Efficient Estimation of Joint Queries from Multiple OLAP Databases

ELAHEH POURABBAS
National Research Council, Rome, Italy
and
ARIE SHOSHANI
Lawrence Berkeley National Laboratory, Berkeley, California

Given an OLAP query expressed over multiple source OLAP databases, we study the problem of estimating the result OLAP target database. The problem arises when it is not possible to derive the result from a single database. The method we use is linear indirect estimation, commonly used for statistical estimation. We examine two obvious computational methods for computing such a target database, called the "Full cross product" (F) and the "Pre-aggregation" (P) methods. We study the accuracy and computational cost of these methods. While the F method provides a more accurate estimate, it is more expensive computationally than P. Our contribution is in proposing a third new method, called the "Partial Pre-aggregation" method (PP), which is significantly less expensive than F, but is just as accurate. We prove formally that the PP method yields the same results as the F method, and provide analytical and experimental results on the accuracy and computational benefits of the PP method.

Categories and Subject Descriptors: H.2.4[**Database Management**]:Systems-*query processing* [**H.2.7[Database Administration]:Data warehouse and repository**]: H.2.8[**Database Administration**]:Statistical Databases

General Terms: Management, Performance

Additional Key Words and Phrases: OLAP, Query Estimation, Multiple Summary Databases

1. INTRODUCTION

1.1 The Problem

In the 1990's the area of On-Line Analytical Processing (OLAP), which was introduced for the analysis of transactions of enterprise data, has attracted a lot of interest in the research community [Agrawal et al. 1997], [Codd et al. 1993], [Gray

Part of this work was done while E. Pourabbas was visiting Lawrence Berkeley National Laboratory and was supported by a Fulbright Fellowship.
A. Shoshani was supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.
Authors' address: E. Pourabbas, Institute of Systems Analysis and Computer Science "Antonio Ruberti", National Research Council, Viale Manzoni, 30 I-00185 Rome, Italy; email: pourabbas@iasi.cnr.it; A. Shoshani, Lawrence Berkeley National Laboratory, 1 Cyclotron Road Berkeley, CA 94720 USA; email: shoshani@lbl.gov.

et al. 1996], [Gyssens and Lakshmanan 1997], [Lenz and Shoshani 1997]. Similar to Statistical Databases that were introduced in the 1980's [Chan and Shoshani 1981], OLAP databases have a data model that represents one or more "measures" over a multidimensional space of "dimensions", where each dimension can be defined over a hierarchy of "categories" [Codd et al. 1993]. In the OLAP domain, such databases and aggregations over them are often referred to as "data cubes" [Gray et al. 1996]. Similarities and differences between Statistical and OLAP Databases are discussed in [Shoshani 1997].

In many socio-economic applications only summarized data or aggregated data are available because the base data for the summaries (often referred to as "micro-data") are not kept or are unavailable for reasons of privacy. For example, the Census Bureau is required by law to protect information about individuals, and therefore releases only summary data. Similarly, patient data in hospitals are confidential, but the summary data from hospitals are extremely valuable to health authorities. Another reason for keeping only summary data is to improve performance of OLAP databases rather than computing over the base data. The evaluation of queries is performed over the summary data to achieve quick answers rather than on the base data.

We will refer to Statistical Databases or OLAP Databases that contain summarized data as "summary databases", and the measures associated with them as "summary measures" over the dimensions. Each summary measure must have a "summary operator" associated with it, such as "sum", or "average". For example, in the summary database of population by state, race, and age, "population" is the measure, the summary operator is "sum", and "state", "race", and "age" are the dimensions. In this paper, we address the problem of estimating queries expressed over multiple summary databases. That is, given that the base data is not available and that a query cannot be derived from a single summary database, we examine the process of estimating the desired result from multiple summary databases by a method of interpolation common in statistical estimation, called "linear indirect estimation". Essentially, this method takes advantage of the fact that the summary databases were derived from the same base data, and therefore are correlated. For example, suppose that we have a summary database of "total-income by age, education-level, and sex" and another summary database of "population by state, age, race, and sex". If we know that there is a correlation between "population" and "total-income" of states, we can infer the result "total-income by state" even though state is not one of the dimensions in the "total-income" database. We say, in this case, that "population" was used as a *proxy measure* to estimate "total-income by state". The problem we are addressing is how to efficiently answer joint queries over such summary databases.

1.2 Results

Given two source summary databases that were generated from the same base data, each having a summary measure over a set of dimensions, the linear indirect estimation method is used to generate a target database. Typically, the requested summary measure from one database, which we refer to as the primary database measure, is applied over a subset of the dimensions from the second database, which we refer to as the "proxy" database. The resulting "target" database will

have therefore a "target measure" that is the same as the "primary" database, and "target dimensions" that exist in the proxy database, and possibly from the primary database as well. The estimation is achieved by first calculating the target measure over the full cross product of the dimensions from both databases using proportional estimation, and then aggregating over all the non-target dimensions, i.e., the dimensions that are not requested in the result (we give a detailed example in Section 2).

The cost of generating the full cross product can be prohibitive for large databases, and therefore it is a common practice to aggregate over all the non-target dimensions of both databases first (i.e., before generating the full cross product), and only then generate the cross product using proportional estimation to generate the result. However, this method, which we call the "pre-aggregation" method (P), while computationally efficient, yields results that are not as accurate as the "full cross product" method. In Section 2.3, we describe the method for calculating the accuracy of the result relative to the precise result derived from the base data.

We have observed that the summary databases used to generate the estimated results typically have some dimensions in common. For example, in the databases mentioned above: "total-income by age, education-level, and sex" and "population by state, age, race, and sex", "age" and "sex" are in common. This is shown schematically in Figure 1 where we use the X-node notation introduced in [Chan and Shoshani 1981] to represent the cross product of the dimensions below it, and the summary measure above it. We conjectured that we can get the same accuracy as the "full cross product" method by pre-aggregating only the non-common dimensions. For example, if the target database is total-income-by-state, we can aggregate first over "education-level" in the total-income database, and over "race" in the "population" database. Then, we can form the cross product of the resulting databases, and finally aggregate over the non-target common dimensions "sex" and "age" of this cross-product. It turns out that our conjecture was correct, and this produces precisely the same result as the "full cross product" method. We call this method the "partial pre-aggregation" method. This method saves unnecessary computations for obtaining the same accuracy as the full cross product method, and can be many-fold more efficient depending on the number and the cardinalities of non-common dimensions that can be pre-aggregated.

In this paper, we prove formally that the "partial pre-aggregation" (PP) method is as precise as the "full cross product" (F) method. We also prove that the PP method is always less expensive than the F method, provided that there is at least one non-common dimension in each of the source summary databases. Furthermore, we prove that "partial pre-aggregation" can be applied together with operations over category hierarchies of the dimensions. For example, if the state dimension includes the organization of states into regions (i.e., a category hierarchy), and the query requests total-income by region, we could aggregate first within the dimension "state-region" to the "region" level and only then apply the "pre-aggregation" method over the dimensions. This reduces the computational cost. Conversely, we prove that if dis-aggregation from the region to the state level is desired, one can perform the dis-aggregation after the partial pre-aggregation method is applied, again reducing the computational cost. In addition to these results we develop

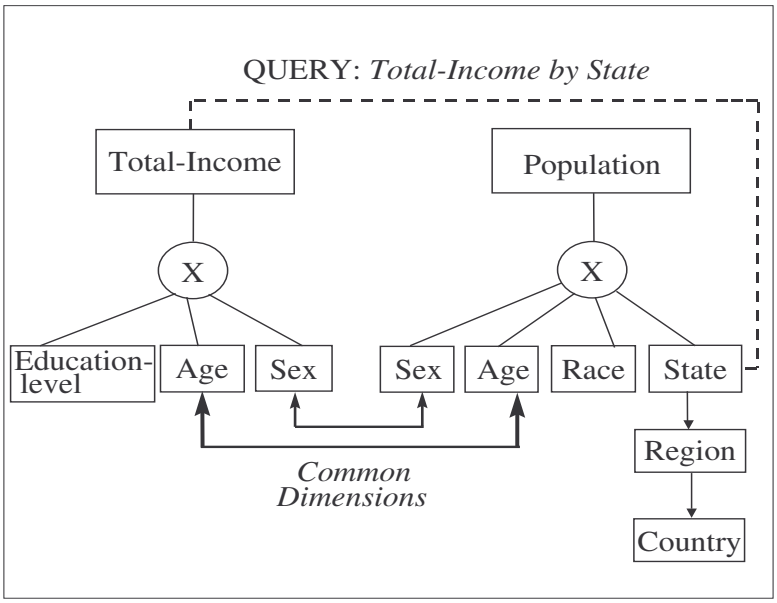

Fig. 1. Illustration of two summary databases, and a joint query

formulas for the computational cost of the three methods: F, P, and PP. Using these formulas and a measure of accuracy, called Average Relative Error, we derive experimental results showing the trade-off between the gain in accuracy versus the increase in computational cost. We show that the gain in accuracy can be very large, especially when the cardinality-product of the dimensions is high, which is usually the case. We also show the large gain in computational cost of the PP method versus the F method when the cardinality-product is large. Finally, we extend the main result of applying the PP method for two databases to the case of three or more databases. We prove that it is possible to pre-aggregate over non-common dimensions without loss in accuracy, provided that these dimensions appear in only one database each (i.e., a dimension that appears in two or more databases is considered a common dimension). Furthermore, we show that under a more strict condition, called the proxy-non-commonality (PNC) condition, applying the proxy databases in any order yields the same result, and therefore the same accuracy.

Incidentally, the cardinality of some dimensions can be very large. Even pre-aggregating over a single dimension with a large cardinality can reduce the computational complexity many fold. Dimensions with large cardinalities exist when items such as type of products, diseases, and chemical or biological species are used as dimensions (often with a classification hierarchy). The development of such hierarchical codes is often done as part of standards activities, such as NAICS, the North American Industry Classification System. There are similar large standardization activities in various business and scientific domains.

1.3 Related Works

The abovementioned problem falls under the general area of answering queries from multiple summary databases. In this area, several studies were published. In [Malvestuto 1993], for instance, the definition of a universal statistical database containing several summary tables which share the same summary measure is examined. Given a query, a system of linear equations over the universal database is constructed whose solutions satisfy the query. A similar approach has been used in [Ng and Ravishankar 1995], where the universal table scheme definition was not used for practical implementation reasons. Instead, the authors consider combinations of the given tables and use a measure for the best quality of the response to the query. Both of these papers assume that the databases have the same summary measure. In this paper, we target the problem of non-homogeneous summary databases, where the summary measures are different.

In [Faloutsos et al. 1997], the estimation of the unknown contents of a summary database (i.e., the values in each cell of a table) when the marginal distributions [1] of the table are given, is addressed. Their approach is based on interpolating the values from the marginal values by enforcing some criterion, such as the smoothness of the distribution of values. In our work, we take a different approach of estimating the values of the target database by using additional information from proxy databases.

In [Buccafurri et al. 2001], the authors studied the problem of estimating range queries over aggregate data using a probabilistic approach for computing the expected value and variance of the answers. The estimation of a range query is based on the knowledge of a compressed representation of the data cube. Specifically, the data cube is partitioned into blocks of possibly different sizes where each block contains a number of aggregate data values. In this work, the authors addressed query estimation without making any assumption on the data distribution. In contrast, we rely on the correlation of data distributions between the summary databases for the estimation of the target results.

In a previously published paper [Pourabbas and Shoshani 2003], we introduced the idea of using partial pre-aggregation which is the subject of this paper, and illustrated its usefulness with examples. In this paper, we prove formally that our proposed method is as accurate as the full cross product method (in the previous paper only a sketch of the proof was provided). We also prove in this paper that the PP method can be applied with aggregation over the category hierarchy (roll-up), and dis-aggregation (drill-down) operators over the category hierarchies in order to reduce computational cost. In addition, we introduce a method for estimating the computation cost, and prove that the PP method is always more efficient than the F method. We show, by way of examples, that the savings in computation costs that our method provides can be many-fold depending on the number of non-common dimensions that can be pre-aggregated. Finally, we expand the methodology from two databases to multiple databases.

The paper is structured as follows. The next section describes the methodology we use to reduce the cost of estimating the joint queries when using a proxy

[1] "Marginals" is a commonly-used term in Statistics that refers to the summary of rows and columns in the "margins" of a table.

database, and introduce the method for evaluating the accuracy. In this section we also introduce the joint query syntax which provides the basis for a formal analysis of the results in this paper. Section 3 provides the formal definitions and examples of the three methods for generating the estimation of the query results. In Section 4 we prove that the PP method gives the same results as the F method. In Section 5 we develop the methodology for applying the PP method together with roll-up and drill-down operations over category hierarchies. In Section 6 we develop formulas for performance evaluation of the three methods, and prove that the PP method is always less expensive computationally than the F method. Section 7 evaluates the accuracy and cost trade-offs. Finally, Section 8 extends the results of using the PP method to three or more databases. Section 9 contains the conclusions.

2. METHODOLOGY AND FORMAL MODEL

2.1 Approach

We focus on a class of queries defined by a single summary measure over multiple summary databases where the result requires aggregation along one or more dimensions. The summary operators that we consider in this paper are COUNT, SUM. However, given that both COUNT and SUM are computed (thus doubling the computational cost), the AVERAGE operator can be supported as well. As stated above, it is assumed that the base data, or micro-data, are not available for privacy reasons or are no longer available when the multiple databases are queried.

We assume that common dimensions in the summary databases have the same domain values and each dimension has at least two possible values. Range queries may specify a subset of the values. For example, the query might ask for results only for "Black" and "Hispanic" for the race dimension that has seven race values. For the purpose of evaluating computational costs, we consider only queries that include all possible values for each dimension of interest. Thus, our joint query syntax does not include the ability to specify a subset of dimension values. However, we note that it is possible to restrict the computation for subsets of values by simply calculating only the combination of values of interest. For example, for the joint query to estimate population by state and race, where state is restricted to "Alabama" and "Georgia", and race is restricted to "Black" and "Hispanic", only the corresponding four cells have to be calculated.

To motivate our work, let us reconsider the summary databases mentioned in the previous section and shown in Figure 1. Note that from now on we use the shorter term "Education" instead of "Education-level". One database represents *Total-Income* by *Education*, *Age*, and *Sex*, and the second database represents *Population* by *State*, *Age*, *Race*, and *Sex*. All the dimensions have a single category hierarchy level except one of the dimensions that has three levels of categories: *State* → *Region* → *Country*. The query over the two summary databases: find the "*Total-Income* by *State*", is represented with the broken line in Figure 1. Similar queries at a higher level of the category hierarchy are "*Total-Income* by *Region*", and "*Total-Income* by *Country*". Note that "State" is not a dimension in the Total-Income summary database, and thus it is not possible to derive "*Total-Income* by *State*" from that summary database alone.

Consider the databases mentioned in Figure 1 written in the following nota-

tion: "summary-measure (dimension,...,dimension)". Using this notation the two databases are: *Total-Income(Age,Education,Sex)* and *Population(State,Age,Race, Sex)*. Our goal is to derive *Total-Income(State)*.

One obvious method of estimating this result is to aggregate each of the source summary databases to the maximum level. We call this the Pre-aggregation (P) method. In this case, we can aggregate *Population(State,Age,Race,Sex)* over *Age*, *Race*, *Sex* to produce *Population(State)*[2] and *Total-Income(Age,Education,Sex)* over *Age*, *Education*, and *Sex* to produce *Total-Income(•)*, where the symbol "•" indicates aggregation over all the dimensions. Then, we can calculate the proportional estimated values using linear indirect estimation (see Section 2.2) to produce *Total-Income(State)*. Another possibility is to produce the full cross product: *Total-Income(State,Age,Education,Race,Sex)* using *Population* as a proxy summary measure. Then, from this result we can aggregate over *Age*, *Education*, *Race* and *Sex* to get *Total-Income(State)*. We call this the Full cross product (F) method. This method is based on the well-known "small area estimation" methodology according to which the most accurate result that can be obtained using linear indirect estimation is when the solution is based on the largest number of cells that can be generated from the source summary databases.

