Similarity Measures for Multidimensional Data

Eftychia Baikousi, Georgios Rogkakos, Panos Vassiliadis

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

Abstract— **How similar are two data-cubes? In other words, the question under consideration is: given two sets of points in a multidimensional hierarchical space, what is the distance value between them? In this paper we explore various distance functions that can be used over multidimensional hierarchical spaces. We organize the discussed functions with respect to the properties of the dimension hierarchies, levels and values. In order to discover which distance functions are more suitable and meaningful to the users, we conducted two user study analysis. The first user study analysis concerns the most preferred distance function between two values of a dimension. The findings of this user study 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. The second user study aimed in discovering which distance function between two data cubes, is mostly preferred by users. The two functions that drew the attention of users where (a) the summation of distances between every cell of a cube with the most similar cell of another cube and (b) the Hausdorff distance function. Overall, the former function was preferred by users than the latter; however the individual scores of the tests indicate that this advantage is rather narrow.**

II. 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 [13]. 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) organizing alternative distance functions in a taxonomy of functions and (b) experimentally assessing the effectiveness of each distance function via a user study. The novelty of our work is not in the suggestion of new distance functions, but rather, it lies (a) in the adjustment of existing distance functions in the OLAP setting and (b) in their evaluation –via two user studies- in order to discover which distance function is mostly preferred by the users.

In the related literature there are a number of papers that have pointed out the necessity of having appropriate similarity measures in order to discover objects that are similar to each other and measure in a quantitative way the distance among them. Most of them examine similarity measures used between objects that are described from various features such as in image retrieval or data that are stored in a hierarchical taxonomy. Notably, [8] and [9] describe how similarity measures used by human perception and computer science follow different properties. The authors provide a collection of references where the metric axioms have been refuted. Computer scientists in the areas of data mining and information retrieval have also considered the problem of introducing appropriate similarity measures. Few papers have associated the areas of mathematics and computer science and have introduced similarity measures for lattices by mapping them with semantic hierarchies [4].

So far, related work have dealt with similar problems in different ways; however, this particular problem has not been dealt per se. Specifically, Sarawagi in [10] and [11] 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 element of this work with ours is the usage of the Manhattan distance in the process 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. [2] 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 [5] 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 of an OLAP cube.

The main findings of our approach are due to two user studies that have been conducted to assess which distance functions appear to work better for the users (Section III). The first experiment involved 15 users of various backgrounds and the *Adult* real dataset [1]. 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 between two values of a dimension is preferred depending on the user group or the type of data. The first user study showed

978-1-4244-8958-9/11/$26.00 © 2011 IEEE

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 either on the hierarchical shortest path, or with regard to ancestor values.

The second user study involved 39 users and the results of the first user study were taken into account. Each user was given 14 scenarios that contained a reference cube and three variant cubes. The purpose of this second user study concerns the most preferred distance function between two data cubes. Two distance functions have been in the center of attention in this study: the Hausdorff distance function and the *closest relative* function that sums the individual distances of cells of the two cubes. The latter has been selected by users at a remarkably higher percentage of occasions than the former (57% vs. 38%); however, if one considers the winner per scenario the result is only 6 vs. 5 in favor of closest relatives. Thus, we conclude that although the closest relative has an advantage over Hausdorff, this cannot be overemphasized.

Roadmap. We start by (Section II) providing a taxonomy of distance functions for cubes based on a detailed study of the characteristics of dimension hierarchies, levels and members. At first, we organize our families of functions as follows: Initially we describe functions that can be applied between two specific values that belong to the same dimension (section II-A). Following, we describe distance functions that are applied between two cells of a cube (section II-B) and then distance functions between two OLAP cubes (section II-C). Finally, section III presents the user study experiments along with the results of the most preferred distance functions.

III. Distance Families

In this section, we organize the distance functions that can be used to measure the distance between two cubes in a taxonomy. The formal foundations of modeling multidimensional spaces and cubes are based on an existing model in the related literature [12]. We build our taxonomy of distances progressively: In section II-A, we describe the functions that can be applied between two values for a given dimension. In section II-B we provide a taxonomy for distance functions between two cells of cubes and in section II-C a taxonomy for distance functions between two OLAP cubes. The distance functions described are all normalized within the interval [0, 1] and in many cases, such as in the *weighted sum* distance function, weight factors may be used. The normalization and usage of weight factors in the distance functions is not obligatory. Throughout all our deliberations we will refer to two reference dimensions, *Time* and *Location*. The hierarchies of these dimensions are shown in Figure 1. 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). Figure 2 illustrates the lattice of the dimension *Location* at the instance level.

A. Distance Functions between two Values

In this section, we specify the distance functions that can be applied over two specific values of a dimension. In order to clarify things, distance functions described in this section apply only between two dimension values and not between measure values of a cube.

Assume a dimension *D*, its lattice of level hierarchies $L_1 \prec L_2 \prec \ldots \prec ALL$, and two specific values *x* and *y* from levels of hierarchy L_x and L_y respectively. We classify the distance functions in the following categories: (*1*) *locally computable* and (*2*) *hierarchical computable* distance functions.

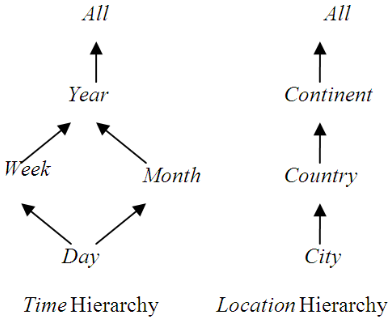


Figure 1. The hierarchy of levels for dimensions *Time* and *Location*

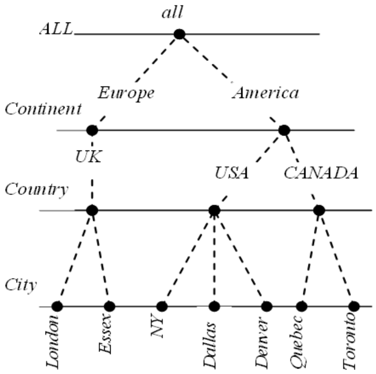


Figure 2. Values of the *Location* dimension

1) Locally Computable Distance Functions

The first category of locally computable distance functions can be divided into three subcategories: (a) Distance functions with explicit assignment of values, (b) Distance functions 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) (the compared values must belong to the same level of the hierarchy). This requires *dom*(L_i) to be a finite set. For example, the distance between two cities can be 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). For instance, assume a dimension *Products* accompanied with an attribute *Weight* which describes the weight of the products and assume a level of hierarchy of the dimension named *Drinks*. In addition, assume two specific values x = 'milk' and y = 'orange juice' where their weight attributes are $x.weight = 500$ and $y.weight = 330$ respectively. Then, the distance between these two values can be expressed according to their weight attribute by making use, for instance, of the Minkowski distance function which is described in the following subsection. Thus, the distance between the values x and y can be defined as $|x.weight - y.weight| = 170$.

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. This function is applicable for all type values even for nominal values. A first option is to use of the simple identity function, resulting in a value from the set {0, 1}, where $dist(x, y) = 0$ if $x=y$, or $dist(x, y) = 1$ if $x \neq y$.

