,P}}$
Cube14	$\delta_{\text{LCA,P}}$	$\delta_{\text{LCA,P}}$	$\delta_{\text{LCA,P}}$

Table 8. Frequencies of preferred distances within each user group for each distance family.

Family	**Same level**	**Hierarchy Path**	**Different levels of hierarchy**	**Highway**
Users_cs	10	69	41	20
Users_non	7	34	20	9
Users_all	17	103	61	29

The *stability* of users on their selections was the sixth observation and was determined by the following results, where the 13th and the 14th scenario were a reordering of the 3rd and 10th scenario respectively. 4 out of 5 users from the set of *Users_non*, 6 out of 10 users from the set of *Users_cs* (consequently, 10 out of the 15 users from the set of *Users_all*) selected exactly the same distance function for both of the two similar scenarios. 1 out of 5 users from the set of *Users_non*, 4 out of 10 users from the set of *Users_cs* (consequently, 5 out of 15 users from the set of *Users_all*) selected exactly the same distance function for only one out of the two similar scenarios. There were no users that gave a different distance function for both the two similar scenarios.

Summary. Overall, our findings indicate that the most preferred distance function is $\delta_{\text{LCA,P}}$, which is expressed in regards to the shortest path of a hierarchy dimension.

Apart from $\delta_{LCA,P}$, the distance functions δ_{Anc} and $\delta_{H,Desc}$ were widely chosen by users. In addition, the most preferred distance function family is the *Hierarchy Path* family.

5. Conclusions

This paper presented a wide variety of distance functions that can be used in order to compute the similarity between two OLAP cubes. The functions were described with respect to the properties of the dimension hierarchies and based on these they were grouped into functions that can be applied (a) between two values from the same level of hierarchy, (b) between two values in different levels, (c) between two cells and (d) between two OLAP cubes. In order to assess which distance function between two values is best in regards to the user needs and data type, we conducted a user study analysis. Our findings clearly indicated that the distance function $\delta_{LCA,P}$, which is expressed in regards to the shortest path of a hierarchy dimension was the most preferred by users in various cases of our experiment. Moreover, two more functions were widely chosen by users. These were the δ_{Anc} function that is expressed in regards to an ancestor value and the $\delta_{H,Desc}$ function that is a highway function, by selecting the representative from a descendant. Future work can be pursued in various directions including (a) the deeper examination of the presented families of functions with more complicated scenarios and (b) the discovery of the foundational reasons for the observed user preferences.

References

[FuWY05] B. C. M. Fung, K. Wang, and P. S. Yu. "Top-Down Specialization for Information and Privacy Preservation", In Proceedings of the 21st IEEE International Conference on Data Engineering (ICDE 2005), Tokyo, Japan, April 5-8, 2005. See also http://ddm.cs.sfu.ca.

[GMNS09] A. Giacometti, P. Marcel, E. Negre, A. Soulet. “Query Recommendations for OLAP Discovery Driven Analysis”. In Proc. ACM 12th International Workshop on Data Warehousing and OLAP (DOLAP 2009), (in conjunction with CIKM 2009), Hong Kong, November 6, 2009.

[HuKR93] D. P. Huttenlocher, G. A. Klanderman, W. J. Rucklidge, “Comparing images using the hausdorff distance”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(9), pp. 850-863, September 1993.

[LiBM03] Y. Li, Z. A. Bandar, D. McLean, “An approach for measuring semantic similarity between words using multiple information sources”, IEEE Transactions on Knowledge and Data Engineering, 15(4), pp. 871-882, July/August 2003.

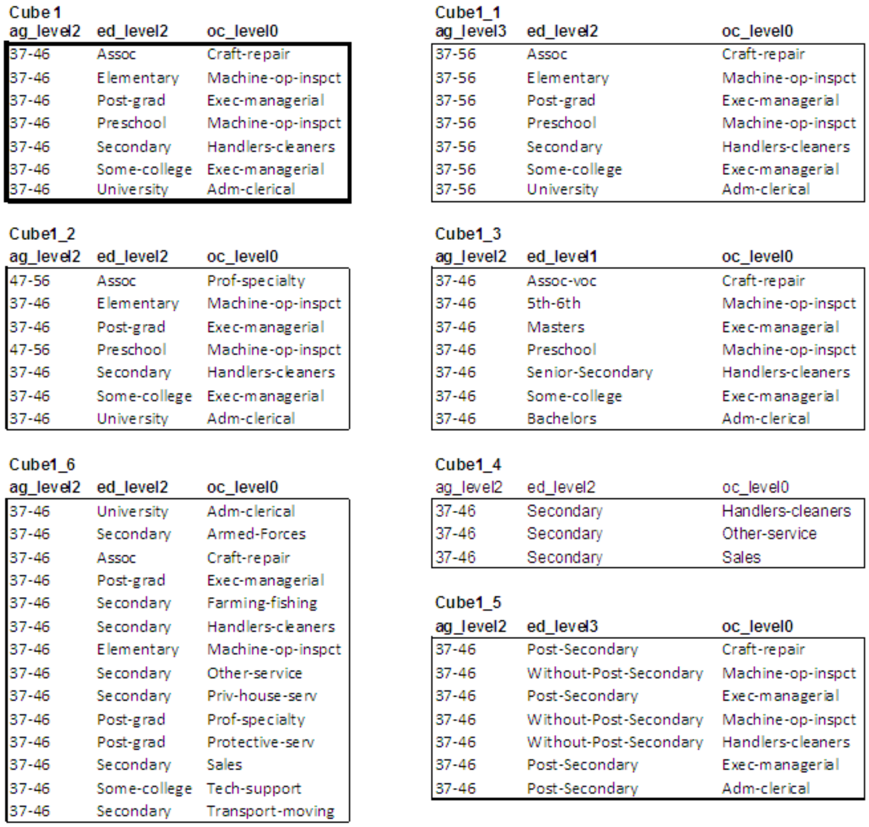
[Sara99] S. Sarawagi, “Explaining differences in multidimensional aggregates” In Proc. 25th Very Large Database Conference (VLDB), pp. 42-53, Edinburgh, Scotland, 1999.

[Sara00] S. Sarawagi, “User-adaptive exploration of multidimensional data”, In Proc. 26th Very Large Database Conference (VLDB), pp. 307-316, Cairo, Egypt, 2000.

[VaSk00] P. Vassiliadis, S. Skiadopoulos, “Modelling and Optimisation Issues for Multidimensional Databases”, In Proc. 12th Conference on Advanced Information Systems Engineering (CAiSE '00), pp. 482-497, Stockholm, Sweden, 5-9 June 2000.

[ZADB06] P. Zezula, G. Amato, V. Dohnal, M. Batko, Similarity Search: the metric space approach. Springer Science + Business Media, Inc., pp. 13-14, 2006.

