The derivation problem for summary data

F. M. Malvestuto

(ENEA, Roma - Italy)

ABSTRACT

Given a statistical database consisting of two summary tables based on a common but not identical classification criterion (e.g., two geographical partitionings of a country) there are additional summary tables that are *derivable* in the sense that they are uniquely (i.e., with no uncertainty) determined by the tables given. Derivable tables encompass not only, of course, "less detailed" tables (that is, aggregated data) but also "more detailed" tables (that is, disaggregated data). Tables of the second type can be explicitly constructed by using a "procedure of data refinement" based on the graph representation of the correspondences between the categories of the two classification systems given. In some cases, that is, when such a graph representation meets the *acyclicity* condition, the underlying database is "equivalent" to a single table (called *representative table*) and then a necessary and sufficient condition for a table to be derivable can be stated.

1. INTRODUCTION

We are interested in situations where the statistical information about a certain phenomenon is provided by two distinct data producers, which adopt different classification systems. That is, we have two summary data sets based on the same entity; for example, a fixed geographic region, but with different partitionings.

In order to extract useful information from these two measurements, an integration of the two summary-data sets is required.

This problem of integration is important to national and international organizations that are producers of comprehensive databases, resulting from pooling data collected by statistical agencies or companies [1, 3, 4, 5, 8, 9].

The situation is well summarized by Sato [9] and Johnson [4] as follows:

«A statistical database is a collection of summary data shared by a community of statistical users. If the summary data come from different data sources, seldom there is consensus among two different producers of statistical information about the classification. This makes it difficult to answer queries. Moreover, a user may adopt a new classification system, which differs from all base classifications. Certainly, summary data has been produced from atomic data, so that any summary data can be reproduced as long as the corresponding atomic data is still available. Unfortunately, this situation seldom occurs.» (Sato)

«In many applications users need to manipulate data that has already been aggregated. Populations counts in socio-demographic data bases are examples of such data. It would be unacceptable to require a user to aggregate the raw census data in each and every query. In essence, a summary set represents data that has already been aggregated. There is no need to specify the aggregation procedure or the grouping on which the summarization is based. The aggregate function is specified by the schema and the grouping is determined from the attributes that appear in the output clause.» (Johnson)

Indeed, both Sato and Johnson are concerned with the so-called "matching" problem (or "comparison problem"): two distinct data collecters supply data on two different phenomena (i.e., two statistical variables, say POPULATION and PRODUCTION) and in both cases data are broken down on a geographical scale but according to two different geographic partitionings. On the contrary, the problem here discussed analyses the case where two statistical agencies collect data on the same phenomenon (i.e., only one statistical variable).

Example 1.
For purpose of illustration, consider the data on the FINAL CONSUMPTION OF ELECTRICITY of Italy in 1985 broken down by Industral Sector. These data are available from two alternative data sources: Industry Department of Italy (abbr. IDI) [11] and Organization for Economic Co-operation and Development (abbr. OECD) [12]. The data are heterogeneous because the two sources adopt different units of measure and different classifications of industrial sectors (see Tables 1 and 2).

TABLE 1: FINAL CONSUMPTION OF ELECTRIC ENERGY (10^{12} Kcal) BY INDUSTRIAL SECTOR Italy 1985
(source: IDI)

Sector	*Quantity*
Siderurgico	15.6
Metalli nonferrosi	4.9
Vetro e Ceramica	2.4
Materiali per costruzioni	5.7
Meccanico	11.8
Estrattivo	1.0
Agroalimentare	5.0
Carta e Grafica	4.8
Edile	0.8
Tessile e Abbigliamento	7.9
Chimico	11.5
Petrolchimico	5.2
Altro	3.0
Total	79.6

TABLE 2: FINAL CONSUMPTION OF ELECTRIC ENERGY (Mtoe) BY INDUSTRIAL SECTOR Italy 1985
(source: OECD)

Sector	*Quantity*
Iron & Steel	1.56
Non-ferrous Metals	0.49
Non-metallic Minerals	0.81
Transport Equipment	0.25
Machinery	0.93
Mining & Quarrying	0.10
Food & Tobacco	0.50
Paper, Pulp & Printing	0.48
Construction	0.08
Textile & Leather	0.65
Chemical & Petrolchemical	1.89
Wood & Wood Products	0.18
Industry (non-specified)	0.04
Total	7.96

The heterogeneity due to the different units of measure is not a real problem because it is always possible (and easy) to convert data from one measure system to the other according to the simple formula of conversion: 1 Mtep $\cong 10^{13}$ Kcal.
On the contrary, the diversity of the two classification systems for industrial sectors is not a trivial question.
The major result of this paper is the proof that such summary data sets as Tables 1 and 2 have a "synergetic behavior", which leads to the evaluation of certain distributions more detailed than the distributions composing the database: *database refinement*.