Another option is to make use of the Minkowski family distance functions especially when the values are of interval type. Minkowski family distance functions can be applied between two ordinal type values under the condition that the ordinal values have been mapped to the set of integer numbers. 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 $|x\text{-}y|$. In order to normalize this function within the interval [0, 1], we can divide the distance value by the difference between the maximum and minimum values of the level where x and y belong to.

2) Hierarchical Computable Distance Functions

The second category of hierarchical computable distance functions can be divided into four subcategories: (a) Distance functions with respect to an aggregation function, (b) Distance functions with respect to hierarchy path, (c) Percentage distance functions and (d) Highway distance functions.

Distance functions with respect to an Aggregation Function. The distance for two values that do not belong to the detailed level L_1 can be expressed with respect to an aggregation function (e.g., *count*, *max*) applied over the descendants of the two values in a lower level of hierarchy.

Assume an instance x from level L_i and $desc_{L_\mathrm{L}}^{L_\mathrm{i}}(x)$ 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 $desc_{L_\mathrm{L}}^{L_\mathrm{i}}(x)$ is denoted as $x_\mathrm{aggr} = f_\mathrm{aggr}(desc_{L_\mathrm{L}}^{L_\mathrm{i}}(x))$. Assume two values x and y with $x_\mathrm{aggr} = f_\mathrm{aggr}(desc_{L_\mathrm{L}}^{L_\mathrm{i}}(x))$ and $y_\mathrm{aggr} = f_\mathrm{aggr}(desc_{L_\mathrm{L}}^{L_\mathrm{j}}(y))$, where L_L could be any lower level of L_i and L_j, $x \in L_\mathrm{i}$, $y \in L_\mathrm{j}$ and f_aggr denotes an aggregation function such as *count*, *min*, *max*, *avg* or *sum*. The distance between the values x and y can now be expressed according to the following formula: $dist(x, y) = g(x_\mathrm{aggr}, y_\mathrm{aggr})$, where the function g can be computed from the locally computable functions. The normalized form of this function, within the interval [0, 1], can be expressed as $dist(x, y) = \dfrac{g(x_\mathrm{aggr}, y_\mathrm{aggr})}{max\{g(a_\mathrm{aggr}, b_\mathrm{aggr})\}}$, where a and b are any possible values from the same level of hierarchy as x and y, i.e., $a, b \in L_\mathrm{i}$.

Distance Functions 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 were described by Li, Bandar and McLean in [5]. Here, we describe the 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 $x \in L_\mathrm{x}$ and $y \in L_\mathrm{y}$. We denote the *Lowest Common Ancestor* of x and y as $lca(x,y)$.

The lowest common ancestor $lca(x,y)$, of two values x and y --where $x \in L_\mathrm{x}$ and $y \in L_\mathrm{y}$, $lca(x,y) \in L_\mathrm{z}$ and L_z is any non lower level of L_x and L_y, $L_\mathrm{z} \succ L_\mathrm{x}, L_\mathrm{y}$ -- is a value such that:

$$lca(x,y) = \{z | z = anc_{L_x}^{L_z}(x) \wedge z = anc_{L_y}^{L_z}(y) \wedge (\nexists\ z' \mid z' = anc_{L_x}^{L_z}(x) \wedge z' = anc_{L_y}^{L_z}(y) \wedge L_{z'} \prec L_z\}$$

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

$$d_{path}(x,y) = \frac{w_\mathrm{x} * |path(x, lca)| + w_y * |path(y, lca)|}{(w_\mathrm{x} + w_\mathrm{y}) * |path(ALL, L_1)|}$$

$$d_{depth}(x, y) = \frac{|path(lca, L_1)|}{|path(ALL, L_1)|}$$

The first formula indicates that the distance is expressed as the weighted sum of the length of the path from the values x and y to their lowest common ancestor *lca*. The second formula indicates that the distance of the values is expressed as the length of the path of the lowest common ancestor *lca* from the detailed level L_1 of the hierarchy. These two functions are normalized in the interval [0, 1] by making use of the height of the hierarchy. Specifically, the first formula is divided by $(w_\mathrm{x} + w_\mathrm{y}) * |path(ALL, L_1)|$ whereas the second formula is divided by $|path(ALL, L_1)|$. As an example, assume two values x='NY' and y='Canada' from the hierarchy *Location* denoted in Figure 2 where their lowest common ancestor is the value *lca* = 'America' from the level *Continent*. For simplicity, assume the weighted factors w_x and w_y are set to 1. Therefore, the functions become: d_path= $(|path(x, lca)| + |path(y, lca)|)/ 2*|path(ALL, L_1)|$ and d_depth= $|path(lca, L_1)|/|path(ALL, L_1)|$. The distance between x and y occurs to be d_path= (2+1)/2*3 =0.5 and d_depth=2/3.

Percentage Distance 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 $L_1 \prec \ldots \prec L_\mathrm{L} \prec L_\mathrm{x} \prec L_\mathrm{y} \prec All$ where L_1 denotes the most detailed level. The distance of a value x in a level L_x with regard to its ancestor y in level L_y may be calculated according to the function:

$$dist(x,y)=\frac{|\,desc_{L_i}^{L_x}(x)\,|}{|\,desc_{L_i}^{L_y}(y)\,|}$$
, where L_i is one of L_x, L_L and L_1.

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 with regard to the occurrences of all the other values from L_x whose ancestor is y. In case L_i is L_L(or L_1), the distance is expressed as a percentage of occurrences of the descendants of x in a lower level of hierarchy L_L(or L_1) with regard to the descendants of y in the same lower level L_L(or L_1). As an example, assume the dimension *Location* where its lattice can be visualized in figure 1 and the values of this dimension are visualized in figure 2. Assume the values x='USA' and y='America'. Then, with regard to the above formula the distance between these two values can be computed as:

$$dist('\text{USA}','\text{America}')=\frac{1}{|\,desc_{Country}^{Continent}('\text{America}')\,|}=\frac{1}{2}$$

where L_i is chosen to be the level L_x, i.e., $L_{country}$

$$dist('\text{USA}','\text{America}')=\frac{|\,desc_{City}^{Country}('\text{USA}')\,|}{|\,desc_{City}^{Continent}('\text{America}')\,|}=\frac{3}{5}$$

where L_i is chosen to be the detailed level L_1, i.e., L_{city}

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

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 two representatives can be thought of as a highway [7]. We denote with $r(x)$ and $r(y)$ the representatives of the groups where x and y belong to respectively. Therefore, the distance between the values x and y can be expressed with the following formula:

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

The partial distances between a value and its representative and the distance between the two representatives, $r(x)$ and $r(y)$, depends on the way the representative is selected. In most cases, the representatives are selected so that they belong to the same level of hierarchy and thus their distance can be computed from the locally computable functions, the path functions or the aggregated functions (in case the two representatives belong to different levels their distance may be computed by applying any distance function from the path section or the aggregated distance function section). The main categories of selecting the representative apart from an explicit assignment are with regard to (a) an ancestor and (b) a descendant. For the following, $dist(a, b)$ denotes the distance of any two values a, b. Without loss of generality assume $L_x \prec L_y$ (see Fig. 3). In addition, assume the ancestor of x in level L_y is $x_y = anc_{L_x}^{L_y}(x)$ and 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. In the following, we describe the partial distances of the previous formula depending on the way the representative is selected.