Appendix



Cube 1

ag_level2	ed_level2	oc_level0
37-46	Assoc	Craft-repair
37-46	Elementary	Machine-op-inspct
37-46	Post-grad	Exec-managerial
37-46	Preschool	Machine-op-inspct
37-46	Secondary	Handlers-cleaners
37-46	Some-college	Exec-managerial
37-46	University	Adm-clerical

Cube1_1

ag_level3	ed_level2	oc_level0
37-56	Assoc	Craft-repair
37-56	Elementary	Machine-op-inspct
37-56	Post-grad	Exec-managerial
37-56	Preschool	Machine-op-inspct
37-56	Secondary	Handlers-cleaners
37-56	Some-college	Exec-managerial
37-56	University	Adm-clerical

Cube1_2

ag_level2	ed_level2	oc_level0
47-56	Assoc	Prof-specialty
37-46	Elementary	Machine-op-inspct
37-46	Post-grad	Exec-managerial
47-56	Preschool	Machine-op-inspct
37-46	Secondary	Handlers-cleaners
37-46	Some-college	Exec-managerial
37-46	University	Adm-clerical

Cube1_3

ag_level2	ed_level1	oc_level0
37-46	Assoc-voc	Craft-repair
37-46	5th-6th	Machine-op-inspct
37-46	Masters	Exec-managerial
37-46	Preschool	Machine-op-inspct
37-46	Senior-Secondary	Handlers-cleaners
37-46	Some-college	Exec-managerial
37-46	Bachelors	Adm-clerical

Cube1_6

ag_level2	ed_level2	oc_level0
37-46	University	Adm-clerical
37-46	Secondary	Armed-Forces
37-46	Assoc	Craft-repair
37-46	Post-grad	Exec-managerial
37-46	Secondary	Farming-fishing
37-46	Secondary	Handlers-cleaners
37-46	Elementary	Machine-op-inspct
37-46	Secondary	Other-service
37-46	Secondary	Priv-house-serv
37-46	Post-grad	Prof-specialty
37-46	Post-grad	Protective-serv
37-46	Secondary	Sales
37-46	Some-college	Tech-support
37-46	Secondary	Transport-moving

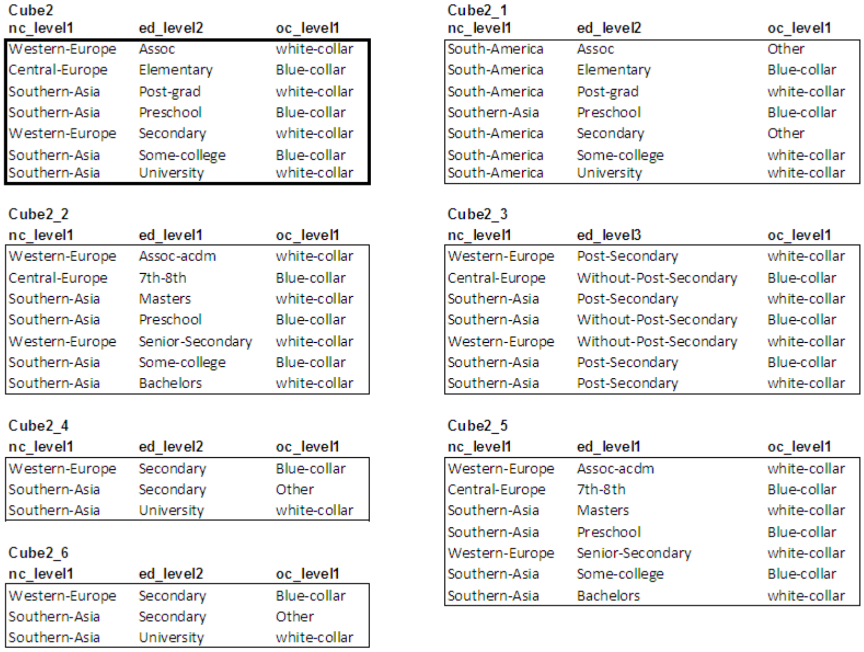
Cube1_4

ag_level2	ed_level2	oc_level0
37-46	Secondary	Handlers-cleaners
37-46	Secondary	Other-service
37-46	Secondary	Sales

Cube1_5

ag_level2	ed_level3	oc_level0
37-46	Post-Secondary	Craft-repair
37-46	Without-Post-Secondary	Machine-op-inspct
37-46	Post-Secondary	Exec-managerial
37-46	Without-Post-Secondary	Machine-op-inspct
37-46	Without-Post-Secondary	Handlers-cleaners
37-46	Post-Secondary	Exec-managerial
37-46	Post-Secondary	Adm-clerical

Fig. 3. Cube scenario 1.



Cube2

nc_level1	ed_level2	oc_level1
Western-Europe	Assoc	white-collar
Central-Europe	Elementary	Blue-collar
Southern-Asia	Post-grad	white-collar
Southern-Asia	Preschool	Blue-collar
Western-Europe	Secondary	white-collar
Southern-Asia	Some-college	Blue-collar
Southern-Asia	University	white-collar

Cube2_1

nc_level1	ed_level2	oc_level1
South-America	Assoc	Other
South-America	Elementary	Blue-collar
South-America	Post-grad	white-collar
Southern-Asia	Preschool	Blue-collar
South-America	Secondary	Other
South-America	Some-college	white-collar
South-America	University	white-collar

Cube2_2

nc_level1	ed_level1	oc_level1
Western-Europe	Assoc-acdm	white-collar
Central-Europe	7th-8th	Blue-collar
Southern-Asia	Masters	white-collar
Southern-Asia	Preschool	Blue-collar
Western-Europe	Senior-Secondary	white-collar
Southern-Asia	Some-college	Blue-collar
Southern-Asia	Bachelors	white-collar

Cube2_3

nc_level1	ed_level3	oc_level1
Western-Europe	Post-Secondary	white-collar
Central-Europe	Without-Post-Secondary	Blue-collar
Southern-Asia	Post-Secondary	white-collar
Southern-Asia	Without-Post-Secondary	Blue-collar
Western-Europe	Without-Post-Secondary	white-collar
Southern-Asia	Post-Secondary	Blue-collar
Southern-Asia	Post-Secondary	white-collar

Cube2_4

nc_level1	ed_level2	oc_level1
Western-Europe	Secondary	Blue-collar
Southern-Asia	Secondary	Other
Southern-Asia	University	white-collar