In the above example, we can combine the two tables given and determine some more detailed distributions with no margin of error. For example, the classification of industrial sectors used in the following Table 3 is finer than the two used in Tables 1 and 2.

TABLE 3: FINAL CONSUMPTION OF ELECTRIC ENERGY (Mtoe) BY INDUSTRIAL SECTOR Italy 1985
(sources: IDI, OECD)

Sector	*Quantity*
Iron & Steel	1.56
Non-ferrous Metals	0.49
Glass & Pottery	0.24
Construction-ware	0.57
Transport Equipment	0.25
Machinery	0.93
Mining & Quarrying	0.10
Food & Tobacco	0.50
Paper, Pulp & Printing	0.48
Construction	0.08
Textile & Leather	0.65
Clothing Chemical	0.14
Chemical	1.15
Petrolchemical	0.52
Other Chemical	0.08
Wood & Wood Products	0.18
Industry (non-specified)	0.04
Total	7.96

In Sections 5 and 6 we shall show how the entries of Table 3 have been computed.
The database refinement is accomplished by incorporating into the conceptual database schema the correspondences between categories of two homogeneous classification systems. Unlike other authors [4, 5, 8, 9], who make use of relations (called "correspondence tables" or "comparison tables") to state which category corresponds to which category, we shall represent such correspondence by means of a (labelled) bipartite graph, we call *correspondence graph*. Our approach seems more promising because graphs are endowed with an algebraic structure richer than relations. For example, Sato needs an iterative procedure to define his "binding relation" [9] which is nothing but a "connected component" of our correspondence graph.
Another important result of this paper is that *acyclic* correspondence graphs enjoy the following desirable property: the whole database is equivalent to a single table, called the *representative* table of the database ("perfect refinement" property). As a consequence of this property, we have that with acyclic correspondence graphs we can test the derivability simply by checking whether it is less detailed than the representative table of the underlying database.
The paper is organized as follows.
Section 2 gives basic definitions.
In Section 3 classification systems are viewed as elements of a partition lattice and their correspondence relationships are represented in terms of bipartite graphs. Also the lattice operators of "sum" and "product" are interpreted in graphical terms.
In Section 4 we shall discuss the problem of the

"measurability of an arbitrary partition", that is, the derivability of an arbitrary distribution from the database.
In Section 5 we shall state a theorem that fix the necessary and sufficient condition for the measurability of the product partition. When this condition is satisfied, we say that the underlying database is *perfectly refinable*.
In Section 6 we shall present a way to refine a database, when it is not perfectly refinable (*partial refinement*). The key point is the definition of a graph-theoretical operator, introduced ad hoc and called *mix*, which merges the base partitions in such a way that the resulting partition turns out to be measurable.
In Section 7 we shall discuss the extreme case when a data base can't be refined at all. Section 8 contains some concluding remarks.

2. BASIC DEFINITIONS AND TERMINOLOGY

We are given a collection of *unitary data* relative to a certain *universe* (or *population*) of *statistical unities*. Such unitary data are the result of observation or measurement of some features of interest on the unities of the universe. According to the collection procedure used, the statistical unities are modelled by a set of descriptors, called *attributes,* which may be both quantitative and qualitative.
We may get a synthetic view of the phenomenon under examination by grouping the unitary data in some way. To specify a synthetic view we need a "classification system" and a "measurement variable".

classification system
Usually, the classification procedure can be conceptually regarded as the result of the following three operations.

STEP 1. Cast a certain number of variables as classification criteria. Such variables, referred to as *classification variables* (or *categorical variables* or *category attributes*), may either belong to the set of attributes of the statistical unities of the underlying universe or be new variables functionally dependent on those (that is, computable from those).
STEP 2. Partition the values of each classification variable in a finite number of named groups. Starting from such a partition, it is possible to define a directed-tree structure of meaningful categories, called a "category hierarchy", whose leaves, root and intermediate nodes will be referred to as *categories*, *universal category* and *compound categories,* respectively.
STEP 3. Classify the statistical unities based on the equivalence relation induced by the set of classification variables chosen. Each of the resulting classes is uniquely identified by a categorical vector, which has as many components as the classification variables. The aggregation of two or more classes yields what will be called an *aggregate*. The universe (or *universal aggregate*) can be obtained by choosing for each classification variable the corresponding universal category.