The proposed Partial Pre-aggregation (PP) method achieves the same accuracy as the F method but at a much lower computational cost. This is achieved by noticing that it is possible to pre-aggregate over all the dimensions that are not in common to the two source databases before performing the cross product, and still achieve the same accuracy of the full cross product computation. According to this method, in our example, *Sex* and *Age* are in common to the two databases. Thus, we first aggregate over *Education* in the *Total-Income(Age,Education,Sex)* database to produce *Total-Income(Age,Sex)*, and over *Race* in *Population(State,Age,Race,Sex)* database to produce *Population(State,Age,Sex)*. Then, we use the linear indirect estimation to produce *Total-Income(State,Age,Sex)*. Note that we have *State* in *Total-Income(State,Age,Sex)* in addition to *Sex*, *Age*, because it is the dimension we want in our result. Finally, we aggregate over *Age*, and *Sex* in *Total-Income(State,Age,Sex)* to produce *Total-Income(State)*. We show that this result is as accurate as the result obtained by using the F method, but by performing the aggregations over the non-common dimensions first we reduce the computation needed. We extend these results to the case of the dimensions that are defined over category hierarchies. We also extend these results to the case of multiple source summary databases.

2.2 The Linear Indirect Estimation Method

Our methodology of estimating the result of a joint query is based on the linear indirect estimation method, known in the literature as *Small Area Estimation (SAE)* [Rao 2003]. There is great interest in the *SAE* method because of the tendency in many countries to base future censuses on administrative record systems [Chand and Alexander 1996]. This technique is quite popular, not only from a theoretical point of view [Ghosh and Rao 1994], but it is used in practice in commercial prod-

[2] *Population(State)* is equivalent to *Population(State,ALL,ALL,ALL)*, where ALL indicates the construct introduced in [Gray et al. 1996]. For the sake of brevity, we use the first notation.

ucts, such as RTI international [RTI]. The main idea of such an approach is to use data from surveys of variables of interest at the national or regional level, and to obtain estimates at more geographically disaggregated levels such as counties or other small areas. This approach is characterized by indirect estimation techniques, and is used in many domains (e.g., socio-economic area [Pfeffermann 2002], and health area [Elliott et al. 1996]).

An indirect estimation calculates values of the variable of interest using available auxiliary (called *predictor* or *proxy*) data at the local level that are correlated with the variable of interest [Ghosh and Rao 1994], [Pfeffermann 2002], [Schaible 1996]. For example, suppose that we have *Total-Income* at the *Region* level, and *Population* at the *State* level. We can use the population data (at the *State* level) as a proxy for predicting the *Total-Income* at a *State* level. It is assumed that the *Total-Income* and *Population* are correlated, and therefore the distribution of *Total-Income* at the *State* level is proportional to the distribution of *Population* at the *State* level. Specifically, we can use the following proportions for a particular $State_i$ that belongs to $Region_j$:

$$\textit{Total-Income(State}_i\textit{)=Total-Income(Region}_j\textit{)}\ \frac{Population(State_i)}{Population(Region_j)}.$$

The population of $Region_j$ is calculated by summing up the population of all the states in that region. A more generalized notation for all state values is:

$$\textit{Total-Income(State)=Total-Income(Region)}\frac{Population(State)}{\sum_{Region} Population(State)}$$

In the above example, we illustrate how Linear Indirect Estimation is used to generate proportional fractions of the proxy measure. Now, suppose that the databases have multiple dimensions, some of which can be in common between the databases. The approach of "small area" estimation is to estimate the result based on the smallest possible areas by calculating the full cross product of the dimensions, and then aggregating to the desired dimensions. For example, given the databases *Income(Age,Education,Sex)*, and *Population(State,Age,Race,Sex)*, where *Income(State,Age)* is requested, the small area estimation methodology requires the calculation (by linear indirect estimation) of the cross product *Income(State,Age, Education,Sex,Race)* and then aggregating over that to get *Income(State,Age)*. The expression for generating the cross product in the above example can be found in Section 3, Example 2.

Formally, let i denote a small area. A target measure $Y(d)$ is provided over a set of dimensions d. $Y(d)$ was generated from $Y(d) = \sum_i Y(i,d)$. $Y(i,d)$ is no longer available. However, auxiliary information in the form of $X(i,d)$ is available. A linear indirect estimation of Y for small area i is defined by:

$$\hat{Y}(i) = \sum_d \hat{Y}(i,d) = \sum_d Y(d)\frac{X(i,d)}{X(d)}$$

where $X(d) = \sum_i X(i,d)$. $X(i,d)/X(d)$ represents the proportion of the population of small area i relative to the total population over set of dimensions d. Note that in this method, the sum over all estimated values, $\sum_i \hat{Y}(i)$, must be equal to sum over the true value $\sum_d Y(d)$ [Ghosh and Rao 1994]. This condition is illustrated in

the examples in Section 3 (see Table IV).

A more general case is when the set of dimensions for X and Y does not fully overlap. Let d_X and d_Y represent the set of dimensions for X and Y, respectively. The above formula can be generalized, as follows:

$$\hat{Y}(i, d_X \cup d_Y) = Y(d_Y) \frac{X(i, d_X)}{\sum_{i, \left(d_X - (d_X \cap d_Y)\right)} X(i, d_X)} \tag{1}$$

This is the basis for the F method as defined in Definition 3.1 in Section 3.

2.3 Average Relative Error

The estimate is subject to error. For the purpose of evaluating the estimated errors, we assume in our examples, that we know the values from the base data, and can evaluate the error exactly. This provides us with the means of comparing the accuracy of the results using different computational methods.

A method that is commonly used for measuring accuracy is the average relative error (*ARE*) [Ghosh and Rao 1994]. This is defined simply by taking the absolute value of the difference between each estimated value and the corresponding precise value and dividing by the precise values. The fractions are summed and divided by the number of estimated values. Formally,

$$ARE = \frac{1}{m} \sum_{i=1}^{m} \frac{|\hat{v}_i - v_i|}{v_i}. \tag{2}$$

where $\hat{v}_i$ and v_i are, respectively, the estimated and precise (or base data) values, and m is the number of small areas for which estimated values were calculated.

We use the Average Relative Error (ARE) together with computational cost expressions to evaluate the trade-off of cost and accuracy in Section 7.

2.4 The Joint Query Syntax

We define the syntax of a joint query on two summary databases, in terms of the common and non-common dimensions of the databases. In the following sections, we assume two source summary databases, called DB_P and DB_Q that are used to answer joint queries and produce a *target database* DB_T. The databases are defined as follows: $DB_P = M_P(\{A_P^i \quad 0 < i \leq m\})$, $DB_Q = M_Q(\{A_Q^j \quad 0 < j \leq n\})$, and $DB_T = M_T(\{A_T^k \quad 0 < k \leq t\})$, where M_P, M_Q, and M_T are the measures of the corresponding databases, A_P^i, A_Q^j, and A_T^k are the corresponding dimensions, and m, n, and t are the cardinalities of the corresponding dimensions. In expressing a joint query over the two source summary databases, one of the measures, either M_P or M_Q is selected. Without loss of generality, suppose that M_P is selected. Thus, $M_P = M_T$. M_Q is called the *proxy measure*, DB_Q is called the *proxy database*, and DB_P is called the *primary database*.

Given two source summary databases DB_P and DB_Q that are used to generate a target database DB_T, we can classify the source database dimensions as belonging to three disjoint groups: target dimensions, common dimensions, and non-common dimensions. First, we pick the dimensions in the source databases that are specified

in the target database for the target group; then the *remaining* dimensions are considered common if they are in both source databases, and are considered non-common otherwise. Note that a target dimension can exist in both source databases. For example, given $DB_P = M_P(A_1, A_2, A_3, A_4)$, and $DB_Q = M_Q(A_1, A_3, A_5, A_6)$, and $DB_T = M_P(A_1, A_2, A_5)$. A_1, A_2 and A_5 are classified as target dimensions, and therefore are not eligible for the common or non-common groups. A_3 is classified as common, and A_4 and A_6 are classified as non-common. The target dimensions are further classified as common-target if they exist in both source databases, and as non-common-target if they exist in a single source database only. Thus, A_1 is a common-target dimension and A_2, A_5 are non-common-target dimensions.

We use the following notation: $DB_P = M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}})$, and $DB_Q = M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})$, where C, $\overline{C}$, and T refer to the common, non-common, and target dimension-groups, respectively. Note that $A_P^C = A_Q^C$, and $A_P^{T^C} = A_Q^{T^C}$.

We use the notation A_T for the group of target dimensions $\{A_T^k \;\; 0 < k \leq t\}$. Thus, $DB_T = M_T(A_T)$. Using the notation above, we have $A_T = A_P^{T^C} \cup A_P^{T^{\overline{C}}} \cup A_Q^{T^{\overline{C}}}$.

Note that $A_Q^{T^{\overline{C}}}$ must always exist to make the proxy summarization meaningful. However, $A_P^{T^C}$ and $A_P^{T^{\overline{C}}}$ may or may not exist. Indeed, if $A_Q^{T^{\overline{C}}}$ does not exist, then there is no need to use DB_Q, since the results can be obtained from DB_P only.

Example 1 Let us consider the source summary databases mentioned above: *Total-Income(Age,Education,Sex)*, and *Population(State,Age,Race,Sex)*. For the sake of brevity, we use "Income" to mean "Total-Income" in the rest of paper. Let us assume that the joint query expressed over them is *Income(State)*. In this case, *Income(State)* is the target summary database, *Population(State,Age,Race,Sex)* is the proxy database, and *Income(Age,Education,Sex)* is the primary database. $A_T = \{State\}$ is the target dimension, where $A^{T^C}_{Population} = A^{T^C}_{Income} = \emptyset$, $A^{T^{\overline{C}}}_{Population} = \{State\}$, $A^{T^{\overline{C}}}_{Income} = \emptyset$ are the non-common target dimensions, $A^C_{Population} = A^C_{Income} = \{Age, Sex\}$ are the common dimensions between the source summary databases, and $A^{\overline{C}}_{Population} = \{Race\}$, and $A^{\overline{C}}_{Income} = \{Education\}$ are the non-common dimensions. If the joint query expressed over the source databases is *Income(State,Sex)*, then $A_T = \{State, Sex\}$ and accordingly, $A^{T^C}_{Population} = A^{T^C}_{Income} = \{Sex\}$, $A^{T^{\overline{C}}}_{Population} = \{State\}$, $A^{T^{\overline{C}}}_{Income} = \emptyset$, and $A^C_{Population} = A^C_{Income} = \{Age\}$.

3. DEFINITION OF THE AGGREGATION METHODS

We assume that the joint query is formulated on two source summary databases. In order to describe the proposed Partial Pre-aggregation method, we first define the Full cross product, and the Pre-aggregation methods. For the definition of these methods, we use the formalism defined for a joint query in Section 2.4.

3.1 The Full Cross Product (F) Method

The next definition provides the expression for calculating the estimate of the target database using linear indirect estimation for the full cross product F method.

The expression is in terms of the common, non-common, and target groups of dimensions. Note that consistent with the notation used in the literature, we use the symbol "ˆ" over the target measure to indicate that this is an estimated result.

The following definition is based on Eq. 1 in Section 2.2, where the $d_X \cup d_Y$ represents the union of common and non-common dimensions, $d_X \cap d_Y$ represents the common dimensions, and i represent the target dimensions.

DEFINITION 3.1. Let $M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}})$ and $M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})$ be two source summary databases. The *full cross product estimator* of $DB_T = \hat{M}_P(A_P^{T^C}, A_P^{T^{\overline{C}}}, A_Q^{T^{\overline{C}}})$ is the estimator that is computed by applying the linear indirect estimation as follows:

$$\begin{aligned}&\hat{M}_P[\mathsf{F}](A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}}, A_Q^{\overline{C}}, A_Q^{T^{\overline{C}}}) = \\ &M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}}) \frac{M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})}{\sum_{A_Q^{\overline{C}}, A_Q^{T^{\overline{C}}}} M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})}\end{aligned} \tag{3}$$

and then summarizing over the common and non-common dimensions:

$$\hat{M}_P[\mathsf{F}](A_P^{T^C}, A_P^{T^{\overline{C}}}, A_Q^{T^{\overline{C}}}) = \sum_{A_P^C, A_P^{\overline{C}}, A_Q^{\overline{C}}} \hat{M}_P[\mathsf{F}](A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}}, A_Q^{\overline{C}}, A_Q^{T^{\overline{C}}})$$

Example 2 Table I and Table II represent the data in the source summary databases *Income(Age,Education,Sex)*, and *Population(State,Age,Race,Sex)*. Suppose that *Income(State)* is the target summary database. We first obtain the full cross product summary database by Eq. (3):

$$\begin{aligned}&\hat{I}ncome[\mathsf{F}](State, Age, Education, Race, Sex) = \\ &Income(Age, Education, Sex) \frac{Population(State, Age, Race, Sex)}{\sum_{State, Race} Population(State, Age, Race, Sex)}\end{aligned}$$

which is shown in Table III (only one state is shown); then we summarize all dimensions except the target dimension. The result is shown in Table IV, third column. Note that in this example the single target dimension is in the proxy database *Population*.

3.2 The Pre-aggregation (P) Method

The pre-aggregation method is based on summarizing the summary databases over all common and non-common dimensions before the application of the linear indirect estimation method.

DEFINITION 3.2. Let $M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}})$ and $M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})$ be source summary databases. The *pre-aggregation estimator* of $DB_T = \hat{M}_P(A_P^{T^C}, A_P^{T^{\overline{C}}}, A_Q^{T^{\overline{C}}})$ is the estimator that is computed by pre-summarizing all common and non-common dimensions in the source summary databases as follows:

Table I. *Income(Age,Education,Sex)*

Income			*Sex*	
	Age	*Education*	Male	Female
	< 25	High School	105774000	84364500
		Bachelor or higher degree	161874000	126092500
	$25 \div 34$	High School	219398500	192286000
		Bachelor or higher degree	265824500	234622500
	$35 \div 44$	High School	454273000	447521000
		Bachelor or higher degree	564399000	537749000
	$45 \div 54$	High School	1026221000	943504500
		Bachelor or higher degree	1202296500	1144370000
	$55 \div 64$	High School	1073901500	988478000
		Bachelor or higher degree	1146216500	1027928500
	$65 \div 74$	High School	840126000	799786500
		Bachelor or higher degree	975004500	866734000
	≥ 75	High School	742101500	669896500
		Bachelor or higher degree	751386000	664883000

$$M_P(A_P^{T^C}, A_P^{T^{\overline{C}}}) = \sum_{A_P^C, A_P^{\overline{C}}} M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}})$$
$$M_Q(A_Q^{T^C}, A_Q^{T^{\overline{C}}}) = \sum_{A_Q^C, A_Q^{\overline{C}}} M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_P^{T^{\overline{C}}})$$

and then applying the linear indirect estimation:

$$\hat{M}_P[\mathsf{P}](A_P^{T^C}, A_P^{T^{\overline{C}}}, A_Q^{T^{\overline{C}}}) = M_P(A_P^{T^C}, A_P^{T^{\overline{C}}}) \frac{M_Q(A_Q^{T^C}, A_Q^{T^{\overline{C}}})}{\sum_{A_Q^{T^{\overline{C}}}} M_Q(A_Q^{T^C}, A_Q^{T^{\overline{C}}})} \tag{4}$$

Example 3 Consider Table I and Table II. To apply the P method, we first summarize all common and non-common dimensions in the source summary databases *Income* and *Population*:

$$\sum_{Age,Race,Sex} Population(State, Age, Race, Sex) = Population(State)$$
$$\sum_{Age,Education,Sex} Income(Age, Education, Sex) = Income(\bullet)$$

Then, by applying linear indirect estimation, we obtain:

$$\hat{I}ncome[\mathsf{P}](State) = Income(\bullet) \frac{Population(State)}{\sum_{State} Population(State)}$$

Since the only target dimension is in the proxy database *Population*, we sum over all dimensions in the *Income* database. The result of the target summary database $\hat{I}ncome(State)$ is shown in Table IV, fourth column. We observe that $\hat{I}ncome[\mathsf{F}](State)$ and $\hat{I}ncome[P](State)$ are different, and therefore their Average Relative Errors are different. We will address the accuracy issue in a later section by using the expression for calculating the Average Relative Error (ARE) (see Eq. 2).