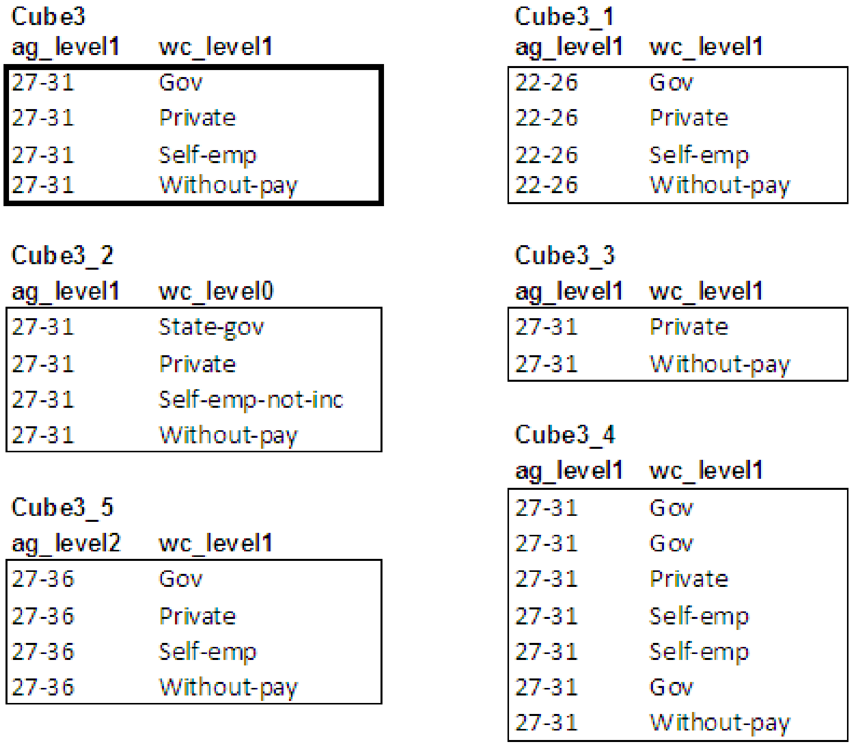
Cube2_5

nc_level1	ed_level1	oc_level1
Western-Europe	Assoc-acdm	white-collar
Central-Europe	7th-8th	Blue-collar
Southern-Asia	Masters	white-collar
Southern-Asia	Preschool	Blue-collar
Western-Europe	Senior-Secondary	white-collar
Southern-Asia	Some-college	Blue-collar
Southern-Asia	Bachelors	white-collar

Cube2_6

nc_level1	ed_level2	oc_level1
Western-Europe	Secondary	Blue-collar
Southern-Asia	Secondary	Other
Southern-Asia	University	white-collar

Fig. 4. Cube scenario 2.



Cube3

ag_level1	wc_level1
27-31	Gov
27-31	Private
27-31	Self-emp
27-31	Without-pay

Cube3_1

ag_level1	wc_level1
22-26	Gov
22-26	Private
22-26	Self-emp
22-26	Without-pay

Cube3_2

ag_level1	wc_level0
27-31	State-gov
27-31	Private
27-31	Self-emp-not-inc
27-31	Without-pay

Cube3_3

ag_level1	wc_level1
27-31	Private
27-31	Without-pay

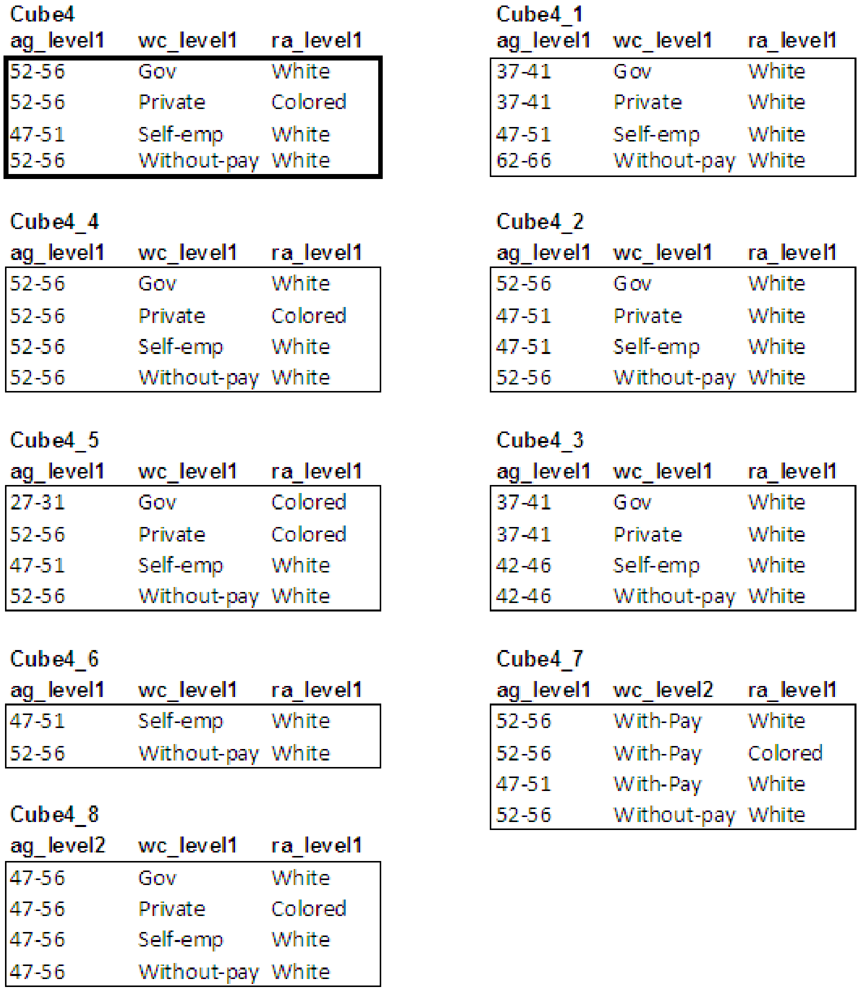
Cube3_4

ag_level1	wc_level1
27-31	Gov
27-31	Gov
27-31	Private
27-31	Self-emp
27-31	Self-emp
27-31	Gov
27-31	Without-pay

Cube3_5

ag_level2	wc_level1
27-36	Gov
27-36	Private
27-36	Self-emp
27-36	Without-pay

Fig. 5. Cube scenario 3.



Cube4

ag_level1	wc_level1	ra_level1
52-56	Gov	White
52-56	Private	Colored
47-51	Self-emp	White
52-56	Without-pay	White

Cube4_4

ag_level1	wc_level1	ra_level1
52-56	Gov	White
52-56	Private	Colored
52-56	Self-emp	White
52-56	Without-pay	White

Cube4_5

ag_level1	wc_level1	ra_level1
27-31	Gov	Colored
52-56	Private	Colored
47-51	Self-emp	White
52-56	Without-pay	White

Cube4_6

ag_level1	wc_level1	ra_level1
47-51	Self-emp	White
52-56	Without-pay	White

Cube4_8

ag_level2	wc_level1	ra_level1
47-56	Gov	White
47-56	Private	Colored
47-56	Self-emp	White
47-56	Without-pay	White

Cube4_1

ag_level1	wc_level1	ra_level1
37-41	Gov	White
37-41	Private	White
47-51	Self-emp	White
62-66	Without-pay	White

Cube4_2

ag_level1	wc_level1	ra_level1
52-56	Gov	White
47-51	Private	White
47-51	Self-emp	White
52-56	Without-pay	White