The mapping from the universe to the system of aggregates defined by a set of classification variables is called a *classification* of the universe. In formal terms, a classification is specified by assigning the universe U of objects to classify, a set of classification variables $C_1, \ldots, C_k$ and for each C_i a partition P_i of its value set. A classification will be denoted by $\mathbf{K} = (U, C_1: P_1, \ldots, C_k: P_k)$.

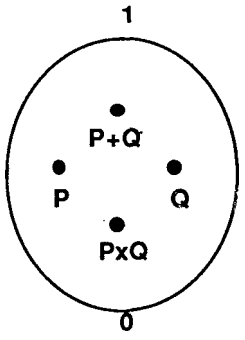
Of course, the level of aggregation or discriminatory power of a classification depends on several factors. Firstly, it depends on how many and which classification variables have been chosen.
Secondly, once chosen the attributes, it depends on how their domains have been partitioned to create the categories.
Now, consider two distinct classifications, say **H** and **K**. If they apply to the same universe and use the same classification variables (e.g., $\mathbf{H} = [U, C_1: P_1, \ldots, C_k: P_k]$ and $\mathbf{K} = [U, C_1: Q_1, \ldots, C_k: Q_k]$), then they are said to be *homogeneous*.

measurement variable
A measurement variable is taken as an additive set function defined on the space of aggregates. Typical examples of aggregate functions are the COUNTing function and the SUMming function (this being based on a quantitative attribute of the individuals of the underlying universe).
Two sources are called *alternative* if the data produced by them refer to two homogeneous classifications and to the same measurement variable.

3. ALGEBRA OF PARTITIONS

We may compare two homogeneous classifications simply by comparing the respective partitions of the domains of each categorical variable. With no loss of generality, we can confine our considerations to the case of only one classification variable, C.
The family of partitions defined on the domain of the values of the classification variable C has the algebraic structure of a lattice based on the relationship of refinement: we say that partition P is *finer* than partition Q (or that Q is *coarser* than P), denoted $P \leq Q$ if each category of Q is a subset of some category of P.
The *zero* (**0**) of the lattice is given by the *point partition*, whose categories are the singleton subsets of the domain. The *unity* (**1**) of the lattice is given by the *trivial partition,* where there exists only one category, i.e., the universal category.
The lattice operation of *product* (sometimes called *join*) of two partitions P and Q, denoted by $P \times Q$, is defined as the coarsest of the partitions that are finer than both P and Q.
The lattice operation of *sum* (sometimes called *meet*) of two partitions P and Q, denoted by $P + Q$, is defined as the finest of the partitions that are coarser than both P and Q.
With regards to the algebraic notions of "greatest lower bound" (glb) and "least upper bound" (lub), we have

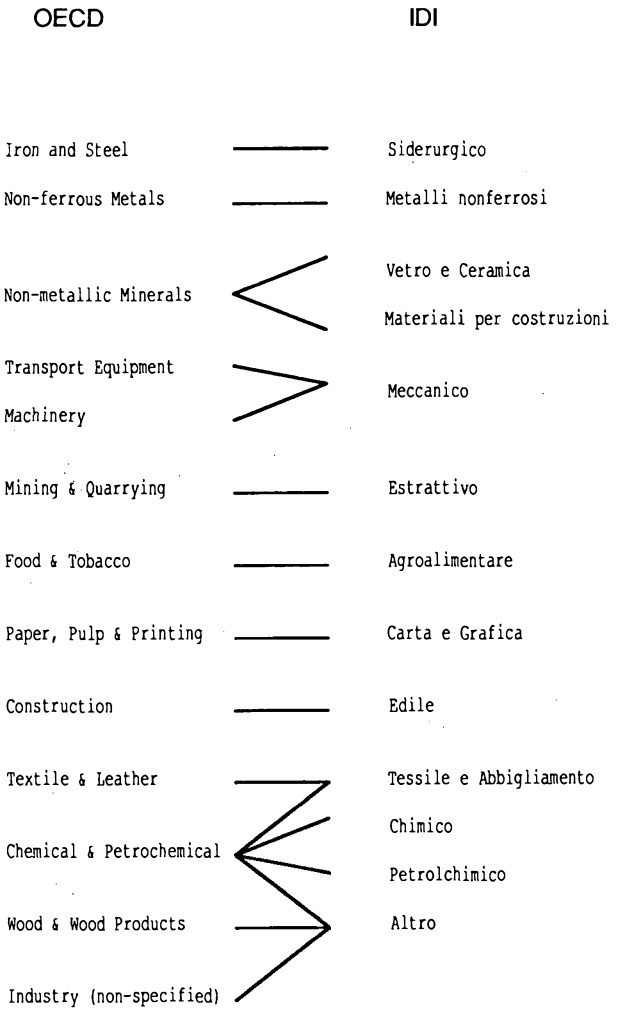
$$\text{glb}(P,Q) = P \times Q \qquad \text{and} \qquad \text{lub}(P,Q) = P + Q$$



The algebraic structure of partitions is shown in figure, where the aggregation level increases in the direction from the point partition (**0**) to the trivial partition (**1**).