Table II. *Population(State,Age,Race,Sex)*

Population			*Sex*			*Sex*		
Age	*Race*	*State*	Male	Female	*State*	Male	Female	...
< 25	White	Alabama	290	221	California	802	715	...
	Black		270	204		569	687	...
	Hispanic		241	149		443	467	...
	Other		250	136		390	222	...
$25 \div 34$	White		495	537		802	704	...
	Black		343	343		569	490	...
	Hispanic		332	154		443	375	...
	Other		353	163		290	274	...
$35 \div 44$	White		906	670		1065	945	...
	Black		586	590		880	920	...
	Hispanic		420	296		667	668	...
	Other		344	286		491	445	...
$45 \div 54$	White		1130	969		3190	3490	...
	Black		870	577		2860	3090	...
	Hispanic		646	585		1960	1910	...
	Other		567	498		1579	1789	...
$55 \div 64$	White		739	559		2180	2870	...
	Black		618	530		2399	2390	...
	Hispanic		511	390		1710	1790	...
	Other		440	375		1520	1639	...
$65 \div 74$	White		475	430		1716	1379	...
	Black		527	474		1310	1127	...
	Hispanic		456	360		867	890	...
	Other		335	269		640	715	...
≥ 75	White		497	468		1116	865	...
	Black		429	318		880	612	...
	Hispanic		394	335		765	555	...
	Other		306	239		500	518	...

We show that the accuracy difference between the PP method and the P method can be very significant, and is sensitive to the cardinality of the dimensions: the larger the cardinality, the larger the difference.

3.3 The Partial-Pre-aggregation (PP) Method

This method was devised with the expectation that it will provide the same accuracy as the F method but with a lower computational complexity. As mentioned above, the main idea is to summarize the source summary databases only over non-common dimensions, and then estimate the target summary database with the common and target dimensions.

DEFINITION 3.3. Let $M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}})$ and $M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})$ be source summary databases. The *partial pre-aggregation estimator* of $DB_T = \hat{M}_P(A_P^{T^C}, A_P^{T^{\overline{C}}}, A_Q^{T^{\overline{C}}})$ is the estimator that is computed by pre-summarizing over all the non-common dimensions in the source databases as follows:

Table III. $\hat{I}ncome[F](State, Age, Education, Race, Sex)$

$\hat{I}ncome$				Sex	
State	*Age*	*Education*	*Race*	Male	Female
Alabama	< 25	High school	White	2080894.105	1366602.250
			Black	1937384.167	1261479.000
			Hispanic	1729294.756	921374.3678
			Other	1793874.228	840986.0001
		Bachelor or higher degree	White	3184550.573	2042545.078
			Black	2964926.396	1885426.226
			Hispanic	2646471.338	1377100.528
			Other	2745302.218	1256950.817
	.....	.....	.....	.....	.....
	.....	.....	.....	.....	.....
	.....	.....	.....	.....	.....
	≥ 75	High school	White	19478449.720	17832407.830
			Black	16813390.200	12116892.500
			Hispanic	15441668.390	12764650.900
			Other	11992767.840	9106721.091
		Bachelor or higher degree	White	19722146.400	17698950.230
			Black	17023744.070	12026209.770
			Hispanic	15634860.520	12669120.360
			Other	12142810.460	9038566.464
	Total			696964726.800	540632187.100
.....	.....	.....	.....	.....	.....

Table IV. $\hat{I}ncome(State)$ by methods F (or PP) and P, and their ARE

	Income=I	$\hat{I}([F]/[PP])$	$\hat{I}[P]$	$\frac{\lvert\hat{I}_i[F]-I_i\rvert}{I_i}$	$\frac{\lvert\hat{I}_i[P]-I_i\rvert}{I_i}$
Alabama (AL)	958808000	1237596914	1238898065	0.290766153	0.292123204
California (CA)	3925821500	3357547451	3241886947	0.144752901	0.174214379
Florida (FL)	2070083000	2346319696	2227279446	0.133442328	0,075937267
Nevada (NV)	825903000	846220715	862525859	0.024600607	0.044342809
Missouri (MO)	640774500	873014896	1012815964	0.362437013	0.580612156
New Jersey (NJ)	1091973500	851977231	852821696	0.219782136	0.219008798
Texas (TX)	5026737500	4709126342	4707215468	0.063184353	0.063564495
Virginia (VA)	2030547000	2319814110	2398172234	0.142457727	0.181047389
Washington (WA)	1686365000	1715395645	1715397321	0.017214924	0.017215918
Total	18257013000	18257013000	18257013000		
ARE				0.155404238	0.183118491

$$M_P(A_P^C, A_P^{T^C}, A_P^{T^{\overline{C}}}) = \sum_{A_P^{\overline{C}}} M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}})$$
$$M_Q(A_Q^C, A_Q^{T^C}, A_Q^{T^{\overline{C}}}) = \sum_{A_Q^{\overline{C}}} M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})$$

then estimating the cross product:

Table V. $\hat{I}ncome$[PP](State,Age,Sex)

$\hat{I}ncome$		Sex		
State	*Age*	Male	Female	
Alabama	< 25	19082697.78	10952464.27	
	$25 \div 34$	42617914.01	30419041.28	
	$35 \div 44$	86519239.21	65229031.38	
	$45 \div 54$	159694598.8	123830217.71	
	$55 \div 64$	137391938.4	105596069.57	
	$65 \div 74$	123408500.9	101351843.79	
	≥ 75	128249837.6	103253519.1	
	Total	696964726.8	540632187.1	
.....	.....	.....	.....	.....

$$\hat{M}_P[\mathsf{PP}](A_P^C, A_P^{T^C}, A_P^{T^{\overline{C}}}, A_Q^{T^{\overline{C}}}) = M_P(A_P^C, A_P^{T^C}, A_P^{T^{\overline{C}}}) \frac{M_Q(A_Q^C, A_Q^{T^C}, A_Q^{T^{\overline{C}}})}{\sum_{A_Q^{T^{\overline{C}}}} M_Q(A_Q^C, A_Q^{T^C}, A_Q^{T^{\overline{C}}})}$$

and finally summarizing over the common dimensions as follows:

$$\hat{M}_P[\mathsf{PP}](A_P^{T^C}, A_P^{T^{\overline{C}}}, A_Q^{T^{\overline{C}}}) = \textstyle\sum_{A_P^C} \hat{M}_P[\mathsf{PP}](A_P^C, A_P^{T^C}, A_P^{T^{\overline{C}}}, A_Q^{T^{\overline{C}}}) \tag{5}$$

Example 4 Consider again Table I and Table II. First, we summarize the source summary databases over non-common dimensions as follows:

$$\textstyle\sum_{Education} Income(Age, Education, Sex) = Income(Age, Sex)$$
$$\textstyle\sum_{Race} Population(State, Age, Race, Sex) = Population(State, Age, Sex)$$

Then, by applying the linear indirect estimation, we obtain

$$\begin{aligned} \hat{I}ncome[\mathsf{PP}](State) &= \textstyle\sum_{Age,Sex} \hat{I}ncome[\mathsf{PP}](State, Age, Sex) \\ &= \textstyle\sum_{Age,Sex} \left(Income(Age, Sex) \frac{Population(State, Age, Sex)}{\sum_{State} Population(State, Age, Sex)} \right) \end{aligned}$$

In Table V, the result of the cross product for one state is illustrated. This was used to generate $\hat{I}ncome[\mathsf{PP}](State)$. We note that the results obtained by $\hat{I}ncome[\mathsf{PP}](State)$ were identical to $\hat{I}ncome[\mathsf{F}](State)$ shown in Table IV, and therefore are shown in the same (third) column.

In Table VI, an example is shown with a common target dimension over source summary databases in Table I and Table II. In this case, the target summary database is *Income(State,Sex)*, and the estimation by the PP method shown in the fourth column is obtained by summarizing over the common dimension as follows:

$$\begin{aligned} \hat{I}ncome[\mathsf{PP}](State, Sex) &= \textstyle\sum_{Age} \hat{I}ncome[\mathsf{PP}](State, Age, Sex) \\ &= \textstyle\sum_{Age} \left(Income(Age, Sex) \frac{Population(State, Age, Sex)}{\sum_{State} Population(State, Age, Sex)} \right) \end{aligned}$$

Table VI. $\hat{I}ncome(State, Sex)$ by methods F (or PP) and P, and ARE

State	*Sex*	*Income=I*	$\hat{I}([F]/[PP])$	$\hat{I}[P]$	$\frac{\lvert\hat{I}_i[F]-I_i\rvert}{I_i}$	$\frac{\lvert\hat{I}_i[P]-I_i\rvert}{I_i}$
Alabama	Male	544850500	696964727	705127459	0.279185257	0.294166857
	Female	413957500	540632187	537115816	0.306008919	0.297514397
California	Male	2018050500	1728559519	1669518558	0.143450811	0.172707245
	Female	1907771000	1628987932	1571081867	0.146130258	0.176482992
Florida	Male	1154800500	1300477704	1237122315	0.126149239	0.071286612
	Female	915282500	1045841991	994424671	0.142643928	0.086467479
Nevada	Male	430034000	441392745	446580724	0.026413598	0.038477711
	Female	395869000	404827970	415739712	0.022631148	0.050195170
Missouri	Male	328957000	444459592	523955277	0.351117599	0.592777404
	Female	311817500	428555304	488594342	0.374378615	0.566924057
New Jersey	Male	552702500	430903383	440179785	0.220370122	0.203586405
	Female	539271000	421073849	412360106	0.219179505	0.235337879
Texas	Male	2582131500	2417157398	2397791813	0.063890666	0.071390511
	Female	2444606000	2291968943	2306049423	0.062438306	0.056678490
Virginia	Male	1043676000	1183334254	1222374551	0.133813802	0.171220332
	Female	986871000	1136479857	1174123104	0.151599203	0.189743243
Washington	Male	873594000	885547178	886146019	0.013682761	0.014368252
	Female	812771000	829848467	828727459	0.021011413	0.019632171
Total		18257013000	18257013000	18257013000		
ARE					0.155783064	0.183830956

The estimation using the P method, shown in the fifth column, is obtained by summarizing first over all common and non-common dimensions as follows:

$$\sum_{Age,Race} Population(State, Age, Race, Sex) = Population(State, Sex)$$
$$\sum_{Age,Education} Income(Age, Education, Sex) = Income(Sex)$$

and then by applying the linear indirect estimation:

$$\hat{I}ncome[\mathsf{P}](State, Sex) = Income(Sex)\frac{Population(State, Sex)}{\sum_{State} Population(State, Sex)}$$

4. THE EQUIVALENCE OF PP AND F METHODS FOR TWO SOURCE SUMMARY DATABASES

As discussed in Section 2.2, the indirect linear estimation over the full cross product of the dimensions provides the most accurate estimated result. It follows that other methods, such as the pre-aggregation P method may be statistically less accurate. In this section, we prove that the partial pre-aggregation over dimensions in the PP method gives precisely the same result as the F method. Thus, we need only to analyze the accuracy of the P method relative to the PP method, since the PP method produces the same accuracy as the F method.

In the following theorem we use the definitions for methods F and PP introduced in the previous section.

THEOREM 4.1. *The estimation of any joint query* $\hat{M}_P(A_P^{T^C}, A_P^{T^{\overline{C}}}, A_Q^{T^{\overline{C}}})$ *over* $M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}})$ *and* $M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})$ *using the methods F and PP give the same results.*

PROOF. We show F⇔PP as follows:

$$
\begin{aligned}
&\hat{M}_P[\mathsf{F}](A_P^{T^C}, A_P^{T^{\overline{C}}}, A_Q^{T^{\overline{C}}}) \\
&= \textstyle\sum_{A_Q^C, A_P^{\overline{C}}, A_Q^{\overline{C}}} \left(\hat{M}_P[\mathsf{F}](A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}}, A_Q^C, A_Q^{\overline{C}}, A_Q^{T^{\overline{C}}}) \right) \\
&= \textstyle\sum_{A_Q^C, A_P^{\overline{C}}, A_Q^{\overline{C}}} \left(M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}}) \frac{M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})}{\sum_{A_Q^{\overline{C}}, A_Q^{T^{\overline{C}}}} M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})} \right) \\
&= \textstyle\sum_{A_Q^C} \left(\sum_{A_P^{\overline{C}}, A_Q^{\overline{C}}} \left(M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}}) \frac{M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})}{\sum_{A_Q^{\overline{C}}, A_Q^{T^{\overline{C}}}} M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})} \right) \right) \\
&= \textstyle\sum_{A_Q^C} \left(\left(\sum_{A_P^{\overline{C}}} M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}}) \right) \left(\frac{\sum_{A_Q^{\overline{C}}} M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})}{\sum_{A_Q^{\overline{C}}, A_Q^{T^{\overline{C}}}} M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})} \right) \right) \\
&= \textstyle\sum_{A_Q^C} \left(\left(\sum_{A_P^{\overline{C}}} M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}}) \right) \left(\frac{\sum_{A_Q^{\overline{C}}} M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})}{\sum_{A_Q^{T^{\overline{C}}}} \left(\sum_{A_Q^{\overline{C}}} M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}}) \right)} \right) \right) \\
&= \textstyle\sum_{A_Q^C} \left(\left(\sum_{A_P^{\overline{C}}} M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}}) \right) \left(\frac{\sum_{A_Q^{\overline{C}}} M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})}{\sum_{A_Q^{T^{\overline{C}}}} \left(M_Q(A_Q^C, A_Q^{T^C}, A_Q^{T^{\overline{C}}}) \right)} \right) \right) \\
&= \textstyle\sum_{A_Q^C} \left(M_P(A_P^C, A_P^{T^C}, A_P^{T^{\overline{C}}}) \frac{M_Q(A_Q^C, A_Q^{T^C}, A_Q^{T^{\overline{C}}})}{\sum_{A_Q^{T^{\overline{C}}}} M_Q(A_Q^C, A_Q^{T^C}, A_Q^{T^{\overline{C}}})} \right) \\
&= \textstyle\sum_{A_P^C} \hat{M}_P[\mathsf{PP}] \left(A_P^C, A_P^{T^C}, A_P^{T^{\overline{C}}}, A_Q^{T^{\overline{C}}} \right)
\end{aligned}
$$

The last term is the same as $\hat{M}_P[\mathsf{PP}](A_P^{T^C}, A_P^{T^{\overline{C}}}, A_Q^{T^{\overline{C}}})$ (see Eq. (5)). □

5. APPLYING THE PP METHOD WITH OPERATIONS OVER CATEGORY HIERARCHIES

Another situation where partial pre-aggregation might apply is when multiple categories of a dimension are involved in a joint query. For example, suppose that the source databases are: *Population(Race,Age)*, and *Income(State,Race,Sex)*, and the desired target database is: *Population(Region,Race)* where the category relationship between *State* → *Region* is known. One possibility to evaluate this query is to first apply the PP method to estimate $\hat{P}$*opulation(State,Race)* by pre-aggregating over the non-common dimensions *Age* and *Sex*, and then aggregating $\hat{P}$*opulation(State,Race)* from the *State* to the *Region* level to obtain $\hat{P}$*opulation(Region, Race)*. The aggregation over *State* to the *Region* level is referred to as "roll-up".