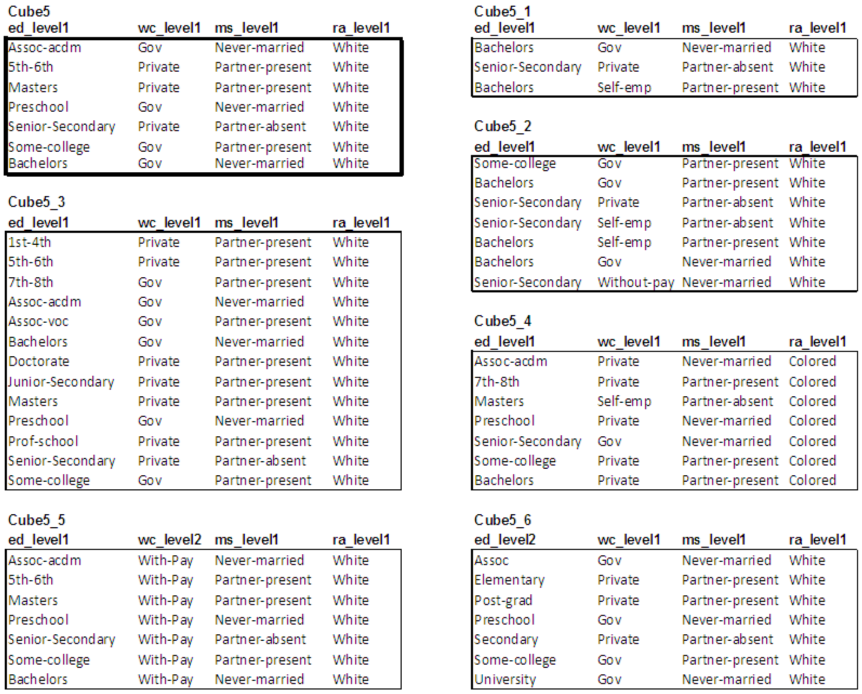
Cube4_3

ag_level1	wc_level1	ra_level1
37-41	Gov	White
37-41	Private	White
42-46	Self-emp	White
42-46	Without-pay	White

Cube4_7

ag_level1	wc_level2	ra_level1
52-56	With-Pay	White
52-56	With-Pay	Colored
47-51	With-Pay	White
52-56	Without-pay	White

Fig. 6. Cube scenario 4.



Cube5

ed_level1	wc_level1	ms_level1	ra_level1
Assoc-acdm	Gov	Never-married	White
5th-6th	Private	Partner-present	White
Masters	Private	Partner-present	White
Preschool	Gov	Never-married	White
Senior-Secondary	Private	Partner-absent	White
Some-college	Gov	Partner-present	White
Bachelors	Gov	Never-married	White

Cube5_1

ed_level1	wc_level1	ms_level1	ra_level1
Bachelors	Gov	Never-married	White
Senior-Secondary	Private	Partner-absent	White
Bachelors	Self-emp	Partner-present	White

Cube5_2

ed_level1	wc_level1	ms_level1	ra_level1
Some-college	Gov	Partner-present	White
Bachelors	Gov	Partner-present	White
Senior-Secondary	Private	Partner-absent	White
Senior-Secondary	Self-emp	Partner-absent	White
Bachelors	Self-emp	Partner-present	White
Bachelors	Gov	Never-married	White
Senior-Secondary	Without-pay	Never-married	White

Cube5_3

ed_level1	wc_level1	ms_level1	ra_level1
1st-4th	Private	Partner-present	White
5th-6th	Private	Partner-present	White
7th-8th	Gov	Partner-present	White
Assoc-acdm	Gov	Never-married	White
Assoc-voc	Gov	Partner-present	White
Bachelors	Gov	Never-married	White
Doctorate	Private	Partner-present	White
Junior-Secondary	Private	Partner-present	White
Masters	Private	Partner-present	White
Preschool	Gov	Never-married	White
Prof-school	Private	Partner-present	White
Senior-Secondary	Private	Partner-absent	White
Some-college	Gov	Partner-present	White

Cube5_4

ed_level1	wc_level1	ms_level1	ra_level1
Assoc-acdm	Private	Never-married	Colored
7th-8th	Private	Partner-present	Colored
Masters	Self-emp	Partner-absent	Colored
Preschool	Private	Never-married	Colored
Senior-Secondary	Gov	Never-married	Colored
Some-college	Private	Partner-present	Colored
Bachelors	Private	Partner-present	Colored

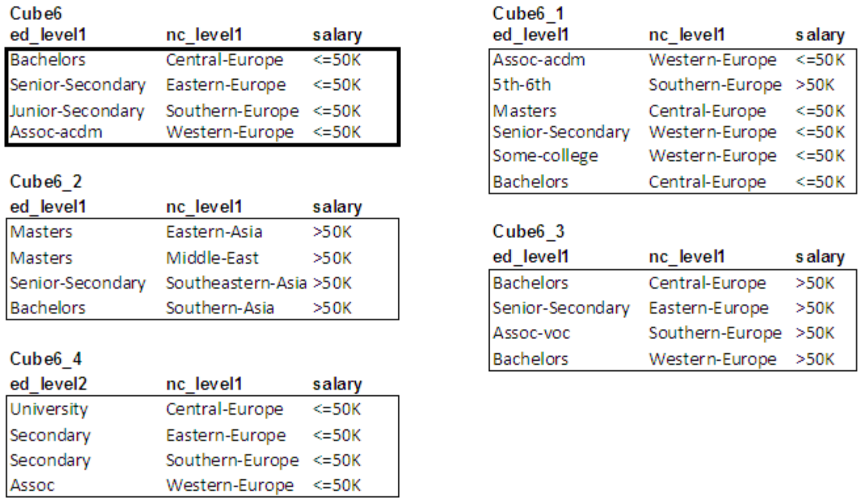
Cube5_5

ed_level1	wc_level2	ms_level1	ra_level1
Assoc-acdm	With-Pay	Never-married	White
5th-6th	With-Pay	Partner-present	White
Masters	With-Pay	Partner-present	White
Preschool	With-Pay	Never-married	White
Senior-Secondary	With-Pay	Partner-absent	White
Some-college	With-Pay	Partner-present	White
Bachelors	With-Pay	Never-married	White

Cube5_6

ed_level2	wc_level1	ms_level1	ra_level1
Assoc	Gov	Never-married	White
Elementary	Private	Partner-present	White
Post-grad	Private	Partner-present	White
Preschool	Gov	Never-married	White
Secondary	Private	Partner-absent	White
Some-college	Gov	Partner-present	White
University	Gov	Never-married	White

Fig. 7. Cube scenario 5.