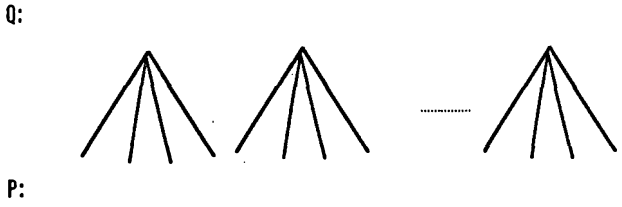
Now, focus on some typical relationships between two arbitrary partitions, P and Q, of the domain of the values of the classification variable C.

Let p and q be categories taken respectively from P and Q. They are said to be *overlapping* if they have a nonempty intersection, that is, if the two aggregates identified by p and q have at least one statistical unity in common. The pairs (p,q) of overlapping categories can be interpreted as the edges of a bipartite graph, G(P,Q), whose two node sets are given by P and Q. Such a graph will be called a *correspondence graph*. Consider the two classifications of industrial sectors of Example 1 again. The correspondence graph is shown below.



The lattice operations of product and sum between partitions have a simple graphical interpretation. The categories of the product partition $P \times Q$ correspond to the edges of G(P,Q). As to the sum partition $P + Q$, its categories coincide with the connected components of G(P,Q).

We can characterize the semantic relationship between two given partitions (that is, between two homogeneous classifications) by analysing the topological properties of the correspondence graph.
Two partitions P and Q are *independent* if G(P,Q) is complete, that is, if all couples (p,q) are overlapping. Otherwise, they are called *dependent*.
Two partitions P and Q are *tree dependent* if G(P,Q) is acyclic, that is, if its connected components are *trees* (a tree is a graph with n nodes and $n - 1$ edges).
A trivial example of tree dependent partitions occurs when partition P is finer than partition Q.



As we saw above, the two classifications of industrial sectors appearing in Tables 1 and 2 give another example of tree dependent partitions.
Acyclic correspondence graphs possess a nice property, which will turn out to be useful later. To state it, we need two additional definitions.
A node is called "pendant" if there is exactly one edge incident to it.
A bipartite graph is called empty if its edge set is empty.

Reducibility of acyclic graphs
A bipartite graph is acyclic (that is, a forest of trees) if and only if it can be reduced to an empty graph by applying repeatedly the following operation of reduction: «if v is a pendant node and is incident to the edge ε, then delete v and ε».

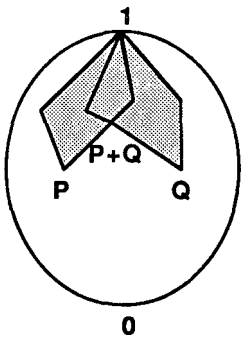
4. THE MEASURABILITY PROBLEM

Consider a database fed by two alternative data sources. This is to say, the stored data refer to the same measurement variable, say x, and are based on two homogeneous classifications. When a user interrogates such a database, the database management system is to be able to decide whether the information required by the user is available. If it is available, then and only then can the query be satisfied. The notion of "available information" encompasses two types of information: "explicit information" (or stored data) and "implicit information" (or derivable data). We focus on this latter type of information.
Of course, the application of any set function to a set of stored data yields something which may be properly called a derived data. However, here we focus on the derivability of distributions, only. In general terms, we are interested in tracing the conditions for a distribution (specified by an arbitrary partition of the classification variable C) to be derivable from the database. To this end, we shall state a formal definition of derivability for distributions in the case of a database consisting of two distributions, $x = f_P(p)$ and $x = f_Q(q)$ based on two homogeneous classifications. The following definition can be easily extended to the case of more than two distributions.