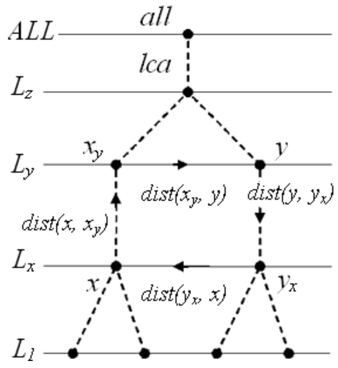


Figure 3. Partial distances between two values in different levels of hierarchy.

a) The representative of a group is an ancestor. The representative of each value x and y could be $r(x)=anc_{L_x}^{L_U}(x)$ and $r(y)=anc_{L_y}^{L_V}(y)$ where L_U and L_V is any upper level of L_x and L_y respectively. L_U and L_V are not obligatory different. In general, the distance between a value x and its representative may be computed through any distance function from the path, the percentage or the aggregated functions. For example, assume two values x='UK' and y='USA' from the level *Country* of the hierarchy *Location* denoted in figure 2. Assume the representative $r(x)$='Europe' and the representative $r(y)$='America'. The distance of the values x and y is by summing the distances *dist*('UK', 'Europe'), *dist*('Europe', 'America') and *dist*('America', 'USA'). In this category there are two special cases:

The representatives $r(x)$ and $r(y)$ coincide in being the lowest common ancestor *lca*, where the formula is simplified as: $dist\,(x, y) = dist\,(x, lca) + dist\,(y, lca)$.

The representative $r(y)$ is identical to the actual value of y. In this case the distance is expressed as a summation of $dist(x, x_y)$ and $dist(x_y, y)$, as shown in figure 3, where x_y is the representative of x from the level L_y. Therefore, the distance $dist(y, r(y)) = 0$. Formally this is expressed as: $dist(x,y)=$

$$dist(x,x_y)+dist(x_y,y)=dist(x,anc_{L_x}^{L_y}(x))+dist(anc_{L_x}^{L_y}(x),y)$$

In case the representative x_y of x and y coincides, the distance is simplified as $dist(x,y)=dist(x,x_y)$. Since $dist(x, x_y)$ and $dist(x_y, y)$ are within the interval [0, 1], the normalized form of $dist(x, y)$ occurs by dividing it by 2. For example, assume two values x = 'USA' and y = 'Europe' from the dimension *Location* as seen in figure 2. The ancestor x_y of x is $anc_{Country}^{Continent}(x)='\text{America}'$. Assume $dist(x, x_y)$ is computed from the percentage family functions. $dist(x_y, y)$ is computed through the first formula from the path family functions where the weighted factors w_x and w_y are set to 1. The distance between x and y becomes *dist*('USA', 'Europe')= ($dist(x, x_y)$ +

$dist(x_y, y))/2$ = ($dist$('USA', 'America') + $dist$('America', 'Europe'))/2 = (1/2 + 2/3)/2 = 7/12.

b) The representative of a group is a descendant. The representative of a group can be selected with respect to the descendants of the group where x belongs. For example, consider countries whose representatives can be selected among their cities, based for instance on the major airport or the highest population. In case the representative $r(x)$ is a value from the domain of L_L (i.e., $r(x)$ picked explicitly by applying a min or max aggregation over the set $desc_{L_x}^{L_L}(x)$), the distance between x and $r(x)$ can be any function from the families of path, percentage or aggregated functions. In case $r(x)$ is an arithmetic type value (i.e., a sum or count aggregation function over the set $desc_{L_x}^{L_L}(x)$), the distance between x and $r(x)$ can be any simple arithmetic function such as the Minkowski. There is a special case where the representative $r(x)$ is identical to the actual value of x. Thus, the distance is expressed as a summation of $dist(y, y_x)$ and $dist(y_x, x)$, where y_x is the representative of y from the level L_x as shown in figure 3. Therefore, the distance $dist(x, r(x))$=0. Formally this is expressed as:

$$dist(x,y) = \frac{dist(y, y_x) + dist(y_x, x)}{2} =$$

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

, where the denominator is set to 2 for normalization reasons. For 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 y_x is computed through the first formula from the path family functions with w_x and w_y set to 1. Then, the distance between x and y becomes

$$dist(\text{'USA'}, \text{'Europe'}) = \frac{dist(y, y_x) + dist(y_x, x)}{2} =$$

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

In the special case where x is a descendant of y the above formula is simplified as: $dist(x,y) = dist(y, y_x)$.

B. Distance Functions between two Cells of Cubes

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, …, L_n^0, M_1^0, M_2^0, …,M_m^0], where L_i^0 is a detailed level and M_j^0 is a detailed measure. In addition, assume two cells from this cube, c_1 = (l_1^1, l_2^1, …, l_n^1, m_1^1, m_2^1, …, m_m^1) and c_2 = (l_1^2, l_2^2, …, l_n^2, m_1^2, m_2^2, …, m_m^2), where l_i^1, $l_i^2 \in dom(L_i^0)$ and m_j^1, m_j^2 denote the values of the corresponding measure M_j^0. The distance between two cells c_1 and c_2 can be expressed with regard to (a) their level coordinates $d_i(L_i^1, L_i^2)$ and (b) their measure values $d_j(M_j^1, M_j^2)$. In other words, $dist(c_1, c_2)= f\,(d_i(L_i^1, L_i^2), d_j(M_i^1, M_i^2))$. The function f can possibly be (a) a weighted sum, (b) Minkowski, (c) min or (d) proportion of common coordinates.

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

$$\frac{\sum_{i=1}^{n} w_i d_i(l_i^1, l_i^2)}{\sum_{i=1}^{n} w_i} + \frac{\sum_{j=1}^{m} w'_j d_j(m_j^1, m_j^2)}{\sum_{j=1}^{m} w'_j}$$

, where w_i and w'_j are parameters that assign a weight for the level L_i and the measure M_j respectively, $d_i(l_i^1, l_i^2)$ denotes the partial distance between two values from dimension D_i and $d_j(m_j^1, m_j^2)$ denotes the partial distance between two instances of the measure M_j^0. Regarding the distance $d_i(l_i^1, l_i^2)$, this can be expressed through the various distance functions (section II-A) between two values from the same dimension. The distance $d_j(m_j^1, m_j^2)$ between two instances of a measure can be calculated through the Minkowski family distance when m_j^1, m_j^2 are of arithmetic type, or through the simple identity function in case m_j^1, m_j^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 with respect 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, the distance between two cells is normalized in the same interval [0, 1]. For example, assume we want to compute the distance between cells c_1, c_2 as shown in Figure 4. Both cells consist of two dimensions (*Time, Location*), with the hierarchy levels of Fig. 1, and contain one measure (*Sales*). In the above formula we set all the weight factors to 0.5 --both for dimensions (w) and measures (w'). The distance between dimensions is computed according to the function d_{path} that takes into account the length of the path of the hierarchy. The distance between the measures is computed through the normalized Manhattan distance function. In addition, assume that the overall maximum and minimum values of the measure sales are 10 and 1 respectively. Then, $d(c_1,c_2)$=

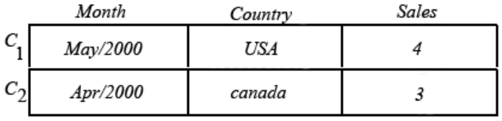
$$\frac{w * d(Month_{c_1}, Month_{c_2}) + w * d(Country_{c_1}, Country_{c_2})}{w + w} +$$

$$\frac{w' * d(Sales_{c_1}, Sales_{c_2})}{w'} =$$

$$\frac{0.5 * 1/3 + 0.5 * 1/3}{0.5 + 0.5} + \frac{0.5 * (|4-3| / |10-1|)}{0.5} = 4/9$$

To compute the distances $d(Month_{c_1}, Month_{c_2})$ and $d(Country_{c_1}, Country_{c_2})$ we refer the reader to Fig. 5 and 6.