Cube6

ed_level1	nc_level1	salary
Bachelors	Central-Europe	<=50K
Senior-Secondary	Eastern-Europe	<=50K
Junior-Secondary	Southern-Europe	<=50K
Assoc-acdm	Western-Europe	<=50K

Cube6_1

ed_level1	nc_level1	salary
Assoc-acdm	Western-Europe	<=50K
5th-6th	Southern-Europe	>50K
Masters	Central-Europe	<=50K
Senior-Secondary	Western-Europe	<=50K
Some-college	Western-Europe	<=50K
Bachelors	Central-Europe	<=50K

Cube6_2

ed_level1	nc_level1	salary
Masters	Eastern-Asia	>50K
Masters	Middle-East	>50K
Senior-Secondary	Southeastern-Asia	>50K
Bachelors	Southern-Asia	>50K

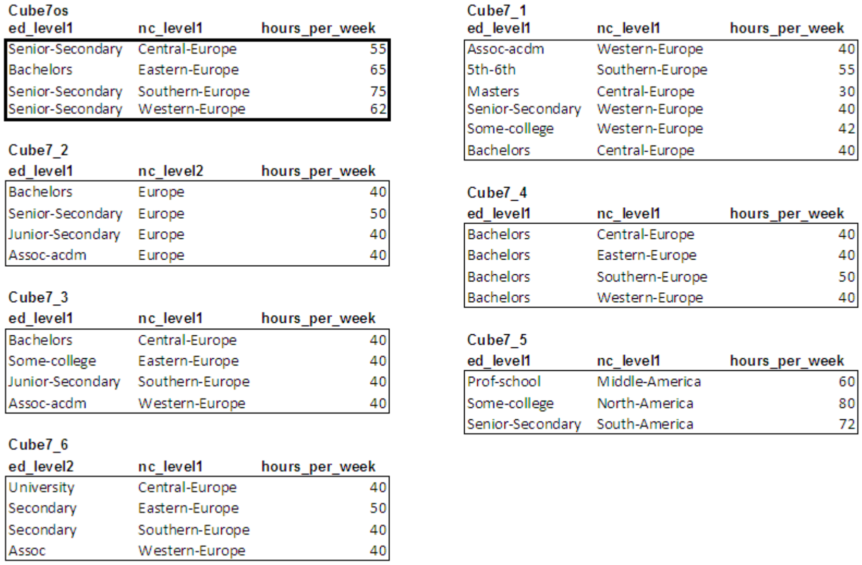
Cube6_3

ed_level1	nc_level1	salary
Bachelors	Central-Europe	>50K
Senior-Secondary	Eastern-Europe	>50K
Assoc-voc	Southern-Europe	>50K
Bachelors	Western-Europe	>50K

Cube6_4

ed_level2	nc_level1	salary
University	Central-Europe	<=50K
Secondary	Eastern-Europe	<=50K
Secondary	Southern-Europe	<=50K
Assoc	Western-Europe	<=50K

Fig. 8. Cube scenario 6.



Cube7os

ed_level1	nc_level1	hours_per_week
Senior-Secondary	Central-Europe	55
Bachelors	Eastern-Europe	65
Senior-Secondary	Southern-Europe	75
Senior-Secondary	Western-Europe	62

Cube7_2

ed_level1	nc_level2	hours_per_week
Bachelors	Europe	40
Senior-Secondary	Europe	50
Junior-Secondary	Europe	40
Assoc-acdm	Europe	40

Cube7_3

ed_level1	nc_level1	hours_per_week
Bachelors	Central-Europe	40
Some-college	Eastern-Europe	40
Junior-Secondary	Southern-Europe	40
Assoc-acdm	Western-Europe	40

Cube7_6

ed_level2	nc_level1	hours_per_week
University	Central-Europe	40
Secondary	Eastern-Europe	50
Secondary	Southern-Europe	40
Assoc	Western-Europe	40

Cube7_1

ed_level1	nc_level1	hours_per_week
Assoc-acdm	Western-Europe	40
5th-6th	Southern-Europe	55
Masters	Central-Europe	30
Senior-Secondary	Western-Europe	40
Some-college	Western-Europe	42
Bachelors	Central-Europe	40

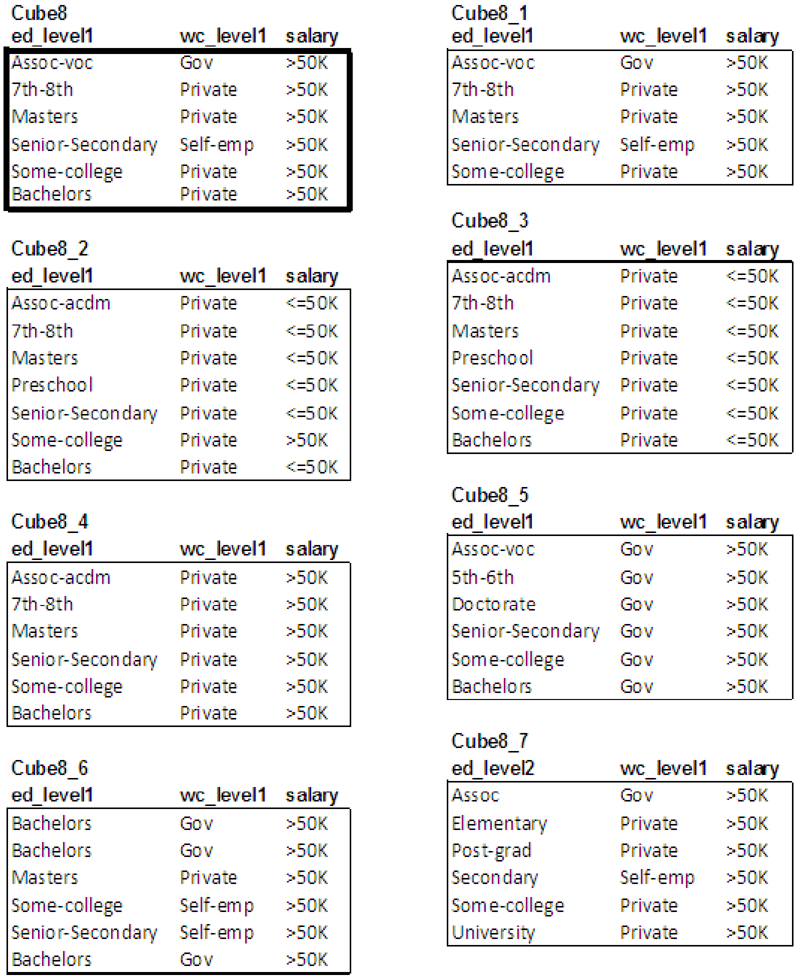
Cube7_4

ed_level1	nc_level1	hours_per_week
Bachelors	Central-Europe	40
Bachelors	Eastern-Europe	40
Bachelors	Southern-Europe	50
Bachelors	Western-Europe	40

Cube7_5

ed_level1	nc_level1	hours_per_week
Prof-school	Middle-America	60
Some-college	North-America	80
Senior-Secondary	South-America	72

Fig. 9. Cube scenario 7.