A partition R is called *measurable* if for each R-category r the value taken by the measurement variable x is uniquely determined by the base distributions $x = f_P(p)$ and $x = f_Q(q)$. Then, the distribution $x = f_R(r)$ is called *derivable*.
The spectrum of measurable partitions will be denoted by MEAS(P,Q).
Of course, if R is coarser than P (or Q), then trivially R belongs to MEAS(P,Q) and the distribution $x = f_R(r)$ is derivable:

$$f_R(r) = \Sigma\, f_P(p) \qquad \text{or} \qquad f_R(r) = \Sigma\, f_Q(q)$$

where the summation is extended over all the P-categories (or Q-categories) composing the R-category r.
This already gives some information on the spectrum MEAS(P,Q) of measurable partitions as the following figure shows. The dotted part is included in MEAS(P,Q), but what we are mainly interested in is the lower part, i.e., the part lying below P and Q.



It is evident that no partition in MEAS(P,Q) can be finer than the product of P and Q. So, the membership of the product to MEAS(P,Q) is an interesting question.
Naturally, if P is finer than Q, then trivially the product of P and Q is in MEAS(P,Q) for it coincides with P.
We ask for the conditions for the product of P and Q to be measurable, that is, for a "joint" distribution to be derivable. This case will be referred to as the "perfect refinement" of the database.

5. PERFECT REFINEMENT

Consider a database consisting of two arbitrary distributions, say $x = f_P(p)$ and $x = f_Q(q)$, respectively over the partitions P and Q. Usually, there remains a more or less large set of distributions $x = f(p,q)$ over the product partition $P \times Q$, all of which satisfy the marginal constraints:

$$f_P(p) = \Sigma\, f(p,q) \qquad \text{and} \qquad f_Q(q) = \Sigma\, f(p,q)$$

We say that two distributions, $x = f_P(p)$ and $x = f_Q(q)$, are *joinable* if $P \times Q$ is in MEAS(P,Q), that is, if there is one and only one distribution $f(p,q)$, called *joint distribution*, that satisfies the above marginal constraints. We shall see that in this case there exists a simple procedure for computing $x = f(p,q)$.

THEOREM 1. Let P and Q be two partitions of the same domain. A necessary and sufficient condition for two arbitrary (consistent) distributions, respectively defined over P and Q, to be joinable is that P and Q be tree dependent.
PROOF. Suppose that the correspondence graph G(P,Q) is connected. Otherwise, each connected component will be considered separately. Then the uniquess of the joint distribution is equivalent to the condition that the system of algebraic equations formed by the marginal constraints admits one and only one solution. This algebraic system has n - 1 independent equations, if n is the number of nodes of G(P,Q), and as many unknowns as the edges of G(P,Q). Now G(P,Q) has at least n - 1 edges (n - 1 is the number of edges in a tree, which is the minimally connected graph). From Cramér's theorem it follows that the algebraic system admits one and only one solution if and only if the number of unknowns matches the number of (independent) equations, that is, if and only if G(P,Q) is a tree. QED

So, in case of acyclicity the database is equivalent to the table defined by the joint distribution: case of perfect refinement of the database. We may call this table the *representative* table of the database.
Though the joint distribution might be computed using Cramér's formula, it is more convenient to resort to the following algorithm, based on the reducibility property of acyclic graphs.

Computing x = f(p,q)

STEP 1. If p is a pendant node and (p,q) is the incident edge, then put

$$f(p,q) \leftarrow f_P(p) \qquad \text{and} \qquad f_Q(q) \leftarrow f_Q(q) - f_P(p)$$

Delete both node p and edge (p,q) in G(P,Q).

STEP 2. If q is a pendant node and (p,q) is the incident edge, then put

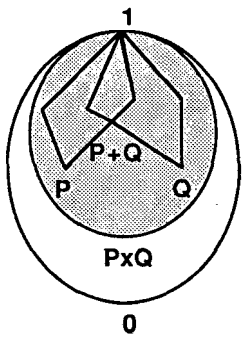
$$f(p,q) \leftarrow f_Q(q) \qquad \text{and} \qquad f_P(p) \leftarrow f_P(p) - f_Q(q)$$

Delete both node q and edge (p,q) in G(P,Q).

STEP 3. If G(P,Q) is empty, then exit. Otherwise, go to step 1.

This algorithm was used in Section 1 to generate Table 3.

To summarize, we can conclude that if the P and Q are tree dependent partitions, then the spectrum MEAS(P,Q) of measurable partitions is that one shown in figure.



Consequently, if we are given an arbitrary distribution, we are able to decide whether it is derivable or not. It is sufficient to check its generating partition is coarser than or equal to the product partition.
When a database is not perfectly refinable, then we can get either a partial refinement or no refinement.