In Figure 5 we see that the length of the path between the nodes *a* and *lca* is 1, and the length of the path between the nodes *b* and *lca* is 1 again. According to the function d_{path}, $d(Month_{c_1}, Month_{c_2}) = \frac{1+1}{6} = \frac{1}{3}$. In a similar manner, by using the information that derives from Figure 6 $d(Country_{c_1}, Country_{c_2}) = \frac{1+1}{6} = \frac{1}{3}$.



	Month	Country	Sales
c_1	May/2000	USA	4
c_2	Apr/2000	canada	3

Figure 4. Instances of cells c_1 and c_2

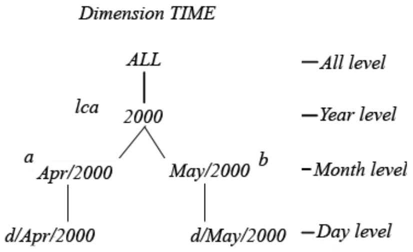


Figure 5. Lattice of the dimension TIME for the values of cells of Figure 4

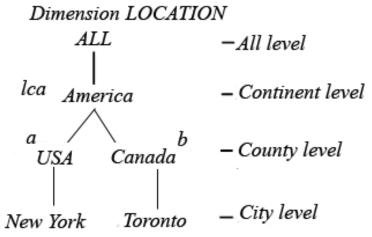


Figure 6. Lattice of the dimension LOCATION for the values of cells of Figure 4

2) Distance functions between two Cells of a Cube Expressed with regard to the Minkowski Family Distances.

In this section, we describe the possible distance functions between two cells of a cube by using the Minkowski family distances. In general, the Minkowski distance is defined via the formula $L_p[(x_1,...,x_n),(y_1,...,y_n)] = \sqrt[p]{\sum_{i=1}^{n} d_i(x_i,y_i)^p}$, 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 $l_i^1, l_i^2 \in dom(L_i)$ and m_j^1, m_j^2 denote the values of the corresponding measure M_j. 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 dimension (section II-A).

So far, the distance between two cells is described only with regard to their level coordinates. However, the distance between two cells can also be expressed by taking into consideration their measure values, too. The Minkowski family distances can be applied, as well, with regard to the partial distances $d_j(m_j^1, m_j^2)$. Therefore, the distance between two cells can be expressed by adding the equivalent two formulas. Depending on the value of *p* (1, 2, .., ∞) the Minkowski distance is defined as:

$$L_p = \sqrt[p]{\sum_{i=1}^{n}(d_i(l_i^1,l_i^2))^p} + \sqrt[p]{\sum_{j=1}^{m}(d_j(m_j^1,m_j^2))^p} .$$

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:

$$\min_{d_i}\{d_i(l_i^1,l_i^2)\} + \min_{d_j}\{d_j(m_j^1,m_j^2)\} =$$
$$\min\left\{d_1(l_1^1,l_1^2), d_2(l_2^1,l_2^2),..., d_n(l_n^1,l_n^2)\right\}$$
$$+ \min\left\{d_1(m_1^1,m_1^2), d_2(m_2^1,m_2^2),..., d_m(m_m^1,m_m^2)\right\}.$$

Therefore, the distance between two points is expressed as the minimum distance of their level coordinates plus the minimum distance of their measure values.

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*:

$$\frac{count(l_i^1 = l_i^2 \forall i \in \{1,2,...,n\})}{n} + \frac{count(m_j^1 = m_j^2 \forall j \in \{1,2,...,m\})}{m}$$

The above formula defines 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.

C. Distance Functions between two OLAP Cubes

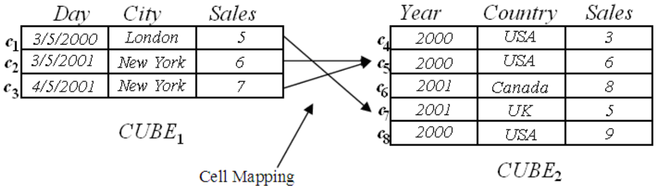
Assume two OLAP cubes *C* and *C'* defined over the same detailed schema $[L_1^0, L_2^0, \ldots, L_n^0, M_1^0, M_2^0, \ldots, M_m^0]$, where L_i^0 is a detailed level and M_j^0 is a detailed measure. In addition, assume that cube *C* consists of *l* cells of the form $c = (l_1, l_2, \ldots, l_n, m_1, m_2, \ldots, m_m)$ 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_j, m_j' denote the values of the corresponding measure M_j^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 II-B. 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) *closest relative*, (b) *Hausdorff* distance, (c) a *weighted sum*, (d) *Minkowski* distance, and (e) *Jaccard's coefficient*. For example, assume we want to compute the distance between the two cubes $CUBE_1$ and $CUBE_2$ as shown in Figure 7. $CUBE_1$ consists of three cells whereas $CUBE_2$ consists of 5 cells. Each cell in both cubes consists of two dimensions in different levels of hierarchy and the measure *Sales*. Specifically, each cell of $CUBE_1$ is of the form *c* = (*Day*, *City*, *Sales*) and each cell of $CUBE_2$ is of the form *c*' = (*Year*, *Country*, *Sales*). The distance between the two cubes can be expressed by applying a function *f* over the partial distances *dist*(*c*, *c*') of the cells of the two cubes.

1) Cell Mapping and Categories of Distance Functions according to it

The aforementioned function *f* can be computed either (i) over the full space of cell combinations of cells from the two cubes (families (a), (b) and (e)), or, (ii) over a specific subset of this space that is defined via a specific mapping of the cubes' cells (families (c) and (d)). In this section, we introduce the method that is used in order to map the cells of one cube to the cells of another cube. We refer to this method as *Cell Mapping*. For two cubes C_1 and C_2, the simple mapping of their cells includes the connection of every cell of the cube C_1 with one cell of the cube C_2. Intuitively, the mapping of a cell in cube C_1 tries to capture the discovery of the "closest possible representative" of this cell in cube C_2. The "closest representative" is the cell of the cube C_2 with the less distance among the dimension values with the cell of the cube C_1. In principle, the Cell Mapping method can be thought of as a relation that connects the cells of a cube to the cells of another cube (i.e., one can consider several candidate "representatives" of a cell). However, in our setting, this relation is reduced to a function, since we are interested in mapping each cell from the first cube to only one cell from the second cube. This is done for reasons of simplicity and allows the elegant definition of cube distances (see next). We impose the restriction that the function is total, i.e., each and every cell from the first cube is mapped to a cell of the second cube. We do not require that the mapping is 1:1 and onto; thus, in the second cube there might be a cell in which more than one cells from the first cube, or, no cells at all, are mapped to it.