Cube8

ed_level1	wc_level1	salary
Assoc-voc	Gov	>50K
7th-8th	Private	>50K
Masters	Private	>50K
Senior-Secondary	Self-emp	>50K
Some-college	Private	>50K
Bachelors	Private	>50K

Cube8_2

ed_level1	wc_level1	salary
Assoc-acdm	Private	<=50K
7th-8th	Private	<=50K
Masters	Private	<=50K
Preschool	Private	<=50K
Senior-Secondary	Private	<=50K
Some-college	Private	>50K
Bachelors	Private	<=50K

Cube8_4

ed_level1	wc_level1	salary
Assoc-acdm	Private	>50K
7th-8th	Private	>50K
Masters	Private	>50K
Senior-Secondary	Private	>50K
Some-college	Private	>50K
Bachelors	Private	>50K

Cube8_6

ed_level1	wc_level1	salary
Bachelors	Gov	>50K
Bachelors	Gov	>50K
Masters	Private	>50K
Some-college	Self-emp	>50K
Senior-Secondary	Self-emp	>50K
Bachelors	Gov	>50K

Cube8_1

ed_level1	wc_level1	salary
Assoc-voc	Gov	>50K
7th-8th	Private	>50K
Masters	Private	>50K
Senior-Secondary	Self-emp	>50K
Some-college	Private	>50K

Cube8_3

ed_level1	wc_level1	salary
Assoc-acdm	Private	<=50K
7th-8th	Private	<=50K
Masters	Private	<=50K
Preschool	Private	<=50K
Senior-Secondary	Private	<=50K
Some-college	Private	<=50K
Bachelors	Private	<=50K

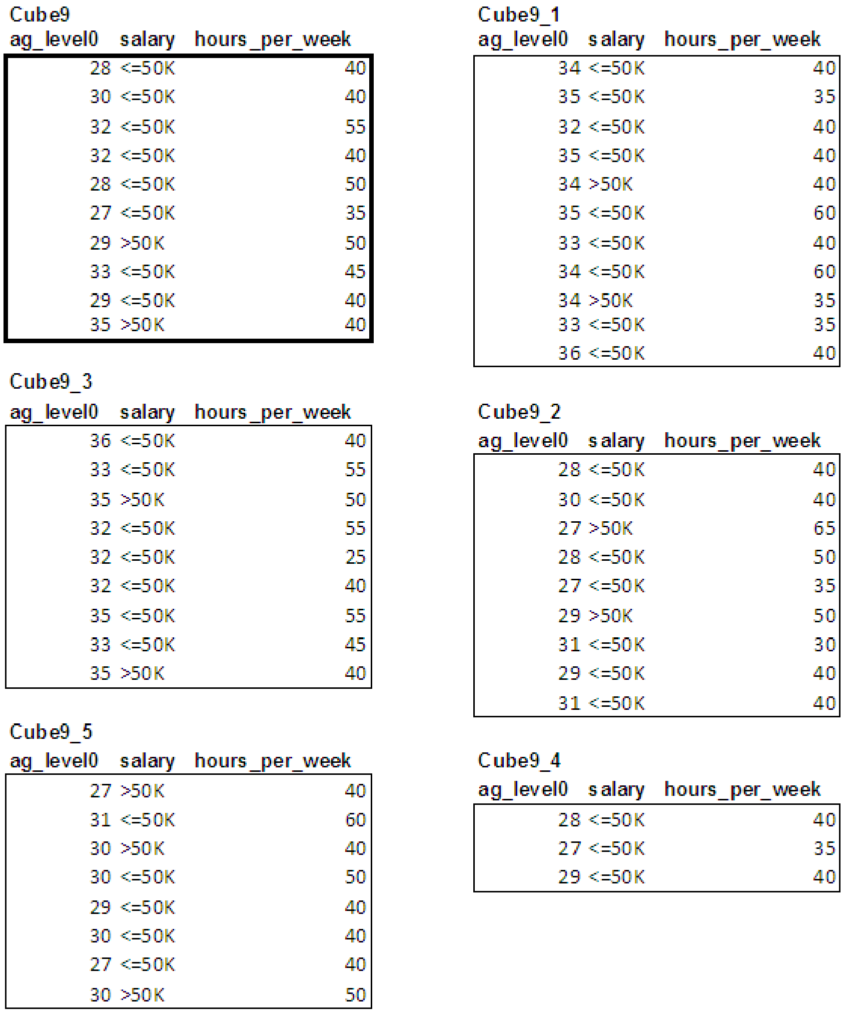
Cube8_5

ed_level1	wc_level1	salary
Assoc-voc	Gov	>50K
5th-6th	Gov	>50K
Doctorate	Gov	>50K
Senior-Secondary	Gov	>50K
Some-college	Gov	>50K
Bachelors	Gov	>50K

Cube8_7

ed_level2	wc_level1	salary
Assoc	Gov	>50K
Elementary	Private	>50K
Post-grad	Private	>50K
Secondary	Self-emp	>50K
Some-college	Private	>50K
University	Private	>50K

Fig. 10. Cube scenario 8.



Cube9

ag_level0	salary	hours_per_week
28	<=50K	40
30	<=50K	40
32	<=50K	55
32	<=50K	40
28	<=50K	50
27	<=50K	35
29	>50K	50
33	<=50K	45
29	<=50K	40
35	>50K	40

Cube9_3

ag_level0	salary	hours_per_week
36	<=50K	40
33	<=50K	55
35	>50K	50
32	<=50K	55
32	<=50K	25
32	<=50K	40
35	<=50K	55
33	<=50K	45
35	>50K	40

Cube9_5

ag_level0	salary	hours_per_week
27	>50K	40
31	<=50K	60
30	>50K	40
30	<=50K	50
29	<=50K	40
30	<=50K	40
27	<=50K	40
30	>50K	50

Cube9_1

ag_level0	salary	hours_per_week
34	<=50K	40
35	<=50K	35
32	<=50K	40
35	<=50K	40
34	>50K	40
35	<=50K	60
33	<=50K	40
34	<=50K	60
34	>50K	35
33	<=50K	35
36	<=50K	40