6. PARTIAL REFINEMENT

Suppose now that the base distributions, $x = f_P(p)$ and $x = f_Q(q)$, are not joinable, that is, there exists a set of possible joint distributions. Let R be a partition coarser than the product of the partitions P and Q, on which the base distributions $x = f_P(p)$ and $x = f_Q(q)$ are defined. We know that R is measurable if all joint distributions have the same marginal distribution over R.
A special measurable partition can be obtained by merging the partitions P and Q in the following way.
Consider the correspondence graph G(P,Q). By *blocks* (or "biconnected components") of G(P,Q) we mean its maximal connected subgraphs, each of which is such that the removal of a node (and of the incident edges) is not enough to disconnect it. A block is called *degenerate* if it consists of only one edge; otherwise it is called *nondegenerate*.
The *mix* of P and Q, denoted P * Q, is defined as the partition formed by the blocks of G(P,Q).

Example 2.
Consider the following two geographical partitions P and Q of the 20 Italian regions, we shall index by the lexichographic ordering: Abruzzi (1), Basilicata (2), Calabria (3), Campania (4), Emilia Romagna (5), Friuli Venezia Giulia (6), Lazio (7), Liguria (8), Lombardia (9), Marche (10), Molise (11), Piemonte (12), Puglia (13), Sardegna (14), Sicilia (15), Toscana (16), Trentino Alto Adige (17), Valle d'Aosta (18), Veneto (19), Umbria (20).

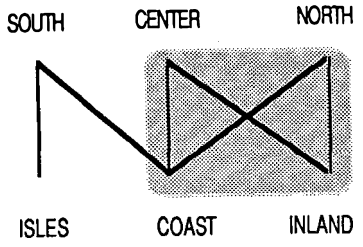
The partition P groups them into the following three geographical categories:

NORTH = {5, 6, 8, 9, 12, 17, 18,19}
CENTER = {7, 10, 16, 20}
SOUTH = {1, 2, 3, 4, 11, 13, 14, 15}

The partition Q groups the Italian regions into the following geographical categories:

ISLES = {14, 15}
COAST = {1, 2, 3, 4, 5, 6, 7, 8, 10,11, 13, 16, 19}
INLAND = {9, 12, 17, 18, 20}

The correspondence graph relative to this pair of partitions is shown below:



The mix T = P * Q turns out to be formed by the following three categories

t1 = {14, 15} (i.e., ISLES)
t2 = {1, 2, 3, 4, 11, 13}
t3 = {5, 6, 7, 8, 9, 10, 12, 16, 17, 18, 19, 20}

It should be noticed that in general the mix of two partitions coincides with the product partition if and only if the two partitions are tree dependent. However, in most cases the mix P * Q is coarser than the product of P and Q, is finer than their sum and need not be comparable with P and Q.
The distribution $x = f_T(t)$ will be called *mixture* of the distributions $x = f_P(p)$ and $x = f_Q(q)$. In order to compute it, we might resort to Cramér's rule. But, there is a more efficient method, based on a graph representation of the mix T.
Given the correspondence graph G(P,Q), let T be the mix of P and Q. Consider the labelled graph G* on the categories of T, that is, on the blocks of G. Two nodes (i.e., T-categories) in G* are joined if the corresponding blocks in G(P,Q) share a node (which labels the edge joining them). The graph G* will called *graph block*.

For the geographic partitions of Example 2, we have the following block graph G*

An important property of block graph is that it is always acyclic. It follows that it is possible to compute the mixture $x = f_T(t)$ by exploiting the reducibility of acyclic graphs.

Computing $x = f_T(t)$

Phase I. *Elimination of pendant blocks*
Consider a pendant block t. Distinguish two cases:

(a) If t is a degenerate block, t = (p,q), then either p or q occurs in no other block. Let it be p. Then, put

$f_T(t) \leftarrow f_P(p)$ and $f_Q(q) \leftarrow f_Q(q) - f_P(p)$.

(b) If t is a nondegenerate block, $t = (\pi, \rho)$, then either π or ρ appears in no other block. Let it be π. Then ρ contains only one Q-node q in common with some other block. Then, put

$f_T(t) \leftarrow f_P(\pi)$ and $f_Q(q) \leftarrow f_Q(\rho) - f_P(\pi)$.

So, remove the block t.

Phase II. *Elimination of isolated blocks*
Consider an isolated block t. Distinguish two cases:

(a) If t is a degenerate block, t = (p,q), then put

$f_T(t) \leftarrow f_P(p)$ or (equivalently) $f_T(t) \leftarrow f_Q(q)$

(b) If t is a nondegenerate block, $t = (\pi, \rho)$, then put

$f_T(t) \leftarrow f_P(\pi)$ or (equivalently) $f_T(t) \leftarrow f_Q(\rho)$

So, remove the block t.