As an example assume the cubes that are presented in the Figure 7. The cells c_1, c_2, c_3 of $CUBE_1$ are mapped to the cells c_7, c_5, c_5 of $CUBE_2$ respectively. Moreover, in the same figure the cells c_4, c_6, c_8 of $CUBE_2$ are not mapped with any cell of $CUBE_1$. We can also observe that the cell c_5 of $CUBE_2$ is mapped with two cells of $CUBE_1$.

The cell mapping method needs to compute the distances between the dimensions of each cell of the first cube with the dimensions of every cell of the second cube and ignores the distance between the measures. So, if the distance between two cells c_1, c_2 is expressed as $f(d_i(L_i^1, L_i^2), d_i(M_j^1, M_j^2))$, then the mapping method considers only the $d_i(L_i^1, L_i^2)$. Thus, each cell of the first cube is mapped to the cell of the second cube with the lowest $d_i(L_i^1, L_i^2)$ distance.



	Day	City	Sales
c_1	3/5/2000	London	5
c_2	3/5/2001	New York	6
c_3	4/5/2001	New York	7

$CUBE_1$

	Year	Country	Sales
c_4	2000	USA	3
c_5	2000	USA	6
c_6	2001	Canada	8
c_7	2001	UK	5
c_8	2000	USA	9

$CUBE_2$

Figure 7. Instances of two cubes and the mapping of their cells

In our taxonomy, two distance functions between cubes use the cell mapping method. These are (a) distance functions expressed with regard to the *Closest Relative* and (b) the distance function expressed by *Hausdorff* distance. After the mapping has been accomplished, the distances between the mapped cells are computed. Finally, the computation of the distance between the two cubes is performed on the basis of the distances among the mapped cells.

The distance functions that can be used in order to compute the distance between two OLAP cubes can be divided into two categories. The first category involves distance functions that include the cell mapping method. The second category contains distance functions that do not include the cell mapping method. Following, we describe each distance function and formally define it. The distance functions of the first category are the *Closest Relative* and the *Hausdorff Distance* (section II-C-2) that include the cell mapping method. Then, the category of families that do not consider the cell mapping method in their definition, include the *Weighted Sum* function, the *Minkowski family* of distance functions, the *Jaccard's Coefficient* and the *minimum of distances* function.

2) Distance Functions that Include Mappings

This subsection contains the description of the distance functions that involve the Cell Mapping method. These distance functions are the *Closest Relative* and the *Hausdorff* and are described as follows.

Distance function between two cubes expressed with regard 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 another cube through the formula:

$$\frac{\sum_{i=1}^{k}(dist(c_i,c'))}{k} \; \forall c' \mid dist_{\dim}(c_i,c') = \min\{dist_{\dim}(c_i,c')\}$$

where $dist_{\dim}$ denotes the distance of two cells excluding the distance of their measures. In the above formula, $\forall c' \mid dist_{\dim}(c_i,c') = \min\{dist_{\dim}(c_i,c')\}$ reveals the cell mapping. Each one of the *k* cells from cube *C* is mapped to the cell of the cube C' that has the minimum $dist_{\dim}$ from it.

As an example, we will detail the computation of the distance between the cubes $CUBE_1$ and $CUBE_2$ shown in Figure 7. The first step is to map the cells of the cube $CUBE_1$

to the appropriate cells of the cube $CUBE_2$. In order to simplify the example, the computational part of the cell mapping method is not described here, but the cell mapping is denoted in Figure 7 through arrows between the cells of the two cubes. The distance function used in this example for the purpose of computing the distance between the cells of the two cubes is the weighted sum. The weight that was used is 0.5, equal for both the dimensions and measures. In addition, the distance function used to measure the distance between the dimensions is the d_{path} function. The cells c_1, c_2, c_3, are mapped to the cells c_7, c_5, and c_5 respectively. According to this mapping, in order to compute the distance between the two cubes, the needed distances between cells are:

$$d(\mathrm{c}_1, \mathrm{c}_7) = \frac{0.5*1/6+0.5*1/6}{0.5+0.5}+\frac{0.5*(|5-5|/|10-1|)}{0.5} = 1/6$$

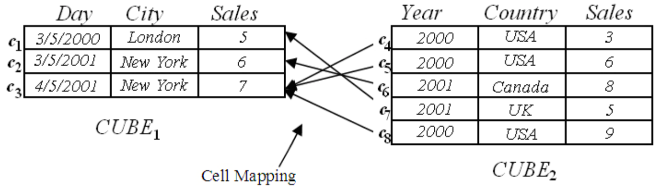
$$d(\mathrm{c}_2, \mathrm{c}_5)= \frac{0.5*1/6+0.5*1/6}{0.5+0.5}+\frac{0.5*(|6-6|/|10-1|)}{0.5} = 1/6$$

$$d(\mathrm{c}_3, \mathrm{c}_5)= \frac{0.5*1/6+0.5*1/6}{0.5+0.5}+\frac{0.5*(|6-7|/|10-1|)}{0.5} =5/18$$

For the above computations we refer the reader to Figures 5 and 6 where the hierarchies of the dimensions *TIME* and *LOCATION* are presented. With the above distances, we can now compute the full distance between the cubes $CUBE_1$ and $CUBE_2$ through the first formula of the *closest relative* family functions:

$$d(CUBE_1,CUBE_2)=\frac{d(c_1,c_7)+\ d(c_2,c_5)+\ d(c_3,c_5)}{3}=\frac{11}{54}$$

Distance functions between two cubes expressed by Hausdorff distance. In this category, the distance between two cubes can be expressed by using the *Hausdorff* distance [3]. The Hausdorff distance between two cubes can be defined as $H(C,C') = max(h(C,C'), h(C',C))$ where $\mathrm{h}(C,C') = \max\limits_{c\in C}\{\min\limits_{c'\in C'}\{dist(c,c')\}\}$ and *dist* (*c*, *c'*) is the distance between two cells *c* and *c*' from the cubes *C* and *C*' respectively. 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*).



Day	City	Sales
3/5/2000	London	5
3/5/2001	New York	6
4/5/2001	New York	7

Year	Country	Sales
2000	USA	3
2000	USA	6
2001	Canada	8
2001	UK	5
2000	USA	9

Figure 8. Instances of cubes CUBE1 and CUBE2 and the mapping of the cells of the cube CUBE2 to the cells of the cube CUBE1

In the Hausdorff distance function, the cell mapping method is bidirectional. That means that except from the mapping that we have examined in the closest relative function, we also need the extra mapping from the cells of cube *C*' to the cells of cube *C*.

When the bidirectional mapping is completed, we obtain two sets of mapped cells. In each set, for every pair of mapped cells, we compute their distance considering their measures as well. Thus, we have two sets of minimum distances between cells, the set of minimum distances from the cells of cube *C* to the cells of cube *C*' and the set of minimum distances between from the cells of cube *C*' to the cells of cube *C*. From each of the two sets we pick the greatest distance and finally from these two distances we pick the greater one.