Another possibility is to roll-up *Income(State,Race,Sex)* to generate *Income(Region, Race,Sex)* before applying the PP method. The question is whether the two evaluation orders produce results with the same accuracy. If this is the case, then it is possible to perform roll-up and pre-aggregation operations together, thus eliminating the need for intermediate results and consequently reducing the computational

cost. We prove in Section 5.1 that the same accuracy is achieved regardless of the order of applying pre-aggregation and roll-up operations.

A more interesting question is about performing the disaggregation over the category hierarchy. This is referred to as drill-down. This situation occurs when different categories of the same hierarchy appear in the dimensions of the source summary databases. For example consider the source databases *Population(Region,Race,Age)*, and *Income(State,Race,Sex)*. Note that the dimension *Region* in the *Population* database, and the dimension *State* in the *Income* database belong to the same category hierarchy. Now, suppose that the target database is *Population(State,Race)*. We need to drill-down the population from the *Region* to the *State* level by using the *Income* database as a proxy. The usual technique for dis-aggregating by proxy is to use the linear indirect estimation by forming the full cross product at the lower category level (i.e., forming the drilled-down cross product), and then aggregating over the non-target dimensions. For this example, it is necessary to form the cross product $\hat{P}opulation(State,Race,Age,Sex)$, and then to aggregate over *Age*, and *Sex*.

We illustrate disaggregation by proxy using the example source databases introduced above. The drilled-down cross product is generated as follows:

$$\hat{P}opulation(State, Race, Age, Sex) = \\ Population(Region, Race, Age)\frac{Income(State, Race, Sex)}{\sum_{Sex} Income(Region, Race, Sex)}$$

Note that the term in the denominator *Income(Region,Race,Sex)* in the above expression is obtained by a roll-up operation on *Income(State,Race,Sex)*.

Here, again, the question is whether we can perform the PP method operations and the drill-down operation in either order and get results with the same accuracy. We prove that this is the case in Section 5.2.

In Section 5.3, we discuss the case where multiple category hierarchies are involved. We provide a procedure for the steps that need to be taken to achieve the maximum pre-aggregation that can be applied in this case without loss in accuracy.

As is stated in Section 2 on methodology, we consider here only the summary operators COUNT and SUM, and assuming that COUNT and SUM are available for all cells, the results also apply to the AVERAGE operator.

5.1 Using the PP Method with Rolling-up on Category Hierarchies

In this section we prove that the result obtained by rolling-up before applying the PP method is the same as performing the roll-up operation last. We assume that the dimension hierarchy to which the roll-up operator is applied is summarizable. Summarizability is a condition that states that it is possible to obtain from the summary database defined at category level A_1 of a given hierarchy, another summary database defined at the higher level A_2 of the same hierarchy by using the roll-up function. The conditions for summarizability or correctness of aggregations in OLAP are discussed in [Lenz and Shoshani 1997].

Let the roll-up operator be denoted by $\mathcal{R}_{A_1 \to A_2}(M(A_1))$, where A_1 and A_2 represent two category levels of a category hierarchy. It applies the aggregation function COUNT or SUM to the measure $M(A_1)$, and gives as result $M(A_2)$.

We address the case that each of the source summary databases has only one target dimension that requires a roll-up operation. We use the notation A_P^t and

A^t_Q to represent two different dimension-levels in the category hierarchy of the same dimension t of DB_P and DB_Q. For example, *State* $\rightarrow$ *Region* are two dimension-levels in the dimension *Geographical_area* and *Date* $\rightarrow$ *Month* are two dimension-levels in the dimension *Time*. We use the notation for lower and higher category levels as $A^{t,L}_P \rightarrow A^{t,H}_P$ of target dimension A^t_P. Similarly, $A^{t,L}_Q \rightarrow A^{t,H}_Q$ of target dimension A^t_Q. Finally, we use the notation $A^{T'}_P = A^{T^{C'}}_P \cup A^{T^{\overline{C'}}}_P$ to represent the remaining target dimensions not involved in the roll-up operation. Thus, $A^T_P = A^{T'}_P \cup A^{t,L}_P$. Similarly, $A^T_Q = A^{T'}_Q \cup A^{t,L}_Q$, where $A^{T'}_Q = A^{T^{C'}}_Q \cup A^{T^{\overline{C'}}}_Q$. Using this notation we can now formulate the following definition and theorem.

DEFINITION 5.1. Let $M_P(A^C_P, A^{\overline{C}}_P, A^{T^{C'}}_P, A^{T^{\overline{C'}}}_P, A^{t,L}_P)$ and $M_Q(A^C_Q, A^{\overline{C}}_Q, A^{T^{C'}}_Q, A^{T^{\overline{C'}}}_Q, A^{t,L}_Q)$ be source summary databases and $A^{t,L}_P \rightarrow A^{t,H}_P$, $A^{t,L}_Q \rightarrow A^{t,H}_Q$ represent category hierarchies of dimensions from the source summary databases. We define $\hat{M}$ to be the estimation result of a joint query over the source databases when applying the roll-up operator first and then the PP method. Conversely, we define $\hat{\hat{M}}$ to be the estimation results of a joint query over the source databases when applying the PP method first and then the roll-up operator. The expressions for $\hat{M}$ and $\hat{\hat{M}}$ are provided precisely below.

(i) The estimation result of a joint query $\hat{M}_P(A^{T^{C'}}_P, A^{T^{\overline{C'}}}_P, A^{t,H}_P, A^{T^{\overline{C'}}}_Q, A^{t,H}_Q)$ over the source summary databases is obtained by applying the roll-up operator first to the target dimensions in the target database and proxy database, and then applying the PP method as follows:

$$\mathcal{R}_{A^{t,L}_P \rightarrow A^{t,H}_P}\left(M_P(A^C_P, A^{\overline{C}}_P, A^{T^{C'}}_P, A^{T^{\overline{C'}}}_P, A^{t,L}_P)\right) = M_P(A^C_P, A^{\overline{C}}_P, A^{T^{C'}}_P, A^{T^{\overline{C'}}}_P, A^{t,H}_P)$$

$$\mathcal{R}_{A^{t,L}_Q \rightarrow A^{t,H}_Q}\left(M_Q(A^C_Q, A^{\overline{C}}_Q, A^{T^{C'}}_Q, A^{T^{\overline{C'}}}_Q, A^{t,L}_Q)\right) = M_Q(A^C_Q, A^{\overline{C}}_Q, A^{T^{C'}}_Q, A^{T^{\overline{C'}}}_Q, A^{t,H}_Q)$$

and then the PP method on $M_P(A^C_P, A^{\overline{C}}_P, A^{T^{C'}}_P, A^{T^{\overline{C'}}}_P, A^{t,H}_P)$ and $M_Q(A^C_Q, A^{\overline{C}}_Q, A^{T^{C'}}_Q, A^{T^{\overline{C'}}}_Q, A^{t,H}_Q)$ is given as follows:

$$\hat{M}_P(A^{T^{C'}}_P, A^{T^{\overline{C'}}}_P, A^{t,H}_P, A^C_Q, A^{T^{\overline{C'}}}_Q, A^{t,H}_Q) = M_P(A^C_P, A^{T^{C'}}_P, A^{T^{\overline{C'}}}_P, A^{t,H}_P)\frac{M_Q(A^C_Q, A^{T^{C'}}_Q, A^{T^{\overline{C'}}}_Q, A^{t,H}_Q)}{\sum_{A^{T^{\overline{C'}}}_Q, A^{t,H}_Q} M_Q(A^C_Q, A^{T^{C'}}_Q, A^{T^{\overline{C'}}}_Q, A^{t,H}_Q)}$$

Summarizing over the common dimensions, we have:

$$\sum_{A^C_Q} \hat{M}_P(A^{T^{C'}}_P, A^{T^{\overline{C'}}}_P, A^{t,H}_P, A^C_Q, A^{T^{\overline{C'}}}_Q, A^{t,H}_Q) = \hat{M}_P(A^{T^{C'}}_P, A^{T^{\overline{C'}}}_P, A^{t,H}_P, A^{T^{\overline{C'}}}_Q, A^{t,H}_Q)$$

(ii) The estimation result of a joint query $\hat{\hat{M}}_P(A^{T^{C'}}_P, A^{T^{\overline{C'}}}_P, A^{t,H}_P, A^{T^{\overline{C'}}}_Q, A^{t,H}_Q)$ over the source databases is obtained by applying the PP method to the source databases

and then the roll-up operator on the target dimensions as follows:

$$\hat{\hat{M}}_P(A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_P^{t,L}, A_Q^C, A_Q^{T^{\overline{C'}}}, A_Q^{t,L}) =$$
$$M_P(A_P^C, A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_P^{t,L}) \frac{M_Q(A_Q^C, A_Q^{T^{C'}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L})}{\sum_{A_Q^{T^{\overline{C'}}}, A_Q^{t,L}} M_Q(A_Q^C, A_Q^{T^{C'}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L})}$$

The application of the roll-up operator on $A_P^{t,L}$ in $\hat{M}_P(A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_P^{t,L}, A_Q^C, A_Q^{T^{\overline{C'}}}, A_Q^{t,L})$ will result:

$$\mathcal{R}_{A_P^{t,L} \to A_P^{t,H}} \left(\hat{\hat{M}}_P(A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_P^{t,L}, A_Q^C, A_Q^{T^{\overline{C'}}}, A_Q^{t,L}) \right)$$
$$= \hat{\hat{M}}_P(A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_P^{t,H}, A_Q^C, A_Q^{T^{\overline{C'}}}, A_Q^{t,L})$$

and then on $A_Q^{t,L}$ in this last result will give:

$$\mathcal{R}_{A_Q^{t,L} \to A_Q^{t,H}} \left(\hat{\hat{M}}_P(A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_P^{t,H}, A_Q^C, A_Q^{T^{\overline{C'}}}, A_Q^{t,L}) \right)$$
$$= \hat{\hat{M}}_P(A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_P^{t,H}, A_Q^C, A_Q^{T^{\overline{C'}}}, A_Q^{t,H})$$

Summarizing over common dimension A_Q^C, we have $\hat{\hat{M}}_P(A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_P^{t,H}, A_Q^{T^{\overline{C'}}}, A_Q^{t,H})$.

THEOREM 5.1. *Let* $M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_P^{t,L})$ *and* $M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^{C'}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L})$ *be source summary databases. The estimators of target database* $\hat{M}(A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_P^{t,H}, A_Q^{T^{\overline{C'}}}, A_Q^{t,H})$ *and* $\hat{\hat{M}}(A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_P^{t,H}, A_Q^{T^{\overline{C'}}}, A_Q^{t,H})$ *give the same results.*

PROOF. We present here the sketch of the proof. The formal proof is given in the *electronic appendix*. In this proof the roll-up operator is considered to be a particular kind of summation operation. Hence, the proof proceed by showing that $\hat{M}$ is equivalent to $\hat{\hat{M}}$ through intermediate equations, which commutes/associates/distributes with the other summation operators similar to the proof of Theorem 4.1. □

5.2 Using the PP Method with Drilling-down on Category Hierarchies

This section contains a theorem that proves that similar to the case of roll-up, the same accuracy is achieved regardless of the order of applying pre-aggregation and drill-down operations. In order to prove this, we use the notation introduced in the previous section.

Recall that the drill-down operation can only be performed when the dimensions in the two source summary databases that are involved in the drill-down must belong to the same category hierarchy. Furthermore, the lower category must belong to the proxy database. That is, $A_Q^{t,L} \to A_P^{t,H}$.

DEFINITION 5.2. Let $M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_P^{t,H})$ and $M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^{C'}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L})$ be source summary databases and $A_Q^{t,L} \to A_P^{t,H}$. We define $\hat{M}$ to be the es-

timation result of a joint query over the source summary databases when applying the drill-down operator first and then the PP method. Conversely, we define $\hat{\hat{M}}$ to be the estimation results of a joint query over the source databases when applying the PP method first and then the drill-down operator. The expressions for $\hat{M}$ and $\hat{\hat{M}}$ are provided precisely below.

(i) The estimation result of a joint query $\hat{M}_P(A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L})$ consists of the following steps: 1a) generating the full cross product obtained by the drill-down operation; 1b) summarizing over the common and non-common dimensions in the result of part 1a.

Step 1a (drill-down):

$$\hat{M}_P(A_P^{\overline{C}}, A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_Q^{C}, A_Q^{\overline{C}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L}) =$$
$$M_P(A_P^{C}, A_P^{\overline{C}}, A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_P^{t,H}) \frac{M_Q(A_Q^{C}, A_Q^{\overline{C}}, A_Q^{T^{C'}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L})}{\sum_{A_Q^{\overline{C}}, A_Q^{T^{\overline{C'}}}} M_Q(A_Q^{C}, A_Q^{\overline{C}}, A_Q^{T^{C'}}, A_Q^{T^{\overline{C'}}}, A_P^{t,H})}$$

where the term in the denominator is obtained using roll-up as follows:

$$M_Q(A_Q^{C}, A_Q^{\overline{C}}, A_Q^{T^{C'}}, A_Q^{T^{\overline{C'}}}, A_P^{t,H}) = \mathcal{R}_{A_Q^{t,L} \rightarrow A_P^{t,H}} \left(M_Q(A_Q^{C}, A_Q^{\overline{C}}, A_Q^{T^{C'}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L}) \right)$$

Step 1b (summarization):

$$\sum_{A_Q^{C}, A_P^{\overline{C}}, A_Q^{\overline{C}}} \hat{M}_P(A_P^{\overline{C}}, A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_Q^{C}, A_Q^{\overline{C}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L}) = \hat{M}_P(A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L}) \tag{6}$$

(ii) The estimation result of a joint query $\hat{\hat{M}}_P(A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L})$ consists of the following steps: 2a) pre-aggregating the non-common dimensions over the source summary databases; 2b) applying the drill-down operation to the result of step 2a; 2c) summarizing over the common dimensions on the result of step 2b.
Step 2a (pre-aggregation):

$$M_P(A_P^{C}, A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_P^{t,H}) = \sum_{A_P^{\overline{C}}} M_P(A_P^{C}, A_P^{\overline{C}}, A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_P^{t,H})$$
$$M_Q(A_Q^{C}, A_Q^{T^{C'}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L}) = \sum_{A_Q^{\overline{C}}} M_Q(A_Q^{C}, A_Q^{\overline{C}}, A_Q^{T^{C'}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L})$$

Step 2b (drill-down):

$$\hat{\hat{M}}_P(A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_Q^{C}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L})$$
$$= M_P(A_P^{C}, A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_P^{t,H}) \frac{M_Q(A_Q^{C}, A_Q^{T^{C'}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L})}{\sum_{A_Q^{T^{\overline{C'}}}} M_Q(A_Q^{C}, A_Q^{T^{C'}}, A_Q^{T^{\overline{C'}}}, A_P^{t,H})}$$

Step 2c (summarization):

$$\sum_{A_Q^C} \hat{M}_P(A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_Q^C, A_Q^{T^{\overline{C'}}}, A_Q^{t,L}) = \hat{M}_P(A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L}) \qquad (7)$$

THEOREM 5.2. *Let* $M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_P^{t,H})$ *and* $M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^{C'}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L})$ *be source summary databases and* $A_Q^{t,L} \rightarrow A_P^{t,H}$. *The estimators of target database* $\hat{M}_P(A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L})$ *and* $\hat{\hat{M}}_P(A_P^{T^{C'}}, A_P^{T^{\overline{C'}}}, A_Q^{T^{\overline{C'}}}, A_Q^{t,L})$ *give the same results.*

PROOF. We present here the sketch of the proof. The formal proof is given in the *electronic appendix*. In this proof, the drill-down operator is considered to be a particular kind of a distribution operator. Similar to the proof of Theorem 5.1 this proof shows that $\hat{M}$ is equivalent to $\hat{\hat{M}}$ through intermediate equations, which commutes/associates/distributes with the other summation operators. □

5.3 Applying the PP Method to Multiple Category Hierarchies

Consider the following example of two source summary databases: *Population(Region, Race,Month,Age)* and *Income(State,Race,Year,Sex)*. Suppose the target database is *Population(State,Year)*. We note that this requires drill-down the *Population* from the *Region* level to the *State* level, and roll-up from the *Month* level to the *Year* level. To achieve the desired result, it is possible to first pre-aggregate over the non-common dimensions (excluding dimensions that belong to the same category hierarchy) to get *Population(Region,Race,Month)* and *Income(State,Race,Year)*. Then we can roll-up from the *Month* level to the *Year* level to generate *Population(Region, Race,Year)*. Next, according to the PP method for disaggregation described in Section 5.2, we use *Population(Region,Race,Year)* and *Income(State,Race,Year)* to get $\hat{P}$*opulation(State,Race,Year)*, and finally aggregate over *Race* to get $\hat{P}$*opulation(State, Year)*. Note that if we consider the target database *Income(Region,Month)* the opposite is necessary: roll-up from *State* to *Region*, and drill-down from *Year* to *Month*.