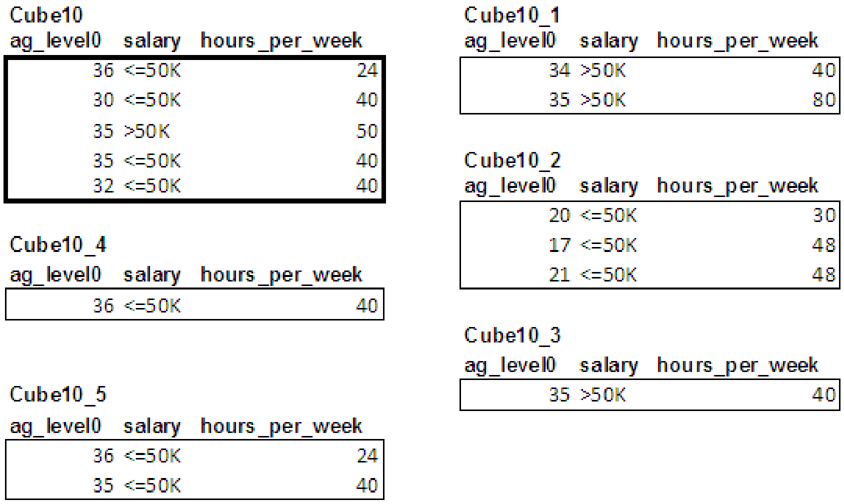
Cube9_2

ag_level0	salary	hours_per_week
28	<=50K	40
30	<=50K	40
27	>50K	65
28	<=50K	50
27	<=50K	35
29	>50K	50
31	<=50K	30
29	<=50K	40
31	<=50K	40

Cube9_4

ag_level0	salary	hours_per_week
28	<=50K	40
27	<=50K	35
29	<=50K	40

Fig. 11. Cube scenario 9.



Cube10

ag_level0	salary	hours_per_week
36	<=50K	24
30	<=50K	40
35	>50K	50
35	<=50K	40
32	<=50K	40

Cube10_4

ag_level0	salary	hours_per_week
36	<=50K	40

Cube10_5

ag_level0	salary	hours_per_week
36	<=50K	24
35	<=50K	40

Cube10_1

ag_level0	salary	hours_per_week
34	>50K	40
35	>50K	80

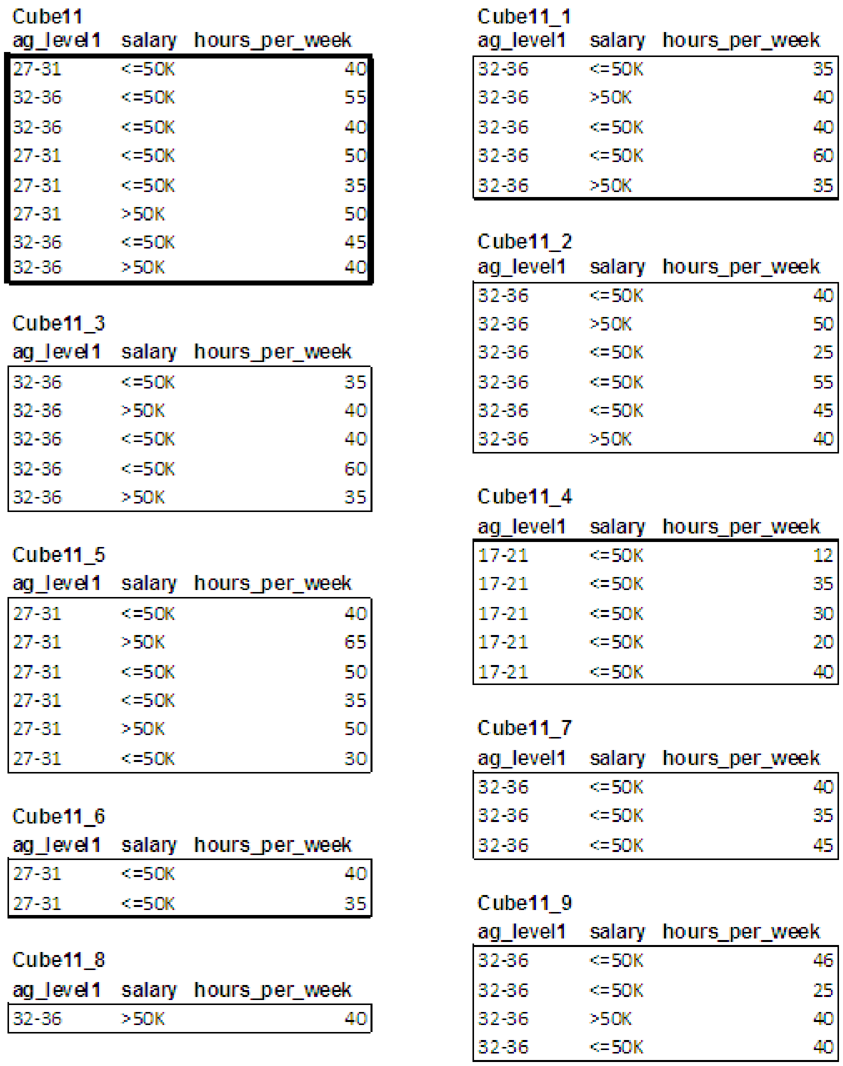
Cube10_2

ag_level0	salary	hours_per_week
20	<=50K	30
17	<=50K	48
21	<=50K	48

Cube10_3

ag_level0	salary	hours_per_week
35	>50K	40

Fig. C12. Cube scenario 10.



Cube11

ag_level1	salary	hours_per_week
27-31	<=50K	40
32-36	<=50K	55
32-36	<=50K	40
27-31	<=50K	50
27-31	<=50K	35
27-31	>50K	50
32-36	<=50K	45
32-36	>50K	40

Cube11_1

ag_level1	salary	hours_per_week
32-36	<=50K	35
32-36	>50K	40
32-36	<=50K	40
32-36	<=50K	60
32-36	>50K	35

Cube11_2

ag_level1	salary	hours_per_week
32-36	<=50K	40
32-36	>50K	50
32-36	<=50K	25
32-36	<=50K	55
32-36	<=50K	45
32-36	>50K	40

Cube11_3

ag_level1	salary	hours_per_week
32-36	<=50K	35
32-36	>50K	40
32-36	<=50K	40
32-36	<=50K	60
32-36	>50K	35

Cube11_4

ag_level1	salary	hours_per_week
17-21	<=50K	12
17-21	<=50K	35
17-21	<=50K	30
17-21	<=50K	20
17-21	<=50K	40

Cube11_5

ag_level1	salary	hours_per_week
27-31	<=50K	40
27-31	>50K	65
27-31	<=50K	50
27-31	<=50K	35
27-31	>50K	50
27-31	<=50K	30