To make things more clear, an example follows. Assume again cubes $CUBE_1$ and $CUBE_2$ as shown in Figure 8. In Figure 8, we can observe the mapping from the cells of $CUBE_2$ to the cells of $CUBE_1$. According to this bidirectional mapping the two resulting sets of minimum distances are:

$S_1\,\{d(c_1,c_7), d(c_2,c_5), d(c_3,c_5)\}$

$S_2\,\{d(c_4,c_3), d(c_5,c_3), d(c_6,c_2), d(c_7,c_1), d(c_8,c_3)\}$.

The distances of the set S_1 are already computed on a previous example, so here we only need to compute the distances of set S_2. The distances $d(c_5,c_3)$, $d(c_7,c_1)$ coincide with the distances $d(c_3,c_5)$, $d(c_1,c_7)$ respectively. The computations below use the same distance functions between values and cells and also the same weight factors, as in the previous example.

$$d(\mathrm{c}_4, \mathrm{c}_3)= \frac{0.5*1/6+0.5*1/6}{0.5+0.5}+\frac{0.5*(|3-7|/|10-1|)}{0.5}=\frac{11}{18}$$

$$d(\mathrm{c}_6, \mathrm{c}_2)= \frac{0.5*1/6+0.5*3/6}{0.5+0.5}+\frac{0.5*(|8-6|/|10-1|)}{0.5}=\frac{10}{18}$$

$$d(\mathrm{c}_8, \mathrm{c}_3)= \frac{0.5*1/6+0.5*1/6}{0.5+0.5}+\frac{0.5*(|9-7|/|10-1|)}{0.5}=\frac{7}{18}$$

Now, the Hausdorff distance between the cubes $CUBE_1$ and $CUBE_2$ is equal to the next formula:

$d(CUBE_1,CUBE_2)=max\{max\{S_1\},max\{S_2\}\}=$
$max\{max\{1/6,1/6,5/18\}, max\{11/18,5/18, 1/6,10/18,7/18\}\}=$
$max\{5/18,11/18\}=11/18$.

3) *Distance functions that do not include Mappings*

This subsection includes the distance functions that do not include mappings. These functions are the *Weighted Sum* function, the Minkowski family of distance functions, the *Jaccard's Coefficient* and the *minimum of distances* function. The *Weighted Sum* function is expressed through the formula: $\dfrac{\sum\limits_{\mathrm{i=1}}^{\mathrm{l}}\sum\limits_{\mathrm{j=1}}^{\mathrm{k}} w_{ij}dist(c_i,c'_j)}{\sum\limits_{\mathrm{i=1}}^{\mathrm{l}}\sum\limits_{\mathrm{j=1}}^{\mathrm{k}} w_{\mathrm{ij}}}$, where $dist(c_i,c'_j)$ 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.

The distance functions of the *Minkowski family* --depending on the values of the parameter *p* (1, 2, ..., ∞)-- can be expressed as: $L_{\mathrm{p}} = \sqrt[\mathrm{p}]{\sum\limits_{\mathrm{i=1}}^{\mathrm{l}}\sum\limits_{\mathrm{j=1}}^{\mathrm{k}} dist(c_i,c'_j)^{\mathrm{p}}}$, where $dist(c_i,c'_j)$ is the distance between a cell from cube *C* to a cell from cube *C*'.

The distance between two cubes can be expressed with regard to the *Jaccard's coefficient* [13]. The Jaccard's coefficient is defined as: $dist(C,C') = 1 - \frac{|C \cap C'|}{|C \cup C'|}$ and it expresses the ratio between the cardinalities of intersection and union of the cubes *C* and *C*'.

The *Minimum of distances* function expresses the distance between two cubes as the minimum distance among all possible distances between the cells of the compared cubes. Therefore, the distance between *C* and *C*' is expressed as: $dist(C,C') = \min\{dist(c,c') \mid c \in C, c' \in C'\}$, 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.

As a simple example, assume the two cubes from Figure 7 and ignore the arrows that denote the cell mapping. According to the *minimum of distances* function, the distance between the two cubes is computed through the following formula where *j* denotes any cell from *CUBE*$_2$: *d*(*CUBE*$_1$,*CUBE*$_2$)= $\min_j\{d(c_1,c_j), d(c_2,c_j), d(c_3,c_j)\}, \forall j \in \{4,5,\ldots,8\}$ =1/6.

IV. Experiments

A. User Study for Distances between two Values of Dimensions

In this section, we describe a user study we conducted for discovering which distance functions between two values of a dimension seem to be more suitable for user needs. The experiment involved 15 users 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 the set of all users independently of their background as *Users_all.*

Table I.
Adult dataset tables

	Value Type	Tuples	Dim. Levels
Adult fact Table		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

In the experiments we used the "Adult" real data set according to the dimension hierarchies as described in [1]. This dataset contains the fact table *Adult* and 8 dimension tables which are described in Table I.

The purpose of the experiment is to assess which distance function between two values is best with regard to the user preferences. Each user was given 14 case scenarios. Each scenario contained a reference cube and a set of cubes, which we call *variant* cubes, that occurred by slightly altering the reference cube. The 14 scenarios included different kinds of cubes with regard to the value types and the different levels of granularity. For each reference cube which was randomly selected, the variant cubes were generated from the fact table by altering the granularity level for one dimension, or by altering the value range of the reference cube. For instance, assume a reference cube containing 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 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 manually chose the set of variant cubes such that each of them was most similar to the reference cube according to a distance function. In order to observe which distance function is preferred by users depending on the type of data of the cubes, we have organized the 14 scenarios into 3 sets. The first set consists of cubes containing only arithmetic type values (5 scenarios). The second set consists of cubes containing only categorical type values (2 scenarios). The third set consists of cubes containing a combination of both categorical and arithmetic type values (7 scenarios). Due to space limitations all the scenarios used for this user study can be found in [6].

Table II.
Notation of distance functions used in the experiment

Family	Abbr.	Distance function name
Local	δ_{M}	Manhattan
Aggregation	$\delta_{Low,c}$	With respect to a lower level of hierarchy where f_{aggr} =count
	$\delta_{Low,m}$	With respect to a lower level of hierarchy where f_{aggr} = max
Hierarchical Path	$\delta_{LCA,P}$	Lowest common ancestor through d_{path}
	$\delta_{LCA,D}$	Lowest common ancestor through d_{depth}
Percentage	$\delta_{\%}$	Applying percentage function
Highway	δ_{Anc}	With respect to an ancestor x_y
	δ_{Desc}	With respect to a descendant y_x
	$\delta_{H,Desc}$	Highway, selecting the representative from a descendant
	$\delta_{H,Anc}$	Highway, selecting the representative from an ancestor

In each scenario, the users were asked to select the variant cube that seemed more similar to the reference cube based on their personal criteria. The distance functions that have been used in the experiment are shown in Table II, where the first column shows the family in which each distance function belongs to according to section II-A. In the second column there is an abbreviated name for each function. To compute the distance between two cubes, the *Closest Relative* distance function is used (section II-C-2). The distance between two cells of cubes is the weighted sum of the partial distances of the two values, one from each cell, with all weights set to 1 (section II-B-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 and specifically, these are the $\delta_{LCA,P}$, the δ_{Anc} and the $\delta_{H,Desc}$. The frequencies for each one of the top three distance functions in each group of users is shown in Table III.