From this example, the procedure for determining the steps to achieve maximum pre-aggregation without loss in accuracy can be generalized. The procedure is as follows:

(1) Determine if there are different dimensions in the source summary databases that belong to the same category hierarchy. If there are none, perform the method PP described in Section 3. Otherwise, proceed to Step 2. In the example above, there are two such cases: *State* → *Region*, and *Month* → *Year*. So we proceed to Step 2.

(2) Aggregate over the non-common dimensions. In this example, aggregate over *Age* and *Sex* to form *Population(Region,Race,Month)* and *Income(State,Race,Year)*.

(3) Examine the target measure and the target dimensions. In our example, the target was *Population(State,Year)*. Select the source database according to the target measure. In the example above, we select *Population(Region,Race,Month)*.

(4) Roll-up all possible dimensions according to the target dimensions. In this example *Month* will be rolled up to generate *Population(Region,Race,Year)*.

(5) Roll-up all remaining (non-target) common dimensions that belong to the same category hierarchy to the higher level in order to get matching levels. We do not have such a case in this example, but if you add *Profession* and *Professional-category* into the two source summary databases, correspondingly, then *Profession* should be rolled-up to *Professional-category*.

(6) Now, we have databases with either identical dimensions or dimensions that belong to the same category hierarchy. Perform the dis-aggregation (drill-down) according to the method in Section 5.2. In this example, this generates $\hat{P}opulation(State, Race, Year)$.

(7) Finally, aggregate over the common (non-target) dimensions. In this example, aggregating over *Race* generates the desired result *Population(State, Year)*.

We note that in the above procedure whenever there is an opportunity to perform aggregation over non-common dimensions and roll-up and/or drill-down operations, they should be performed together to save the cost of generating the intermediate databases.

6. FORMULAS FOR PERFORMANCE EVALUATION

In applying the PP method to numerous examples we noticed that the performance gain can be several fold. Intuitively, one can expect that because in the PP method we first reduce the dimensionality of the source summary databases before performing the cross product. Yet, we would like to characterize more accurately the computational cost estimation. In this section, we develop cost estimation formulas.

In order to estimate the performance of the various methods, we have to count the number of primitive operations that each step takes. We start with estimating the cost of generating an aggregation over a single database, and follow that with the cost of generating a cross product of two databases. We then use these cost formulas to express the total cost of the methods F, PP, and P. This is followed by a theorem that states that the cost of the PP method is always lower than the cost of the F method provided that the source databases have at least one non-common dimension. We note that our analysis is based on an approximate cost model, and accordingly the above stated results are based on this model.

6.1 The Cost of Generating an Aggregation

In the following Lemma we show that the upper bound of the number of primitive operations (specifically, COUNT or SUM) required for aggregating over a single multi-dimensional summary database is the same regardless of the number of dimensions we aggregate over.

LEMMA 6.1. *The upper bound of the number of primitive operations to aggregate a multi-dimensional summary database to any number of dimensions is the product of the cardinalities of the dimensions.*

PROOF. Consider a multi-dimensional summary database DB_P defined as $M_P(A_P^1, A_P^2, \ldots, A_P^n)$, with cardinalities $|A_P^1|$, $|A_P^2|$, ..., $|A_P^n|$. Suppose that we aggregate over one dimension A_P^i. Then the number of cells that are generated in the output is: $|A_P^1||A_P^2|\ldots|A_P^{i-1}||A_P^{i+1}|\ldots|A_P^n|$. The number of primitive operations required to compute each cell (e.g., SUM) is $|A_P^i| - 1$; that is, the cardinality of A_P^i minus

one operation. For example, if the cardinality is 10, then only 9 operations are needed to generate the sum. Thus, the number of primitive operations is:

$$|A_P^1||A_P^2|\dots|A_P^{i-1}|(|A_P^i|-1)|A_P^{i+1}|\dots|A_P^n| \tag{8}$$

It follows that the upper bound can be approximated to $|A_P| = |A_P^1||A_P^2|\dots|A_P^n|$. This upper bound is very tight, especially when the dimension being aggregated has a large cardinality.

Now, consider the case of aggregating over two dimensions A_P^i and A_P^j. The number of cells generated is: $|A_P^1||A_P^2|\dots|A_P^{i-1}||A_P^{i+1}|\dots|A_P^{j-1}||A_P^{j+1}|\dots|A_P^n|$. The number of operations to generate one cell is the number of elements to sum over minus one: $(|A_P^i||A_P^j|-1)$. Again, rounding off this term we get that the cost estimation is again: $|A_P| = |A_P^1||A_P^2|\dots|A_P^n|$. It follows that the same expression results when we consider three dimensions, etc. Thus, the upper bound of the number of primitive operations is $|A_P|$ regardless of the level of aggregation (the number of dimensions we aggregate over). □

LEMMA 6.2. *The lower bound of the number of operations to aggregate a multi-dimensional summary database over any number of dimensions is one half of the product of the cardinalities of the dimensions. This lower bound occurs only in the degenerate case where the aggregation is over a single dimension whose cardinality is 2.*

PROOF. The proof follows directly from expression (8) in the proof of Lemma 6.1, for the case that only one dimension A_P^i is aggregated and $|A_P^i| = 2$. □

Typically, aggregation occurs over more than one dimension and dimensions have more than two values. Therefore, for all practical purposes we will assume the upper bound in the rest of the paper.
If we use the notation of "common", "non-common", and "target" dimensions to represent a database M_P according to the syntax introduced in Section 2.4, the product of the cardinalities of $M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}})$ can be represented by $X_P = X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}}$, where the notation"$X$" represents a cardinality-product. For instance, the total number of operations to generate any aggregation over the summary database DB_P = *Population(State,Age,Race,Sex)* is $X_P = 504$, given that the cardinality of the domain values of the dimensions *State*, *Age*, *Race*, *Sex* are respectively: 9, 7, 4, 2. Suppose *State* is the non-common target dimension, *Age*, *Sex* are the common dimensions, and *Race* is the non-common dimension, it follows that $X_P^C = 14$, and $X_P^{\overline{C}} = 4$, $X_P^{T^C} = 1$, and $X_P^{T^{\overline{C}}} = 9$.

We denote the primitive operation cost (for either COUNT or SUM) as C_{po}. Therefore the cost of aggregations over the database $M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}})$ is

$$C_{po}(X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}})$$

6.2 The Cost of Generating the Cross Product

The formula for calculating a cross product is Eq. 3, shown below, as discussed in Section 3.

$$\hat{M}_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}}, A_Q^{\overline{C}}, A_Q^{T^{\overline{C}}})$$
$$= M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}}) \frac{M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})}{\sum_{A_Q^{\overline{C}}, A_Q^{T^{\overline{C}}}} M_Q(A_Q^C, A_Q^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}})}$$

To find out the cost of generating the cross product, the expression in the denominator has to be evaluated. Note that this expression is a sub-cube that needs to be calculated only once, but its elements are used multiple times in subsequent operations. That cost is simply the cost of aggregating over all the non-common and target dimensions in M_Q. From Lemma 6.1 it follows that this cost is $C_{po}(X_Q^C X_Q^{\overline{C}} X_Q^{T^C} X_Q^{T^{\overline{C}}})$.

Now, for each of the cells in the cross product, it is necessary to perform one multiply and one divide operation. We assume that the cost of divide and multiply is about the same, and that it is usually larger than COUNT or SUM, by a factor $\alpha \geq 1$. The number of cells in the cross product is: $X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{\overline{C}} X_Q^{T^{\overline{C}}}$. Thus the total cost for a cross product is:

$$C_{po}(X_Q^C X_Q^{\overline{C}} X_Q^{T^C} X_Q^{T^{\overline{C}}}) + 2\alpha C_{po}(X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{\overline{C}} X_Q^{T^{\overline{C}}})$$

6.3 Cost Formulas for the Three Methods

We can now derive the cost formulas for methods F, PP, and P. For the F method the total cost consists of the cost of generating the full cross product (shown in square brackets) followed by a post-aggregation to the target dimensions. Specifically the cost is:

$$\Big[C_{po}(X_Q^C X_Q^{\overline{C}} X_Q^{T^C} X_Q^{T^{\overline{C}}}) + 2\alpha C_{po}(X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{\overline{C}} X_Q^{T^{\overline{C}}})\Big]$$
$$+C_{po}(X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{\overline{C}} X_Q^{T^{\overline{C}}}) \quad (9)$$

For the PP method, the total cost consists of first pre-aggregating each of the source summary databases over the non-common dimensions, then generating the cross product of the pre-aggregated databases, and then post-aggregating to the target dimensions. Specifically, the cost is:

$$C_{po}(X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}}) + C_{po}(X_Q^C X_Q^{\overline{C}} X_Q^{T^C} X_Q^{T^{\overline{C}}})$$
$$+ \Big[C_{po}(X_Q^C X_Q^{T^C} X_Q^{T^{\overline{C}}}) + 2\alpha C_{po}(X_P^C X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{T^{\overline{C}}})\Big] + C_{po}(X_P^C X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{T^{\overline{C}}}) \quad (10)$$

Note that in the PP method, generating the cross product and the post-aggregation, the non-common dimensions are not in the expressions since they have been eliminated in the pre-aggregation.

For the P method, the total cost consists of first pre-aggregating each of the source summary databases over the common and non-common dimensions, and then generating the cross product of the pre-aggregated databases. Specifically, the cost is:

$$C_{po}(X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}}) + C_{po}(X_Q^C X_Q^{\overline{C}} X_Q^{T^C} X_Q^{T^{\overline{C}}}) + \left[C_{po}(X_Q^{T^C} X_Q^{T^{\overline{C}}}) + 2\alpha C_{po}(X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{T^{\overline{C}}})\right] \quad (11)$$

Note that generating the cross product in the P method, the common and non-common dimensions are not in the expressions since they have been eliminated in the pre-aggregation.

Using expression (9) and (10) we are now ready to state and prove the following theorem.

6.4 Performance Domination of PP Method over F Method

THEOREM 6.1. *Let $M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}})$ and $M_Q(A_Q^C, A_Q^{\overline{C}}, A_Q^{T^C}, A_Q^{T^{\overline{C}}})$ be source summary databases that have at least one non-common dimension each, then according to the approximate cost model defined previously in this section, using the PP method is always less expensive computationally than using the F method.*

PROOF. The proof consists of comparing the computational cost of the PP and F methods. We wish to show that:

$$\begin{aligned}
&C_{po}(X_Q^C X_Q^{\overline{C}} X_Q^{T^C} X_Q^{T^{\overline{C}}}) + C_{po}(X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}}) \\
&+ \left[C_{po}(X_Q^C X_Q^{T^C} X_Q^{T^{\overline{C}}}) + 2\alpha C_{po}(X_P^C X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{T^{\overline{C}}})\right] + C_{po}(X_P^C X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{T^{\overline{C}}}) \\
&< C_{po}(X_Q^C X_Q^{\overline{C}} X_Q^{T^C} X_Q^{T^{\overline{C}}}) + 2\alpha C_{po}(X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{T^{\overline{C}}} X_Q^{\overline{C}} X_Q^{T^{\overline{C}}}) \\
&\quad + C_{po}(X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{\overline{C}} X_Q^{T^{\overline{C}}})
\end{aligned}$$

The first term in each side is the same and can be removed. Comparing the last terms in each side in the above expression, which represent the cost of post-aggregation, it is obvious that

$$C_{po}(X_P^C X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{T^{\overline{C}}}) < C_{po}(X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{\overline{C}} X_Q^{T^{\overline{C}}})$$

because $X_P^C X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{T^{\overline{C}}}$ is a subset of $X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{\overline{C}} X_Q^{T^{\overline{C}}}$. Therefore, both terms can be removed, and the inequality reduced to:

$$\begin{aligned}
&X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}} + X_Q^C X_Q^{T^C} X_Q^{T^{\overline{C}}} + 2\alpha(X_P^C X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{T^{\overline{C}}}) \\
&< 2\alpha(X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{\overline{C}} X_Q^{T^{\overline{C}}})
\end{aligned}$$

This inequality can be written as follows:

$$\left(\frac{1}{2\alpha X_Q^{\overline{C}} X_Q^{T^{\overline{C}}}}\right) + \left(\frac{1}{2\alpha X_P^{\overline{C}} X_P^{T^{\overline{C}}} X_Q^{\overline{C}}}\right) + \left(\frac{1}{X_P^{\overline{C}} X_Q^{\overline{C}}}\right) < 1$$

Every dimension has, by definition, at least two category values. Also $X_Q^{T^{\overline{C}}}$ must always exist because $A_Q^{T^{\overline{C}}}$ is the proxy dimension(s) (see Section 2.4). Since, we have at least two terms in the denominators of the above expression, the first two fractions must be less than or equal to $\frac{1}{8}\alpha$, and the last fraction must be less than

or equal to $\frac{1}{4}$. Since $\alpha \geq 1$, the expression is always less than 1. Thus, if there is at least one non-common dimension, using the PP method is always less expensive computationally than using the F method. □

Example 5 For illustrative purposes we assume that $\alpha = 2$; that is, the cost of a multiplication or a division is twice the cost of a sum. Let us consider the summary databases $Income(Age, Education, Sex)$ and $Population(State, Age, Race, Sex)$ shown in Tables I and II, which are used to generate the estimation for $\hat{I}ncome(State)$. For this example, we use the cardinalities 50, 10, 7, 2, 12 for the dimensions *State*, *Age*, *Race*, *Sex*, and *Education*, respectively. In other words, we have:

$$X_P^C = 20 \;\; X_P^{\overline{C}} = 12 \;\; X_P^{T^C} = 1 \quad X_P^{T^{\overline{C}}} = 1$$
$$X_Q^C = 20 \quad X_Q^{\overline{C}} = 7 \;\; X_Q^{T^C} = 1 \;\; X_Q^{T^{\overline{C}}} = 50$$

Recall that in this example, only one target dimension exists: $A_Q^{T^{\overline{C}}} = \{State\}$, $A_Q^{T^C} = \emptyset$, $A_P^{T^C} = A_P^{T^{\overline{C}}} = \emptyset$, and therefore we use $X_P^{T^C} = X_P^{T^{\overline{C}}} = 1$ and $X_Q^{T^C} = 1$ to neutralize these terms. Applying the F method, the total cost is $427000C_{po}$, while using the PP method, the total cost is $13240C_{po}$. In other words, the computational cost of the PP method is less than the F method by a factor 32.25.