Apply the above algorithm to Example 2, after denoting the categories of P by p1, p2 and p3 , the categories of Q by q1, q2 and q3 , and the categories of T by

$t1 = (p1,q1) = q1$
$t2 = (p1,q2)$
$t3 = \{(p2,q2), (p2,q3), (p3,q2), (p3,q3)\}$

Step 1 (phase I): t3 is a pendant, nondegenerate block with $\pi = \{p2, p3\}$ and $\rho = \{q2, q3\}$. Note that the P-nodes in π appear in no other block, and ρ contains only one Q-node, q2, that occurs in another block. Then, apply the rule I(b):

$$f_T(t3) \leftarrow f_P(p2) + f_P(p3)$$
$$f_Q(q2) \leftarrow f_Q(q2) + f_Q(q3) - f_P(p2) - f_P(p3)$$

Step 2 (phase I): t2= (p1,q2) is a pendant, degenerate block. Then, apply the rule I(b):

$$f_T(t2) \leftarrow f_Q(q2)$$
$$f_P(p1) \leftarrow f_P(p1) - f_Q(q2)$$

Step 3 (phase II): t1= (p1,q1) is an isolated, degenerate block. Then

$$f_T(t1) \leftarrow f_P(p1) \qquad \text{or (equivalently)} \qquad f_T(t1) \leftarrow f_Q(q1)$$

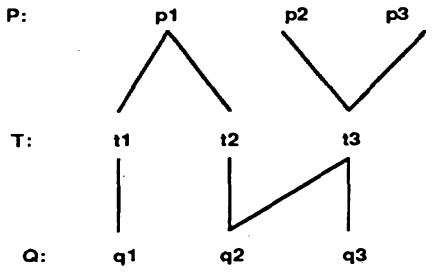
The above algorithm can be viewed as a constructive proof of the following property of mix.

THEOREM 2. The mixture $x = f_T(t)$ of the distributions $x = f_P(p)$ and $x = f_Q(q)$ is always derivable.

Another important property of mix T is the following.

THEOREM 3. The partitions P and T (as well as Q and T) are tree dependent.
PROOF. If G(P,T) were cyclic, then it should contain at least two T-nodes forming a beconnected subgraph. But this contradicts the facts that T-nodes are blocks, that is, maximal biconnected subgraphs of G(P,Q) .

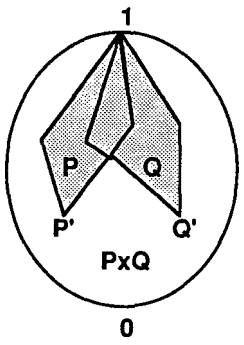
For instance, for the partitions P and Q of Example 2 we have



An immediate consequence of Theorem 3 is that the following two partitions P' and Q'

$$P' = P \times T \qquad \text{and} \qquad Q' = Q \times T$$

are in MEAS(P,Q) , so that the distributions $x = f_{P'}(p')$ and $x = f_{Q'}(q')$ are derivable.



In Example 2, P' is formed by the following categories: q1, (p1,q2), p2, p3 ; and Q' by the following categories: q1, (p1,q2), {(p2,q2), (p3,q2)}, q3 .

It should be noticed that in general one has

$$P' \leq P$$
$$Q' \leq Q$$
$$P' + Q' = P + Q$$
$$P' \times Q' = P \times Q$$

From the relation $P' \leq P$, it follows that the data of the distribution $x = f_P(p)$ can be disaggregated at a lower level. This lowering in aggregation level is the proof of the existence of a "synergy" between the two partitions P and Q . However, in some cases the above procedure results in no refinement of the underlying database, in that both $x = f_{P'}(p')$ and $x = f_{Q'}(q')$ contain no "new" data.
In the next section we shall state the necessary and sufficient conditions for no refinement.