TABLE III
TOP THREE MOST FREQUENT DISTANCE FUNCTIONS FOR EACH USER GROUP.

	Users_all	Users_cs	Users_non
$\delta_{LCA,P}$	40.47%	38.57%	44.28%
δ_{Anc}	18.09%	20.00%	14.28%
$\delta_{H,Desc}$	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 IV summarizes the result of the most frequent distance function for each set of scenarios and each set of users. We observe that for the *categorical* type of cubes, all user groups prefer the $\delta_{LCA,P}$ distance function, whereas for the *arithmetic* and the *arithmetic & categorical* sets, the functions that users mainly prefer are the $\delta_{LCA,P}$ and δ_{Anc}. More than one distance functions appear as winners in Table IV due to ties in the frequency of occurrences for each function.

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 most frequent winner function was $\delta_{LCA,P}$ with a 35.71% percentage for both the *Users_all* and the *Users_cs* group (5 of the 14 scenarios), and 57.14% for the *Users_non* group (8 of the 14 scenarios). The most frequent function for 14 of the 15 users was the $\delta_{LCA,P}$ function. For one user from the *Users_cs* group the most frequent function was the $\delta_{LCA,D}$.

TABLE IV
THE MOST FREQUENT DISTANCE FUNCTION FOR EACH SET OF SCENARIOS.

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

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

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

TABLE V
FREQUENCIES OF PREFERRED DISTANCES WITHIN EACH USER GROUP FOR EACH DISTANCE FAMILY.

	Local	Aggre-gation	Hierarchy Path	Perce-ntage	High-way
Users_cs	1	9	69	9	52
Users_non	2	5	34	5	24
Users_all	3	14	103	14	76

The *selection stability* of users (i.e., discrepancies in users' answers at the same questions) was the sixth issue. The *selection stability* was determined by setting the 13th and the 14th scenario to be replicas 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 users from *Users_all* set) selected the same function for both of the two similar scenarios. The rest of the users selected the same function for only one out of the two repeated scenarios.

Summary. Overall, the findings indicate that the most preferred distance function is the $\delta_{LCA,P}$, which is expressed with respect to the shortest path of a hierarchy dimension. A null hypothesis stating that the fact that 40.47% of the times $\delta_{LCA,P}$ was chosen as a winner is due to a random phenomenon, has a *p*-value of 6.6×10^{-5}. Apart from the $\delta_{LCA,P}$, the distance functions δ_{Anc} and $\delta_{H,Desc}$ were also popular with the users. In addition, the most preferred distance function family is the *Hierarchy Path* family.

B. User Study for Distances between two Cubes

In the previous user study, the overall observation was that the users prefer the $\delta_{LCA,P}$ distance function between two values of the same dimension. Based on this result, and also by setting the *weighted sum* function as the distance function between cells, we set up the second user study in order to examine which distance function between two cubes is preferred by the users. Specifically, we try to find out which distance function among the two functions that include the *cell mapping* method (section II-C-2) is most closely related to the human perception. These two distance functions are namely the *closest relative* and the *Hausdorff* distance function. Table VI shows the distance functions that were used in this user study.

The user study contained 14 new scenarios. Each scenario included 4 cubes named *A*, *B*, *C* and *D*. The cube *A* in every scenario was the reference cube. The users were asked to order the rest of the three cubes from the most similar to the less similar when compared to the cube *A*. The cubes *B*, *C* and *D* were chosen such that one of them was the closest to the cube *A* according to the *closest relative* function and another was the closest to cube *A* according to the *Hausdorff* distance function. The remaining cube was chosen to be the most distant from cube *A* for both distance functions. Due to space limitations all the scenarios used for this user study can be found in [6].

All scenarios were uploaded as jpeg pictures in an html page where users were asked to complete an answer sheet and send it back to us via email. The URL of this page was sent to the email-list of the graduate students of the Computer Science Department of the University of Ioannina.

TABLE VI
THE DISTANCE FUNCTIONS THAT ARE USED IN THE SECOND USER STUDY

between two cubes	*Hausdorff*
	Closest relative
between two cells of cubes	*weighted sum*
between two values of a dimension	$\delta_{LCA,P}$
between two measures	*Manhattan*

In order to test a user's answer reliability, in the 6th scenario, the cube *B* was identical with the cube *A*. Moreover, in order to measure the users' stability, the 13th and 14th scenarios were replicas of the 5th and 9th scenarios respectively with a reordering on the columns of the cubes.

TABLE VII
FREQUENCY OF CHOSEN AS FIRST DISTANCE FUNCTION AMONG ALL THE ANSWERS

	Frequency	**Percentage**
Hausdorff	154	38%
Closest relative	232	57%
Most distant cube	21	5%

The 12 first scenarios can be divided into three groups according to the weights in the distance function between cells. The first 4 scenarios consist of cubes that do not include measures. We refer to this group as the *no_measures* group. The next 4 scenarios consist of cubes that include measures where the weight factors on measures and dimensions in the function *between cells* are not equal. Specifically, assuming that cubes consist of *k* dimensions and *l* measures, the weight factors were set to *k*/(*l*+*k*) for the dimensions and *l*/(*l*+*k*) for the measures. We refer to this group as the *not_equal* group. Finally, the last four scenarios consist of cubes that include measures and the weight factors on the measures and on the dimensions in the *between cells* distance function are equal and set to 0.5. We refer to this group as the *equal* group.

TABLE VIII
USER STABILITY

	User_OK		**User_Half_OK**		**User_Stable**	
scenario	**Freq.**	**Perc.**	**Freq.**	**Perc.**	**Freq.**	**Perc.**
13th	28	75%	5	13%	24	65%
14th	19	51%	8	21%	24	65%

The number of users that responded with an answer sheet was 39. Two of the 39 users did not choose the cube *B* in the sixth scenario as the most similar to the cube *A*. For that reason their answers were not taken into consideration. We refer to the remaining 37 users as *valid_users*.

The first finding of this user study concerns the most *frequent distance function* that was chosen from the users as their first choice. Among all the 11 (scenarios) * 37 (users) = 407 answers (the sixth scenario is excluded), 232 times ($\approx$57%) the users gave as their first choice the cube that represents the *closest relative* distance function. The cube that represents the *Hausdorff* distance function was chosen 154 times ($\approx$38%) as the first choice of the users. Only 21 times ($\approx$5%) the users chose the most distant cube as their first choice. The summarization of the above results is shown in the Table VII.

The second finding of the user study concerns the stability of the user choices. As we mentioned before, the 13th and 14th scenario were replicas of the 5th and 9th scenario respectively. In each of these two scenarios a user that orders the cubes in the same way as in the original scenario is denoted as *user_OK*. A user that gave the same answer for the most similar cube but the order of the other cubes was not the same is denoted as *user_Half_OK*. Finally, a user that was denoted as *user_OK* for both replicas scenarios or denoted as *user_OK* for the one replica scenario and *user_Half_OK* for the other replica scenario is denoted as *user_ Stable*. According to the answers of the valid 37 users of this user study, in the 13th scenario there were 28 *user_OK* users and 5 *user_Half_OK* users. In the 14th scenario there were 19 *user_OK* users and 8 *user_Half_OK* users. The 24 of the 37 ($\approx$65%) users were *user_Stable* users. We believe that a 65% is a safe number that can ensure the stability and reliability of their answers. The Table VIII summarizes the above results and percentages.