7. EVALUATION OF ACCURACY AND COST TRADE-OFFS

While we proved theoretically that there is a benefit of using the PP method rather than the F method since they provide the same accuracy level, there is the practical question of how much computation is saved by using the PP method. A second practical question is how much better is the accuracy of the PP method relative to using the P method, and what is the cost penalty for that. Put it another way, is it worth paying the cost of the PP method over the P method to gain in accuracy? We address these questions in this section. We evaluate first experimentally the accuracy question to find which parameters the accuracy is sensitive to. Then, we evaluate the cost benefit of using the PP method versus the F method, and the cost penalty of using PP method vs. P method.

7.1 Accuracy Analysis

Intuitively, the accuracy of estimation is a function of the correlation between the measures of the source summary databases, and also depends on the common dimensions between them. Therefore, there should be significant loss of accuracy if we use the P method, because we pre-aggregate over common dimensions. However, it is also reasonable to expect that if the cardinalities of the dimensions are small, the difference in accuracy between the two methods will be small, because there is less information to support the correlations. Thus, we selected example databases that vary in these two parameters: the correlation level, and the cardinality level.

We used the following databases: *Income(Age,Race)*, and *Population(State,Age, Race)* to estimate *Income(State)*. For the low cardinalities we used 9, 7, 2 for *State*, *Age*, and *Race*, respectively. For the high cardinality we used 27, 15, and 7, respectively.

We varied the correlation by changing the distribution of the measures (X and Y) over the common dimensions. In general, the closer the distribution patterns,

Table VII. *Correlation* and *Average Relative Error*

High cardinality of common dimensions				*Low cardinality of common dimensions*		
Corr.	PP	P	*Diff.*	PP	P	*Diff.*
0.85	0.199	0.602	0.402	0.147	0.176	0.028
0.7	0.325	0.887	0.562	0.256	0.281	0.025
0.55	0.424	1.043	0.619	0.329	0.343	0.014
0.4	0.441	1.281	0.840	0.351	0.358	0.006

the higher the correlation. We calculated correlation figures using the well-known formula, where the measures are X and Y, and N is the total number of cells over the common dimensions, as follows:

$$r = (1/(N-1)) \sum_{i=1}^{N} \left(\frac{X_i - \overline{X}}{\sigma_X} \right) \left(\frac{Y_i - \overline{Y}}{\sigma_Y} \right)$$

where $\overline{X}, \overline{Y}$ are the mean for X and Y respectively, and $\sigma_X = \sqrt{(1/N) \sum_{i=1}^{N} (X_i - \overline{X})^2}$ is the standard deviation. Similarly, $\sigma_Y = \sqrt{(1/N) \sum_{i=1}^{N} (Y_i - \overline{Y})^2}$.

The results are shown in Table VII and are presented graphically in figures 2 and 3. Figure 2 shows that in the case of high cardinality, the difference between the average relative error (ARE, represented as percent) of the PP method and the P method is quite high for all correlation levels (40% or more). For example, for the high correlation of 0.85, the ARE for the PP method is 19%, while for the P method it is 60%. As expected, the error for higher correlations is lower. Interestingly, the error difference grows as the correlation level gets lower.

In contrast, Figure 3 shows that in the case of low cardinality the difference in accuracy is quite small (3% or less). This observation raises the question as to whether it is worth paying the extra cost of using the PP method when the cardinality is low. We address this question in the next section.

7.2 Cost Trade-offs

The cost formulas for the three methods, F, PP, and P were developed in Section 6. We recall that the cost formulas are:
for F method:

$$\begin{aligned} &\left[C_{po}(X_Q^C X_Q^{\overline{C}} X_Q^{T^C} X_Q^{T^{\overline{C}}}) + 2\alpha C_{po}(X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{\overline{C}} X_Q^{T^{\overline{C}}}) \right] \\ &+ C_{po}(X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{\overline{C}} X_Q^{T^{\overline{C}}}) \end{aligned} \tag{12}$$

for PP method:

$$\begin{aligned} &C_{po}(X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}}) + C_{po}(X_Q^C X_Q^{\overline{C}} X_Q^{T^C} X_Q^{T^{\overline{C}}}) \\ &+ \left[C_{po}(X_Q^C X_Q^{T^C} X_Q^{T^{\overline{C}}}) + 2\alpha C_{po}(X_P^C X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{T^{\overline{C}}}) \right] \\ &+ C_{po}(X_P^C X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{T^{\overline{C}}}) \end{aligned} \tag{13}$$

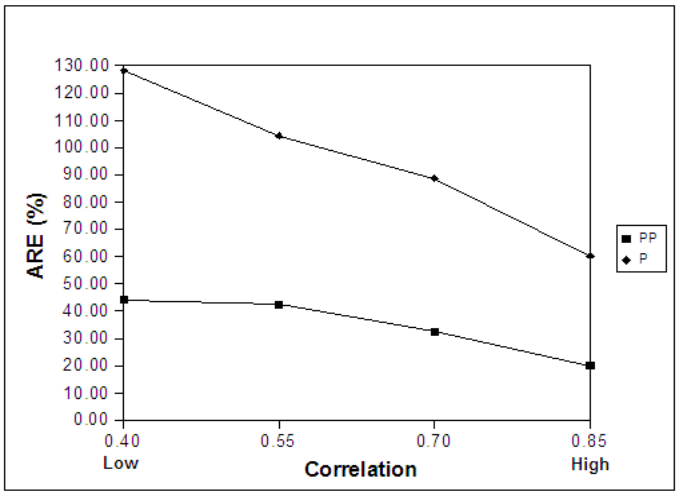

Fig. 2. ARE(%) in the case of high cardinality of dimensions

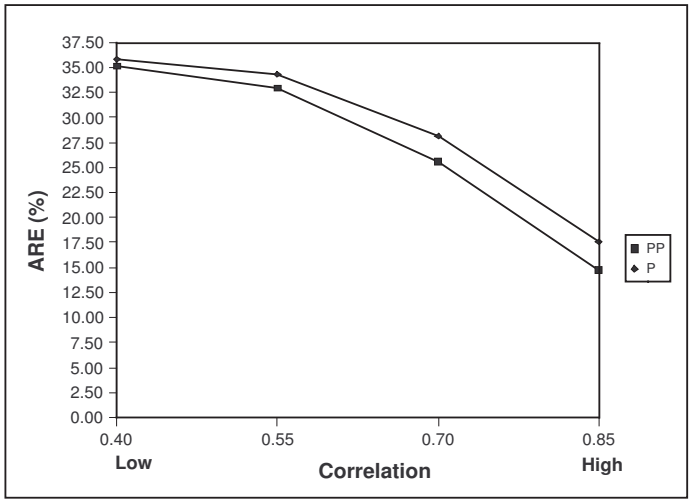

Fig. 3. ARE(%) in the case of low cardinality of dimensions

for P method:

$$C_{po}(X_P^C X_P^{\overline{C}} X_P^{T^C} X_P^{T^{\overline{C}}}) + C_{po}(X_Q^C X_Q^{\overline{C}} X_Q^{T^C} X_Q^{T^{\overline{C}}}) + \left[C_{po}(X_Q^{T^C} X_Q^{T^{\overline{C}}}) + 2\alpha C_{po}(X_P^{T^C} X_P^{T^{\overline{C}}} X_Q^{T^{\overline{C}}})\right] \quad (14)$$

For evaluating the cost tradeoffs, we take the ratios (the gain) of the cost between the methods when applied to the same source databases. From the cost formulas, it is evident that the cost is dependent on the products of cardinalities of the common, non-common, and the target dimensions. Since we cannot get closed form formulas for the ratios, we evaluated the gains and generated multiple graphs to observe the behavior of the gain when varying the cardinality of the dimensions. We note that in the formulas, the cardinality measures represent the product of cardinalities for each type of dimension. For example, the expression X_P^C represents the product of all common dimensions (if we have 3 dimensions with cardinalities of 2, 5, and 10, then $X_P^C = 100$). We will use below the terms "common cardinality-product", "non-common cardinality-product", and "target cardinality-product" for the product of each, correspondingly. The target cardinality-product refers to the product of the cardinality of the common and non-common target dimensions.

We start with graphing the gain of the F method over the PP method. We show two graphs. In Figure 4, we fix the common cardinality-product (=100) and vary the non-common and target cardinalities (between 10 and 200). From this graph, we observe that the gain increases super-linearly with the increase of the cardinality-product of non-common dimensions. We also observe that the target cardinality-product has a large effect on the gain, although the gain decreases as the target cardinality-product increases.

In Figure 5, we fix the non-common cardinality-product (=100), and vary the common and target cardinalities (between 10 and 200). Interestingly, the common cardinality-product has no effect on the gain. This can be explained by observing that $X_P^C = X_Q^C$ can be factored out in both the F method and PP method expressions, and thus have no effect when we take the ratio. We observe, again, that the target cardinality-product has a large effect on the gain, but the gain decreases as the target cardinality-product increases. We also observe that the gain figures are very large, in the order of several 1000's.

How can this very high gain be explained intuitively? We observe that the largest term for the F method has 6 cardinality-product elements, while the largest term for the PP method has only 4 cardinality-product elements. This is the reason that pre-aggregating over the non-common dimensions is highly effective.

Next, we graph the gain of the PP method over the P method. In Figure 6, we fix the common cardinality-product (=100) and vary the non-common and target cardinalities (between 10 and 200). We observe that for low cardinalities the gain is in the range of 10's, but quickly goes down to single digit levels as the cardinality increases even for high target cardinalities. In Figure 7, we fix the non-common cardinality-product (=100), and vary the common and target cardinalities (between 10 and 200). Here again, we see a very small effect on the gain when varying the common cardinality-product.

Why is the gain of the P method over the PP method relatively small compared to the PP method over the F method case? In contrast to the previous case, we observe that largest terms for the PP method as well as the P method have 4 cardinality-product elements, and thus the ratio is small.

7.3 Discussion

We can now respond to the two questions posed in the beginning of this section.

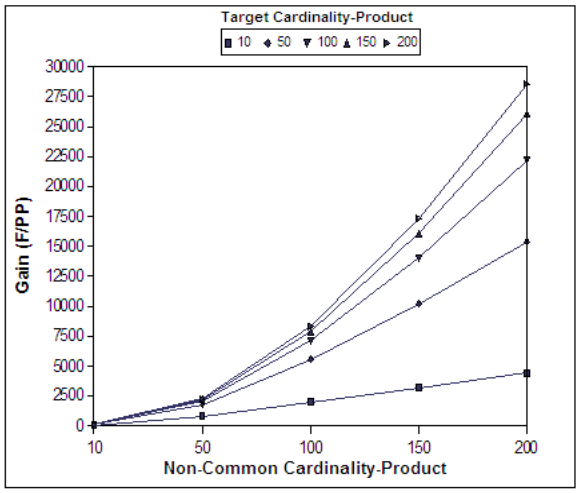

Fig. 4. Gain of F method over PP method with fixed common cardinality-product

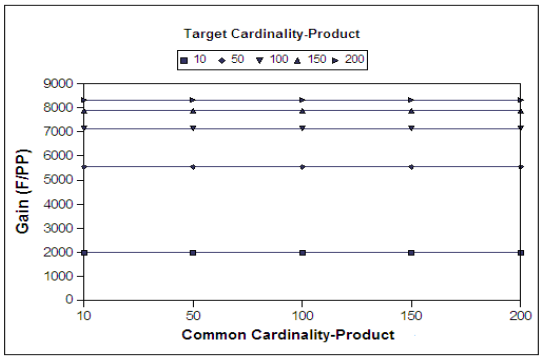

Fig. 5. Gain of F method over PP method with fixed non-common cardinality product

1) How much computation is saved by using the PP method rather than the F method?

The experimental results show that the savings behave as a super-linear function of the product of the non-common cardinalities. Furthermore, the gain increases with the target cardinality-product, but in a sub-linear fashion. We note that it is only necessary to have two non-common dimensions with cardinality of 10 each to get a gain in the order of 100. Typical cardinalities can range from low (e.g., 2 for sex), to quite high (e.g., 50 for states), or even 100's (e.g., product types). Thus, pre-aggregation over just a few non-common dimensions can achieve very high gains.

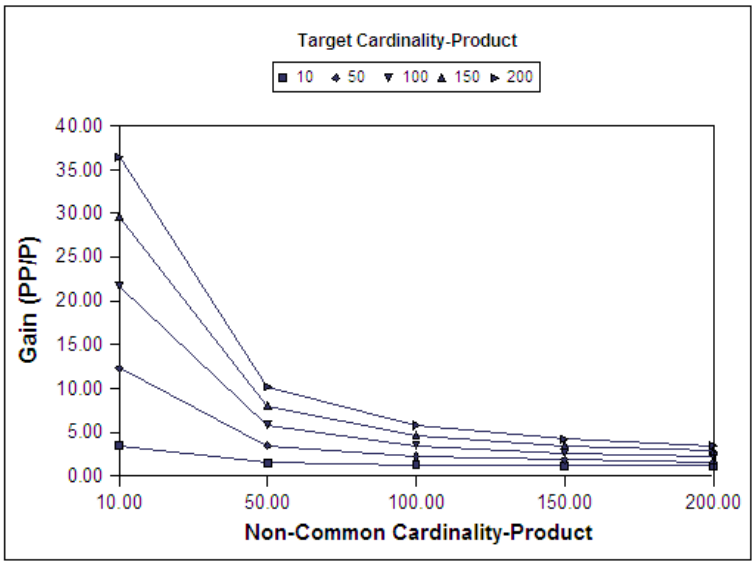

Fig. 6. Gain of PP method over P method with fixed common cardinality-product

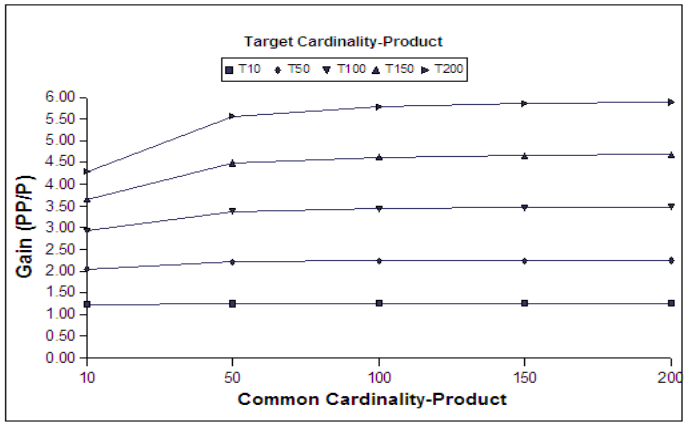

Fig. 7. Gain of PP method over P method with fixed non-common cardinality-product

2) Is it worth paying the cost of the PP method to gain in accuracy?

The cost penalty for using the PP method relative to the P method is low for high cardinality-product (order of 3 or less). Even for low cardinality-product the gain is relatively low (order of a few 10's). The main reason is that the pre-aggregation cost in the P and PP methods dominates the total cost. Since the accuracy obtained by the PP method is much higher than the P method for high cardinality-product

of common dimensions, and the penalty is low, there is no doubt that in this case PP method should be used. In the case of a low cardinality-product, where the accuracy gain is low, it may not be worth performing the extra operations even though the cost is relatively low. However, in some applications, even 3% in accuracy is significant, and the moderate extra cost may still be worth even in these cases.