7. NO REFINEMENT

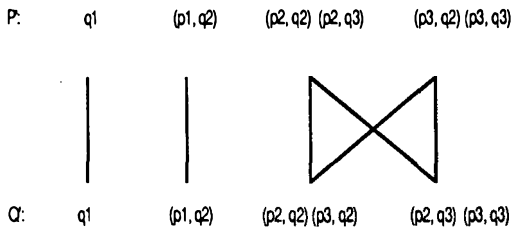
It is clear that the database cannot be refined at all whenever each category of P' and Q' coincides with some category of P or Q . Formally, we say that a database cannot be refined if the following two conditions are satisfied:

$$1)\ \forall\ p' \in P'\ \ (\exists\ p \in P\ \ p' = p)\ \text{ or }\ (\exists\ q \in Q\ \ p' = q)$$
$$2)\ \forall\ q' \in Q'\ \ (\exists\ p \in P\ \ q' = p)\ \text{ or }\ (\exists\ q \in Q\ \ q' = q)$$

Distinguish two cases depending on whether the partitions P and Q are tree dependent. If they are so, then the correspondence graph G(P,Q) is acyclic. Then, the no-refinement condition means each edge contains a pendant node. This assures that no category of $P \times Q$ (i.e., T) simultaneously disaggregates a P-category and a Q-category. Consequently, a connected component of G(P,Q) can be only of one of the following three forms:

It remains to examine the case that P and Q are not tree dependent, that is, when the correspondence graph G(P,Q) is cyclic. In this case, the no-refinement condition means two blocks (no matter if degenerate or nondegenerate) have no common nodes. This assures that no block have the capability of disaggregating the P-nodes and the Q-nodes. Consequently, each connected component of G(P,Q) is a block.

The above-stated conditions of no refinement have the following interesting consequence on the refinement process. Consider the case of a database that can be partially refined. Then, using the above refinement procedure we can construct the partitions P' and Q'. Now, one might hope to further refine P' and Q' by repeating the same procedure that has been applied to P and Q. On the contrary, it is not difficult to check that our refinement procedure can't be applied successfully twice. In fact, consider the correspondence graph G(P',Q') for our running example. It has the following structure



In this graph each connected component is also a block and, therefore, we expect P' and Q' cannot be further refined. This implies can be proved in a direct way by noting that the partition

$$T' = P' * Q'$$

coincides with the partition

$$T = P * Q$$

Hence

$$P'' = P' \times T' = (P \times T) \times T = P \times T = P'$$

and

$$Q'' = Q' \times T' = (Q \times T) \times T = Q \times T = Q'$$

8. CONCLUDING REMARKS

We have presented a way to refine a database consisting of two distributions

$$x = f_P(p) \qquad \text{and} \qquad x = f_Q(q) .$$

The proposed refinement procedure yields two new distributions

$$x = f_{P'}(p') \qquad \text{and} \qquad x = f_{Q'}(q') ,$$

where the generating partitions P' and Q' are finer than P and Q, respectively.
This means that in some cases the query of a user can answered also if the required data are not explicitly contained in the database.
Furthermore, if the partitions P and Q are tree dependent, we are able also to decide whether an arbitrary distribution is derivable or not from the database (it is so if its generating partition is coarser than the product partition).
There remains open the question of deciding the derivability of an arbitrary distribution in the case P and Q are not tree dependent.

REFERENCES

1. C. Chen and P. Hernon, *Numeric databases*, Ablex Publishing Corporation, 1984
2. E. Fortunato, M. Rafanelli, F. Ricci and A. Sebastio, "An Algebra for Statistical Data", *Proc. 3 Int. Workshop on Statistical & Scientific Database Management* 1986, 122-134
3. S. Heiler and A. T. Maness, "Connecting Heterogeneous Systems and Data Sources", Working Group Notes: 2 Int. Workshop on Statistical & Scientific Database Management, in *Database Engineering*, 7:1 (1984) 23-29
4. R. Johnson, "Modelling Summary Data", *Proc. ACM SIGMOD 1981*, 93-97
5. R. Johnson, "A Data Model for Integrating Statistical Interpretations", *TR UCLR-86765* (1981)
6. A. Klug, "Equivalence of Relational Algebra and Relational Calculus Query Languages having Aggregate Functions", *J. ACM* 29:3 (1982) 699-717
7. Z. M. Ozsoyoglu and G. Ozsoyoglu, "An Extension of Relational Algebra for Summary Tables", *Proc. 2 Int. Workshop on Statistical & Scientific Database Management* 1983, 202-211
8. H. Sato, "Handling Summary Information in a Database: Derivability", *Proc. ACM SIGMOD 1981*, 98-107
9. H. Sato, "Fundamental Concepts of Social/Regional Summary Data and Inferences in their Databases", *Thesis Economic Planning Agency, Tokyo* (1982)
10. S.Y.W. Su, "SAM*: a Semantic Association Model for Corporate and Scientific-Statistical Databases", *Information Sciences* 29:2-3 (1983)
11. Ministero dell'Industria, *Bilancio Energetico Nazionale*. Roma,1986
12. OECD, *Energy Balance of OECD Countries*. Paris, 1987