TABLE IX
THE WINNING FUNCTIONS AND THE WINNER FUNCTIONS

Scenario Group	**Scenario**	**Winner function per scenario**		**Group Winner**
no_measures	Scen.1	*Closest relative*	29/37	*Closest relative*
	Scen.2	*Closest relative*	30/37	
	Scen.3	*Closest relative*	31/37	
	Scen.4	*Hausdorff*	25/37	
not_equal	Scen.5	*Hausdorff*	28/37	*Hausdorff*
	Scen.7	*Closest relative*	26/37	
	Scen.8	*Hausdorff*	27/37	
equal	Scen.9	*Hausdorff*	19/37	-
	Scen.10	*Hausdorff*	21/37	
	Scen.11	*Closest relative*	32/37	
	Scen.12	*Closest relative*	22/37	

The third observation concerns the *scenario winner function*. The term *scenario winner function* refers to the function that was mostly selected as the first choice from the users in a specific scenario. Our findings cannot ensure that one of the two functions is more preferred than the other: The *closest relative* function was the scenario winner function for 6 scenarios and the *Hausdorff* function was the scenario winner function for the rest 5 scenarios (Table IX). Observe that the findings of Table VII give a 19% difference between the two prevailing functions --a finding that is not demonstrated in Table IX. This is explained by the fact that when the *closest relative* function is a winner, it wins with an overwhelming majority; on the contrary, when the *Hausdorff* function is a winner, the numbers are lower. The 4th column in Table IX shows how many times the winner function was chosen as a first choice among the 37 valid users.

The fourth observation concerns the (*scenario*) *group winner function* (Table IX). For a group of scenarios, its *group winner* is the function that appeared as *scenario winner* in the majority of the scenarios of the group. For the *no_measures* group the *group winner function* was the *closest relative* function, as it was the *winner function* for the 3 out of the 4 scenarios. For the *not_equal* group the *group winner function* was the Hausdorff, as it was the *winner function* for the 2 out of the 3 scenarios. Finally, for the group *equal*, we have a draw: in two scenarios the winner function was the *closest relative* function and in two scenarios the winner function was the *Hausdorff* function. The above results reveal a user preference in the *closest relative* function for scenarios that do not include measures. On the other hand for the other types of scenarios the results are not clear.

C. Reliability and Validity Considerations

Test Reliability. A possible threat to the test's reliability is the inability of users to understand what was asked from them to perform, or did not handle the test with seriousness and mental concentration. In the 1st user study, the users took the experiment in our presence so we can ensure there were no ambiguous situations or possible misunderstandings. In the 2nd user study, users completed the questionnaire via the web. However, there was a clear description of the setting of the experiment along with an example, so we believe there were not any misunderstandings of what the users should answer. Moreover, we excluded users that failed giving the straightforward answer (in scenario 6 of the 2nd experiment). Finally, in both user studies, we tested the stability of users via replica scenarios.

Test Validity. Possible threats to tests' external validity are the size and the mix of the corpus of users. Naturally, the size of users can always be increased; however we deem that the corpuses we have used are not negligible. Concerning the mix of users, in the 1st experiment we choose to include a group of users with a diversity of backgrounds as well as a clearly distinct group of users with background of computer science (and thus, higher affinity to the notion of comparing two data cubes). An interesting observation is the fact that there are differences of opinions between the *Users_cs* and *Users_non* (Table III and Table V), however these are small and do not change the overall ranking of the preferred functions. Thus, we were able to proceed to a web-based questionnaire in the 2nd study. In addition, the possible scenarios were selected in a way that includes a variety of data types (arithmetic, categorical) and various levels of granularity over the data.

V. Conclusions

This paper presented a 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 a dimension of a multidimensional space, (b) between two points of a multidimensional space and (c) between two sets of points of a multidimensional space.

In order to assess which distance functions are more close to human perception, we conducted two user study analysis. The first user study analysis was conducted in order to discover, which distance function between two values of a dimension is best with regard to the user needs. Our findings indicate that the distance function $\delta_{\mathrm{LCA,P}}$, which is expressed as the length of the path between two values and their common ancestor in the dimension's hierarchy was the most preferred by users in our experiments. Two more functions were widely chosen by users. These were the highway functions δ_{Anc} that is expressed with regard to the ancestor x_y and $\delta_{\mathrm{H,Desc}}$ that is expressed by selecting the representative from a descendant.

The second user study we conducted, took into account the results of the first user study analysis. Specifically, the second user study analysis aimed in discovering which distance function (the *closest relative* or the *Hausdorff* distance function) from the category of distance function between two data cubes, users prefer. Overall, the former function was preferred by the users than the latter; however the individual scores of the tests indicate that this advantage is rather narrow.

References

[1] B. C. M. Fung, K. Wang, and P. S. Yu, "Top-Down Specialization for Information and Privacy Preservation", in proc. of the 21st *IEEE International Conference on Data Engineering* (*ICDE* 2005), 2005, pages 205-216. See also http://ddm.cs.sfu.ca.

[2] 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 conj. with *CIKM* 2009), 2009, pages 81-88.

[3] D. P. Huttenlocher, G. A. Klanderman, W. J. Rucklidge, "Comparing images using the hausdorff distance", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, num. 9, pp. 850-863, September 1993.

[4] Cliff Joslyn, "Poset Ontologies and Concept Lattices as Semantic Hierarchies", in Proc. of the 12th *International Conference on Conceptual Structures* (*ICCS* 2004), 2004, pages 287-302.

[5] 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*, vol. 15, num. 4, pp. 871-882, July/August 2003.

[6] G. Rogkakos, "Similarity Measures for Multidimensional Data," MSc thesis, Univ. of Ioannina, Ioannina, Greece, July. 2010. Available at this paper's web page: http://www.cs.uoi.gr/~ebaikou/publications/2011_ICDE/ that includes questionnaires and findings, too.

[7] P. Sanders and D. Schultes, "Highway Hierarchies Hasten Exact Shortest PathQueries", in proc. of the 13th *Annual European Symposium* (*ESA* 2005), 2005, pages 568-579.

[8] S. Santini and R. Jain, "*Similarity matching*", in 2nd *Asian Conference on Computer Vision* (*ACCV*), 1995, pages 571–580.

[9] S. Santini and R. Jain. "Similarity measures", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, num. 9, pp.871–883, January 1999.

[10] S. Sarawagi, "Explaining differences in multidimensional aggregates", in proc. of the 25th *International Conference on Very Large Data Bases* (*VLDB*), 1999, pages 42–53.

[11] S. Sarawagi, "idiff: Informative summarization of differences in multidimensional aggregates". *Data Min. Knowl. Discov.*, vol. 5, num. 4, pp.255–276, 2001.

[12] P. Vassiliadis, S. Skiadopoulos, "Modelling and Optimisation Issues for Multidimensional Databases", in proc of the 2nd International Conference CAiSE, 2000, pages 482-497.

[13] P. Zezula, G. Amato, V. Dohnal and M. Batko, *Similarity Search: The Metric Space Approach*. ser. Advances in Database Systems. Springer, 2006, vol. 32.