8. EXTENDING THE QUERY ESTIMATION RESULTS TO MULTIPLE SOURCE DATABASES

From a statistical analysis point of view, the most common estimation operation is over two databases, where one is a database with a particular target measure (e.g., sales) using another database as a proxy database with another measure (e.g., income). The advantage of using the PP method for the case of two databases has been presented in previous sections. However, in some cases, the estimation may require the use of more than two databases. In practical situations the databases may come from different sources (such as agencies responsible for economic activities, population census, education, health, etc.) In such cases many of the dimensions in the databases are non-common. In this section, we explore under what conditions we can apply the PP method for performing estimation over more than two databases. We start with an example to illustrate the concepts of using multiple proxy databases.

Consider the following three summary databases:

$$DB_A : Sales(State, Type\_of\_business)$$
$$DB_B : Income(Race, Education, Age)$$
$$DB_C : Population(County, Sex, Age)$$

Suppose that the desired target database is:

$$DB_T = Sales(State, Sex, Race)$$

We will prove in this section the conditions under which we can apply the PP method. In this particular example, our results will show that we can perform pre-aggregation over the dimensions "Type_of_business", "Education" (but not "Age"), and also rolling-up over "County" to the "State" level in order to generate a greatly reduced cross-product of *Sales(State,Sex,Race,Age)*, and then summarize over "Age". As noted earlier, when pre-aggregation is applied even over a small number of dimensions the saving in the computational complexity can be very large if the cardinality of the dimensions is large. In this example, the "Type_of_business" dimension has typically a very large cardinality, and on average, there are a large number of counties for each state, and thus the savings can be several orders of magnitude.

When using multiple proxy databases, the number of databases involved is typically small for the simple reason that the number of databases used cannot exceed the number of dimensions desired in the target database. In the example above, there are three dimensions requested in the target database, where the first (*State*) is in the same hierarchy dimension of another database, the second (*Sex*) is only in

one database, and the third (*Race*) is in another. Therefore, at most the number of databases involved is equal to the number of dimensions in the target database.

8.1 The Order of Performing the Cross Product

Since we have more that one proxy database, the first question is in which order to apply them in the estimation process. It turns out that, in general, the order of applying the proxy database produces different results. Yet, we are interested in finding out the conditions under which the results produced are the same regardless of order. We prove in this section that under a condition that we refer to as "Proxy-Non-Commonality" (PNC) condition the order of applying the proxy databases can be arbitrary. Given several source databases, one of the databases is chosen as a primary database because its measure is requested in the query as a target measure. Accordingly, the PNC condition can be stated as follows:

DEFINITION 8.1. Given a primary database and two or more proxy databases, the *Proxy-Non-Commonality* (PNC) condition requires that all the dimensions of the proxy databases that are not in the primary database must be mutually exclusive.

We show in this section that if the PNC condition holds, the same result is obtained by applying the proxy databases in any order. To illustrate this point consider the example shown in the introductory part of Section 8. The "Sales" database A is the primary database. The dimension-level "County" in database C is in the same dimension (*Geographical_area*) as the dimension-level "State" in the primary database A. Thus, we are only concerned with the remaining dimensions in the databases B and C, and whether they are mutually exclusive:

$$DB_B : Income(Race, Education, Age)$$
$$DB_C : Population(Sex, Age)$$

As can be seen, this example does not fulfill the PNC condition because "Age" is common to both databases. Therefore, the order of applying the proxy databases matters.

Now, consider the following variation on this example, where database A has the target measure:

$$DB_A : Sales(State, Type\_of\_business)$$
$$DB_B : Income(State, Race, Education, Age)$$
$$DB_C : Population(County, Sex, Marital\_status)$$

To evaluate the PNC condition, we do not consider "State" in database B and "County" in database C. As can be seen below the non-commonality PNC condition is met in the remaining dimensions of databases B and C.

$$DB_B : Income(Race, Education, Age)$$
$$DB_C : Population(Sex, Marital\_status)$$

What can be done in case that the PNC condition does not hold? Obviously some order has to be chosen. We will show in the next sections the condition under which the PP method gives the same result as the F method, given a particular order of applying the proxy databases. As will be seen, unlike the PNC condition,

this condition allows partially or fully common dimensions to exist, but states that pre-aggregation can be performed only over the mutually exclusive, non-common dimensions. Further, unless the PNC condition holds, different results can be obtained depending on the order of applying the proxy databases.

The question of which order of the proxy databases to select in order to get the most accurate result is an open question that depends on the statistical distributions of the measures involved over the dimensions. We believe that the choice depends on the correlations between the measures, but we have no proof for this. In the example above, if the correlation between *Sales* and *Income* is statistically high, it suggests that it is better to evaluate in the order (*Sales* (*Income* (*Population*))), i.e., get *Income* using *Population* as proxy first, and then get *Sales* using the result *Income* in the previous step as a proxy next. We consider this problem a challenge for future work.

We prove in the next theorem that if the PNC condition holds, the same result will be produced regardless of the order of applying the proxy Databases.

THEOREM 8.1. *Given a primary database and two or more proxy databases, if the dimensions of the proxy databases that are not in the primary database are mutually exclusive (i.e. the PNC condition holds), then the estimated result is invariant under any order of applying the proxy databases.*

PROOF. We present here the sketch of the proof. The formal proof is given in the *electronic appendix*. The proof is based on a specific notation that splits the dimensions of the proxy databases DB_{Q_i} into the dimensions that are common and the dimensions that are non-common with the primary database DB_P. They are represented by $A^{C_P}_{Q_i}$ and $A^{\overline{C}_P}_{Q_i}$, respectively. According to this notation, $M_P(A^C_P, A^{\overline{C}}_P)$ and $M_{Q_1}(A^{C_P}_{Q_1}, A^{\overline{C}_P}_{Q_1}), \ldots, M_{Q_n}(A^{CP}_{Q_n}, A^{\overline{C}_P}_{Q_n})$ are the source summary databases, where $A^C_P = \bigcup_i A^{C_P}_{Q_i}$, and $A^{\overline{C}}_P \neq A^{\overline{C}_P}_{Qi}$ where $0 < i \leq n$. Then, the PNC condition is expressed as $A^{\overline{C}_P}_{Q_i} \cap A^{\overline{C}_P}_{Q_j} = \emptyset$ where $0 < i, j \leq n$ and $i \neq j$.

Given a particular order of the proxy databases, we show in the appendix that applying the F method yields:

$$\begin{aligned}&\hat{M}_P[\mathsf{F}](A^C_P, A^{\overline{C}}_P, A^{\overline{C}_P}_{Q_1}, \ldots, A^{\overline{C}_P}_{Q_n}) \\ &= \frac{M_P(A^C_P, A^{\overline{C}}_P)\left(\hat{M}_{Q_n}[\mathsf{F}](A^{C_P}_{Q_n}, \ldots, A^{C_P}_{Q_1}, A^{\overline{C}_P}_{Q_n}, \ldots, A^{\overline{C}_P}_{Q_1})\right)}{\sum_{A^{\overline{C}_P}_{Q_1}, \ldots, A^{\overline{C}_P}_{Q_n}}\left(\hat{M}_{Q_n}[\mathsf{F}](A^{C_P}_{Q_n}, \ldots, A^{C_P}_{Q_1}, A^{\overline{C}_P}_{Q_n}, \ldots, A^{\overline{C}_P}_{Q_1})\right)}\end{aligned}$$

where:

$$\hat{M}_{Q_n}[\mathsf{F}](A^{C_P}_{Q_n}, \ldots, A^{C_P}_{Q_1}, A^{\overline{C}_P}_{Q_n}, \ldots, A^{\overline{C}_P}_{Q_1}) =$$
$$\left(\left(\ldots\left(\ldots\left(M_{Q_n}(A^{C_P}_{Q_n}, A^{\overline{C}_P}_{Q_n})\frac{M_{Q_{n-1}}(A^{C_P}_{Q_{n-1}}, A^{\overline{C}_P}_{Q_{n-1}})}{\sum_{A^{\overline{C}_P}_{Q_{n-1}}} M_{Q_{n-1}}(A^{C_P}_{Q_{n-1}}, A^{\overline{C}_P}_{Q_{n-1}})}\right)\ldots\right)\ldots\right)\frac{M_{Q_1}(A^{C_P}_{Q_1}, A^{\overline{C}_P}_{Q_1})}{\sum_{A^{\overline{C}_P}_{Q_1}} M_{Q_1}(A^{C_P}_{Q_1}, A^{\overline{C}_P}_{Q_1})}\right)$$

As can be observed, any permutation of the order of applying the proxy databases provide the same result under the PNC condition and therefore can be represented as a closed form formula as follows:

$$\hat{M}_P[\mathsf{F}](A_P^C, A_P^{\overline{C}}, A_{Q_1}^{\overline{C}_P}, \ldots, A_{Q_n}^{\overline{C}_P}) =$$
$$M_P(A_P^C, A_P^{\overline{C}}) \frac{M_{Q_1}(A_{Q_1}^{C_P}, A_{Q_1}^{\overline{C}_P}) \ldots M_{Q_n}(A_{Q_n}^{C_P}, A_{Q_n}^{\overline{C}_P})}{\left(\sum_{A_{Q_1}^{\overline{C}_P}} M_{Q_1}(A_{Q_1}^{C_P}, A_{Q_1}^{\overline{C}_P})\right) \ldots \left(\sum_{A_{Q_n}^{\overline{C}_P}} M_{Q_n}(A_{Q_2}^{C_P}, A_{Q_n}^{\overline{C}_P})\right)} \quad (15)$$

□

8.2 Conditions for the PP Method to Preserve Accuracy

The next question to address is whether the partial pre-aggregation (method PP) can be applied in the case of multiple proxy databases. The problem can be posed as follows: given a target query and a pre-determined evaluation order over multiple proxy databases for generating the full cross product (method F), under what conditions can the PP method be applied and produce results with the same accuracy? We prove in Theorem 8.2 that a sufficient condition is to apply pre-aggregation to non-common dimensions (dimensions that appear in a single database only) and are not required as primary dimensions. In the example above "Type_of_business", and "Education_level" are such dimensions. We also prove that the above definition of non-common dimensions is also a necessary condition. This is achieved by using a counter example in Theorem 8.3. In order to prove this theorem, we use the notations introduced in the following definition.

Definition 8.2. Let $M_P(A_P^C, A_P^{\overline{C}}, A_P^{T^C}, A_P^{T^{\overline{C}}})$ and $M_{Q_1}(A_{Q_1}^C, A_{Q_1}^{\overline{C}}, A_{Q_1}^{T^C}, A_{Q_1}^{T^{\overline{C}}})$, $\ldots$,$M_{Q_n}(A_{Q_n}^C, A_{Q_n}^{\overline{C}}, A_{Q_n}^{T^C}, A_{Q_n}^{T^{\overline{C}}})$ be source summary databases, which are selected by a pre-determined evaluation order for estimating the target summary database $M_P(A_P^{T^C}, A_P^{T^{\overline{C}}}, A_{Q_1}^{T^{\overline{C}}} \ldots, A_{Q_n}^{T^{\overline{C}}})$. The notations $A_P^C = A_P^{C_{Q_1}} \cup A_P^{C_{Q_2}} \cup \ldots \cup A_P^{C_{Q_n}}$ and $A_{Q_i}^C = A_{Q_i}^{C_P} \cup A_{Q_i}^{C_{Q_1}} \cup \ldots \cup A_{Q_i}^{C_{Q_{i-1}}} \cup A_{Q_i}^{C_{Q_{i+1}}} \cup \ldots \cup A_{Q_i}^{C_{Q_n}}$ indicate common dimensions between any combination of the source databases, where $A_P^{C_{Q_i}} = A_{Q_i}^{C_P}$, $A_{Q_i}^{C_{Q_j}} = A_{Q_j}^{C_{Q_i}}$, and $0 < i, j \leq n$, $i \neq j$. Similarly, $A_P^{T^C} = A_P^{T^{C_{Q_1}}} \cup \ldots \cup A_P^{T^{C_{Q_n}}}$, $A_{Q_i}^{T^C} = A_{Q_i}^{T^{C_P}} \cup A_{Q_i}^{T^{C_{Q_1}}} \cup \ldots \cup A_{Q_i}^{T^{C_{Q_{i-1}}}} \cup A_{Q_i}^{T^{C_{Q_{i+1}}}} \cup \ldots \cup A_{Q_1}^{T^{C_{Q_n}}}$ indicate common target dimensions, where $A_P^{T^{C_{Q_i}}} = A_{Q_i}^{T^{C_P}}$, $A_{Q_i}^{T^{C_{Q_j}}} = A_{Q_j}^{T^{C_{Q_i}}}$.

Theorem 8.2. *Given a primary database, and given a particular order of applying two or more proxy databases, the estimated target database using the PP method over non-common dimensions where each non-common dimension is defined as a dimension that exists in a single source database only is as accurate as the target database generated using the F method.*

Proof. We present here the sketch of the proof. The formal proof is given in the *electronic appendix*. Using the definition of the source summary databases according to Definition 8.2, the proof is by induction. In the first step we show that the estimation result of applying method PP is the same as applying method F (i.e. F⇔PP) for the case of two databases: the primary database M_P, and the first proxy database M_{Q_1}. Then we assume that $\hat{M}_P[\mathsf{F}]_i \Leftrightarrow \hat{M}_P[\mathsf{PP}]_i$ is true for

Table VIII. *Income(Education)*

Income	
Education	
High School	8776533572
Bachelor or higher degree	10026459420

Table IX. *Population(State,Age)*

Population	*State*								
Age	AL	CA	FL	NV	MO	NJ	TX	VA	WA
< 25	1761	3765	2040	1225	3885	1316	6634	3566	2967
$25 \div 34$	2070	3017	2830	2026	3004	2355	7451	5735	3559
$35 \div 44$	4468	5781	3698	3518	2781	2465	14995	10694	4235
$45 \div 54$	13742	20368	9528	3855	4192	3168	26280	8181	7310
$55 \div 64$	3186	16498	9981	2947	2701	3100	17010	8412	8387
$65 \div 74$	3756	11544	6921	2297	2346	2435	13624	7130	4856
≥ 75	2256	5611	6751	1824	1323	2298	8595	4472	3156

step i, where the i-th proxy database was applied, and show that for step $i+1$, where the $i+1$-th proxy database is applied, $\hat{M}_P[\mathsf{F}]_{i+1} \Leftrightarrow \hat{M}_P[\mathsf{PP}]_{i+1}$. □

Given the above result, the question arises as to whether pre-aggregating over dimensions that are common to some but not all databases (which we refer to as "partially-common dimensions" below) can possibly yield the same accuracy. This question was not relevant in the case of two databases, but in the case of three or more databases, some dimensions can only be partially common. We show next that aggregating over "partially-common dimensions" does not yield, in general, the same accuracy, thus justifying our choice of defining a non-common dimension as appearing in a single database only.

THEOREM 8.3. *It is necessary to pre-aggregate only over the non-common dimensions of the source summary databases in order to get the same accuracy as the full cross product.*

PROOF. We only need to show by counter example that the result obtained by pre-aggregating over a partially-common dimension is different from the result of the full cross product method. The counter example is presented next.

Let us consider three source summary databases: *Income(Education)*, *Population(State,Age)*, and *Households(Age,Sex)* shown in Tables VIII, IX, and X, respectively. They are used to estimate the target database *Income(State,Sex)*.