Cube11_7

ag_level1	salary	hours_per_week
32-36	<=50K	40
32-36	<=50K	35
32-36	<=50K	45

Cube11_6

ag_level1	salary	hours_per_week
27-31	<=50K	40
27-31	<=50K	35

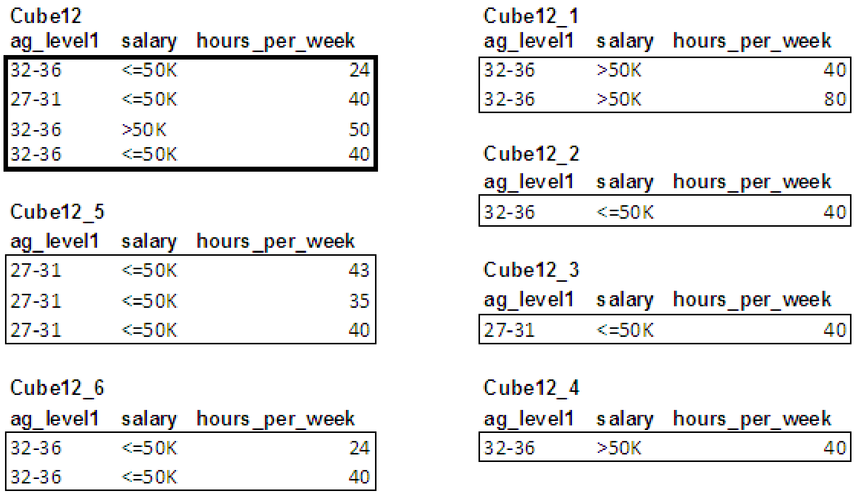
Cube11_9

ag_level1	salary	hours_per_week
32-36	<=50K	46
32-36	<=50K	25
32-36	>50K	40
32-36	<=50K	40

Cube11_8

ag_level1	salary	hours_per_week
32-36	>50K	40

Fig. C13. Cube scenario 11.



Cube12

ag_level1	salary	hours_per_week
32-36	<=50K	24
27-31	<=50K	40
32-36	>50K	50
32-36	<=50K	40

Cube12_1

ag_level1	salary	hours_per_week
32-36	>50K	40
32-36	>50K	80

Cube12_2

ag_level1	salary	hours_per_week
32-36	<=50K	40

Cube12_5

ag_level1	salary	hours_per_week
27-31	<=50K	43
27-31	<=50K	35
27-31	<=50K	40

Cube12_3

ag_level1	salary	hours_per_week
27-31	<=50K	40

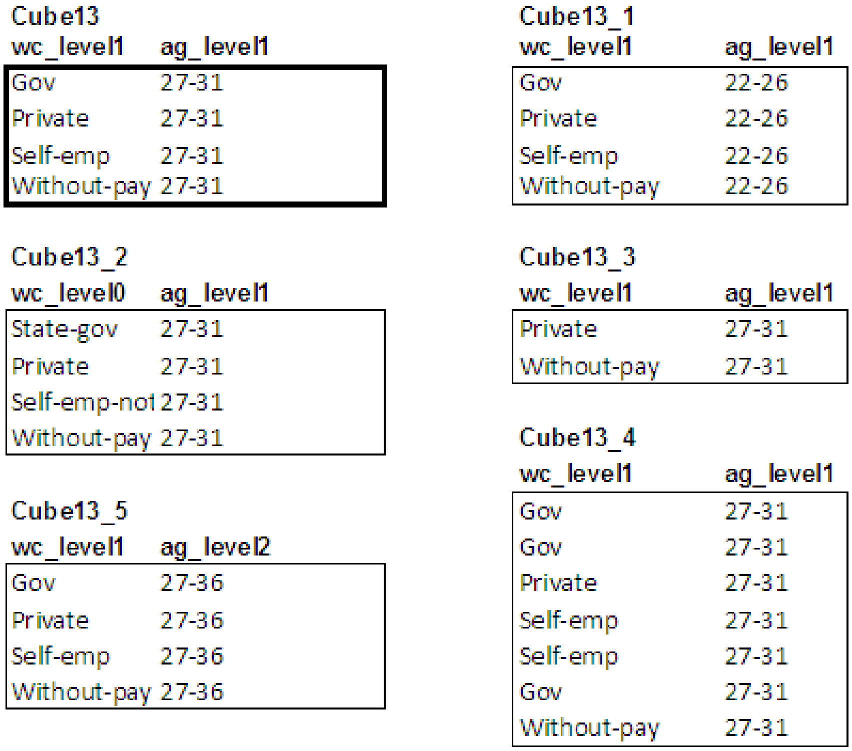
Cube12_6

ag_level1	salary	hours_per_week
32-36	<=50K	24
32-36	<=50K	40

Cube12_4

ag_level1	salary	hours_per_week
32-36	>50K	40

Fig. C14. Cube scenario 12.



Cube13

wc_level1	ag_level1
Gov	27-31
Private	27-31
Self-emp	27-31
Without-pay	27-31

Cube13_1

wc_level1	ag_level1
Gov	22-26
Private	22-26
Self-emp	22-26
Without-pay	22-26

Cube13_2

wc_level0	ag_level1
State-gov	27-31
Private	27-31
Self-emp-not	27-31
Without-pay	27-31

Cube13_3

wc_level1	ag_level1
Private	27-31
Without-pay	27-31

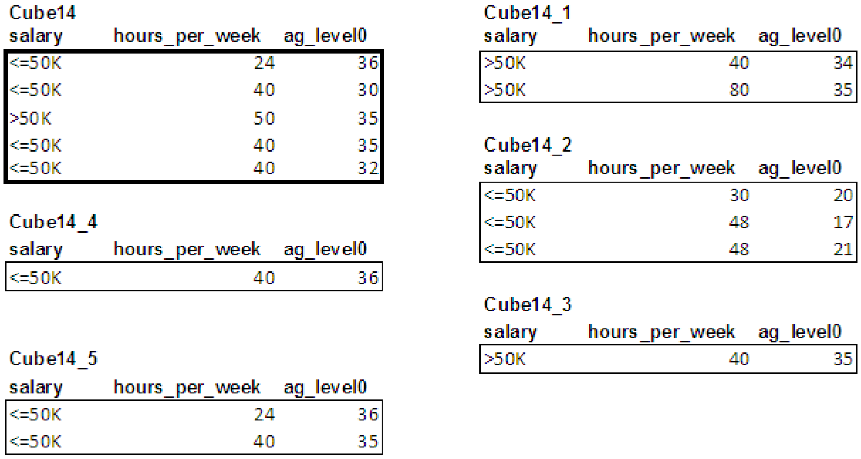
Cube13_5

wc_level1	ag_level2
Gov	27-36
Private	27-36
Self-emp	27-36
Without-pay	27-36

Cube13_4

wc_level1	ag_level1
Gov	27-31
Gov	27-31
Private	27-31
Self-emp	27-31
Self-emp	27-31
Gov	27-31
Without-pay	27-31

Fig. C15. Cube scenario 13.



Cube14

salary	hours_per_week	ag_level0
<=50K	24	36
<=50K	40	30
>50K	50	35
<=50K	40	35
<=50K	40	32

Cube14_4

salary	hours_per_week	ag_level0
<=50K	40	36

Cube14_5

salary	hours_per_week	ag_level0
<=50K	24	36
<=50K	40	35

Cube14_1

salary	hours_per_week	ag_level0
>50K	40	34
>50K	80	35

Cube14_2

salary	hours_per_week	ag_level0
<=50K	30	20
<=50K	48	17
<=50K	48	21

Cube14_3

salary	hours_per_week	ag_level0
>50K	40	35

Fig. C16. Cube scenario 14.