We Apply the PP method in two cases: 1) by pre-aggregating over dimensions that appear only in a single database first (in this case, over "education" only), and 2) by pre-aggregating over partially-common dimensions as well (in this case, over "education" as well as over "age"). The results are reported in Table XI.
The above two cases give different results, thus, proving the theorem. □

Table X. *Households(Age,Sex)*

Income	*Sex*	
Age	Male	Female
< 25	13846	13313
25÷34	15398	16649
35÷44	26322	26313
45÷54	52837	43787
55÷64	36795	35427
65÷74	29572	25337
≥ 75	19535	16751

Table XI. $\hat{I}ncome(State,Sex)$

	PP		*Pre-agg. on part.-common dimensions*	
States	*Male*	*Female*	*Male*	*Female*
Alabama	834363165	745134358	825273249	754224275
California	1771737462	1594864001	1759018983	1607582481
Florida	1107615752	1003285433	1102926882	1007974303
Nevada	464448543	430089380	467388019	427149905
Missouri	528817262	494147427	534489848	488474840
New Jersey	449768600	416707569	452726005	413750165
Texas	2500902918	2281679471	2498856280	2283726109
Virginia	1260416488	1176152726	1273085498	1163483716
Washington	906324518	836537918	910629946	832232490

8.3 The Cost of Generating the Cross Product for Multiple Proxy Databases

As discussed above, if the PNC condition is met, the order of performing the cross product is arbitrary. Furthermore, the closed form formula shown in Eq. (15) can be used in order to compute the result. However, if the PNC condition is not met the computational cost may vary with the order of applying the proxy databases, since the intermediate results may vary in the number of cells they have depending on the order chosen.

To illustrate this point, consider again the example above where the three databases are: *Income(Education,Age)*, *Population(Education,Race)*, *Households(Age,State, Race)* and the target measure is *Income*. Note that the PNC condition is not met, because *Race* appears in the two proxy databases and is not a dimension in the primary database *Income*.

Our goal is to generate the cross product from which the target database will be generated after all the pre-aggregation steps take place according to the PP method. To illustrate the generation of the cross product, suppose that the target database we wish to obtain is *Income(State,Education,Age,Race)*. Each successive intermediate result adds dimensions to the cross product. For example, applying the linear indirect estimation method on *Income(Education,Age)* over *Population(Education,Race)*, will generate the intermediate result $\hat{I}ncome(Education,Age, Race)$. Thus, the dimension *Race* was added to the cross product.

The linear indirect estimation method for the intermediate steps is provided below by applying the PP method and using the notation introduced in Definition 8.2:

$$\hat{M}_P[\mathsf{PP}]_{i+1}(A^C_{i+1}, A^{T^C}_{i+1}, A^{T^{\overline{C}}}_{i+1}) = \hat{M}_P[\mathsf{PP}]_i(A^C_i, A^{T^C}_i, A^{T^{\overline{C}}}_i) \frac{M_{Q_{i+1}}(A^C_{Q_{i+1}}, A^{T^C}_{Q_{i+1}}, A^{T^{\overline{C}}}_{Q_{i+1}})}{\sum_{A^{C_{Q_{i+2}}}_{Q_{i+1}}, \ldots, A^{C_{Q_n}}_{Q_{i+1}}, A^{T^{C_{Q_{i+2}}}}_{Q_{i+1}}, \ldots, A^{T^{C_{Q_n}}}_{Q_{i+1}}, A^{T^{\overline{C}}}_{Q_{i+1}}} M_{Q_{i+1}}(A^C_{Q_{i+1}}, A^{T^C}_{Q_{i+1}}, A^{T^{\overline{C}}}_{Q_{i+1}})} \tag{16}$$

where $0 < i \leq n$, $A^C_{i+1} = A^C_i \cup A^C_{Q_{i+1}}$, and $A^{T^C}_{i+1} = A^{T^C}_i \cup A^{T^C}_{Q_{i+1}}$, $A^{T^{\overline{C}}}_{i+1} = A^{T^{\overline{C}}}_{(i)} \cup A^{T^{\overline{C}}}_{Q_{i+1}}$.

Accordingly, the cost formula for the PP method provided in Section 7, but discarding the pre-aggregation and post-aggregation cost is:

$$\left[C_{po}(X^C_{Q_{i+1}} X^{T^C}_{Q_{i+1}} X^{T^{\overline{C}}}_{Q_{i+1}}) + 2\alpha C_{po}(X^C_{i+1} X^{T^C}_{i+1} X^{T^{\overline{C}}}_{i+1})\right]$$

where $X^C_{Q_{i+1}} = X^{C_P}_{Q_{i+1}} X^{C_{Q_1}}_{Q_{i+1}} \ldots X^{C_{Q_n}}_{Q_{i+1}}$, $X^{T^C}_{Q_{i+1}} = X^{T^{C_P}}_{Q_{i+1}} X^{T^{C_{Q_1}}}_{Q_{i+1}} \ldots X^{T^{C_{Q_n}}}_{Q_{i+1}}$, and

$$X^C_{i+1} = X^C_P (X^{C_{Q_2}}_{Q_1} \ldots X^{C_{Q_n}}_{Q_1})(X^{C_{Q_3}}_{Q_2} \ldots X^{C_{Q_n}}_{Q_2}) \ldots (X^{C_{Q_{i+2}}}_{Q_{i+1}} \ldots X^{C_{Q_n}}_{Q_{i+1}})$$

$$X^C_P = X^{C_{Q_1}}_P \ldots X^{C_{Q_n}}_P$$

$$X^{T^C}_{i+1} = X^{T^C}_P (X^{T^{C_{Q_2}}}_{Q_1} \ldots X^{T^{C_{Q_n}}}_{Q_1})(X^{T^{C_{Q_3}}}_{Q_2} \ldots X^{T^{C_{Q_n}}}_{Q_2}) \ldots (X^{T^{C_{Q_{i+2}}}}_{Q_{i+1}} \ldots X^{T^{C_{Q_n}}}_{Q_{i+1}})$$

$$X^{T^C}_P = X^{T^{C_{Q_1}}}_P \ldots X^{T^{C_{Q_n}}}_P$$

$$X^{T^{\overline{C}}}_{i+1} = X^{T^{\overline{C}}}_P X^{T^{\overline{C}}}_{Q_1} \ldots X^{T^{\overline{C}}}_{Q_{i+1}}$$

Since we are generating the entire full cross product (after performing pre-aggregation over all non-common dimensions) all the dimensions are target dimensions, and the above formula reduces to:

$$\left[C_{po}(X^{T^C}_{Q_{i+1}} X^{T^{\overline{C}}}_{Q_{i+1}}) + 2\alpha C_{po}(X^{T^C}_{i+1} X^{T^{\overline{C}}}_{i+1})\right] \tag{17}$$

It consists of two components: the cost of summarizing over the added dimensions for the denominator expression, and the cost of calculating the cells. Thus, it is obvious that it is advantageous to generate as few cells as possible in the intermediate steps. We illustrate this using the example above.

Suppose that the cardinalities of the dimensions are 2, 5, 7, 50 for *Education*, *Age*, *Race*, *State*, correspondingly. We have two choices of generating the result, as shown below. We will calculate the number of cells that we need to compute in each case.

Choice 1: Apply linear indirect estimation method on *Income* over *Population*, and then over *Households*. The number of cells in *Income* is 2x5 = 10. The first step generates $\hat{I}ncome(Education, Age, Race)$.

The result will have 10x7 = 70 cells. According to formula (17), (where for simplicity we assume that $\alpha = 1$) the cost of the first component is 14, and the second component is 2x70 = 140. The total cost for the first step is therefore: 14+140. The number of cells generated by the second step is 70x50 = 3500.

Using the cost formula, one can similarly show that the cost for the second step is: 1750 + 2x3500 =1750 +7000. The total cost for choice 1 is therefore: 14 + 140 + 1750 + 7000 = 8904.

Choice 2: Apply linear indirect estimation method on *Income* over *Households*, and then over *Population*. The first step generates 10 x 350 = 3500 cells, and the second generates 3500 cells. Similar to the above use of the cost formula, we get the cost for step 1: 1750 + 2 x 3500, and for step 2: 14 + 2x3500. The total cost is: 1750 + 7000 + 14 + 7000 = 15764.

The difference of 6860 between the two choices represents a 77% increase in cost over the least expensive choice. It stems from the cost of computing the cross product of the intermediate steps which depends on the order of selecting the proxy databases. Thus, in order to minimize the cost, the proxy databases should be ordered according to the cardinality-product of the dimensions that are added to the intermediate cross products, from lowest to highest.

To summarize, the procedure for using the PP method over multiple databases can be described as follows.

PROCEDURE 8.1. Given a desired target measure and a set of databases

(1) Check that each database has at least one dimension that is requested in the target database. If this condition does not hold, eliminate those databases;
(2) Check that the dimensions of the remaining databases meet the PNC conditions. If so, the closed form formula given in Eq. (15) (see Section 8) can be used in Step 5. Otherwise, since the order for the most accurate result is unknown, the choice of order can be made to minimize the computation cost according to the cardinality-products of the intermediate steps, from lowest to highest.
(3) Aggregate/disaggregate all dimension hierarchy levels (i.e., roll-up or drill-down) according to the target dimensions;
(4) Pre-aggregate all non-common dimensions;
(5) Generate the cross product of all pre-aggregated databases;
(6) Summarize over non-target dimensions.

As was the case for two databases, we note that in the above procedure whenever there is an opportunity to perform aggregation over non-common dimensions and roll-up and/or drill-down operations, they should be performed together to save the cost of generating the intermediate databases.

9. CONCLUSIONS

In this paper, we proposed an efficient method, called the Partial Pre-aggregation method, for estimating the results of a joint query over two source databases us-

ing linear indirect estimation. The proposed method is based on partitioning the dimensions of the source databases into "common", "non-common", and "target" dimensions. By summarizing over the non-common dimensions first, we reduce the computational and space complexity. We proved that the Partial Pre-aggregation method generates results that are as accurate as the Full cross product method commonly used for statistical estimation. Furthermore, we developed computational cost formulas and showed that the PP method can be more efficient by a large factor. Also, we proved formally that "partial pre-aggregation" can be applied together with operations over category hierarchies of dimensions, and developed a procedure for performing "roll-up" and "drill-down" operations with the partial pre-aggregation method to minimize computational costs. In addition, the main results of applying the PP method for two databases were extended to the case of three or more databases.

Using the computational cost and a measure of accuracy, the Average Relative Error, we derived experimental results showing the gain in accuracy of the partial pre-aggregation method relative to full pre-aggregation (the method with the least cost). We showed that the gain in accuracy can be very large, especially when the cardinality-product of the dimensions is high, which is usually the case.

There are several open questions that we believe are important challenges for future work. One is the question of how to select a primary database given that there are multiple databases available with the same measure in order to get the more accurate estimation results. Intuitively, it stands to reason that the one that includes a larger number of the desired target measures is the better choice, but this has to be proven either statistically or perhaps experimentally. Another open problem is the choice of order in the case of multiple proxies in order to maximize accuracy in the case that the condition that determines if the order is irrelevant fails (called the PNC condition - see Section 8). We note that these problems are independent of the use of the partial pre-aggregation method whose purpose is to reduce the computational complexity. If these questions were answered the partial pre-aggregation method can then be applied to the preferred choice of a primary database and the order of applying the proxy databases.

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library. The appendix contains the proofs of several theorems from the main body of this article.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their excellent reviews and insightful comments, which led to many improvements of this paper.

REFERENCES

RTI. http://www.rti.org/page.cfm?nav=497.

Agrawal, R., Gupta, A., and Sarawagi, S. 1997. Modeling multidimensional databases. In *Proceedings of the 13th International Conference on Data Engineering (ICDE),* Birmingham UK, W. A. GRAY and PER-ÅKE LARSON, Eds. IEEE Computer Society Press, Los Alamitos, Calif., 232–243.

Buccafurri, F., Furfaro, F., and Sacca, D. 2001. Estimating range queries using aggregate data with integrity constraints: a probabilistic approach. In *Proceedings of the 8th International Conference on Database Theory (ICDT),* London, UK, JAN VAN den BUSSCHE, V. VIANU, Eds., Lecture Notes in Computer Science, vol. 1973. Springer-Verlag, Heidelberg, Germany, 390–404.

Chan, P. and Shoshani, A. 1981. SUBJECT: A directory driven system for organizing and accessing large statistical databases. In *Proceedings of the 7th International Conference on Very Large Data Bases (VLDB),* Cannes, France, C. ZANIOLO, C. DELOBEL Eds. IEEE Computer Society Press, Los Alamitos, Calif., 553–563.

Chand, N. and Alexander, C. H. 1996. Small area estimation with administrative records and continuous measurement. In *Proceedings of the Survey Research Methods Section, Annual Meeting of the American Statistical Association (ASA),* Chicago, Illinois. 870–875.

Codd, E. F., Codd, S. B., and Salley, C. T. 1993. Providing OLAP (On-Line Analytical Processing) to User-Analysts: an *IT* mandate. E.F. Codd and Associates, Tech. Rep.

Elliott, P., Cuzick, J., English, D., and Stern, R. 1996. *Geographical and environmental epidemiology: Methods for small-area studies.* Oxford University Press, USA.

Faloutsos, C., Jagadish, H. V., and Sidiropoulos, N. D. 1997. Recovering information from summary data. In *Proceedings of 23rd International Conference on Very Large Data Bases (VLDB),* Athens, Greece, M. JARKE, M. J. CAREY, K. R. DITTRICH, F. H. LOCHOVSKY, P. LOUCOPOULOS, M. A. JEUSFELD, Eds. Morgan Kaufmann, San Francisco, Calif., 36–45.

Ghosh, M. and Rao, J. N. K. 1994. Small area estimation: An appraisal. *Statistical Science 9*, 55–93.

Gray, J., Bosworth, A., Layman, A., and Pirahesh, H. 1996. Data cube: a relational aggregation operator generalizing group-by, cross-tabs and sub-totals. In *Proceedings of 12th International Conference on Data Engineering (ICDE),* New Orleans, Louisiana, Stanley Y. W. SU, Ed. IEEE Computer Society Press, Los Alamitos, Calif., 152–159.

Gyssens, M. and Lakshmanan, L. V. S. 1997. A foundation for multi-dimensional databases. In *Proceedings of 23rd International Conference on Very Large Data Bases (VLDB),* Athens, Greece, M. JARKE, M. J. CAREY, K. R. DITTRICH, F. H. LOCHOVSKY, P. LOUCOPOULOS, M. A. JEUSFELD, Eds. Morgan Kaufmann, San Francisco, Calif., 106–115.

Lenz, H.-J. and Shoshani, A. 1997. Summarizability in OLAP and statistical databases. In *Proceedings of 9th International Conference on Scientific and Statistical Data Management (SSDBM),* Olympia, Washington, USA, Y. E. IOANNIDIS and D. M. HANSEN Eds. IEEE Computer Society, Los Alamitos, Calif., 132–143.

Malvestuto, M. F. 1993. A universal-scheme approach to statistical databases containing homogeneous summary tables. *ACM Transactions on Database Systems 18,* 4, 678–708.

Ng, W.-K. and Ravishankar, C. V. 1995. Information synthesis in statistical databases. In *Proceedings of the Fourth International Conference on Information and Knowledge Management (CIKM),* Baltimore, Maryland. ACM Press, New York, NY, USA, 355–361.

Pfeffermann, D. 2002. Small area estimation-new developments and directions. *International Statistical Review 70,* 1, 125–143.

Pourabbas, E. and Shoshani, A. 2003. Answering joint queries from multiple aggregate OLAP databases. In *Data Warehousing and Knowledge Discovery, 5th International Conference (DaWaK),* Y. KAMBAYASHI, M. K. MOHANIA, W. W*öβ* Eds., Lecture Notes Notes in Computer Science, vol. 2737. Springer-Verlag, Berlin, Germany, 24–34.

Rao, J. N. K. 2003. *Small Area Estimation.* John Wiley and Sons, New Jersey.

Schaible, W. L. 1996. Indirect estimators in U.S. federal programs, bureau of labor statistics. *Lecture Notes in Statistics 108*, 1–212.

Shoshani, A. 1997. OLAP and statistical databases: Similarities and differences. In *Proceedings of 16th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (Tucson, Arizona). ACM Press, New York, NY, USA, 185–196.