# DEEP: Data Exploration through Extension and Projection

Cliff Joslyn and Susan Mniszewski
**DRAFT**

March, 2002

**Abstract**

We introduce the Data Exploration through Extension and Projection (DEEP) methodology for guided, unsupervised exploration of ordinally- and nominally-supported databases. Databases are defined in the context of General Systems Theory (GST) and General Information Theory (GIT) as multidimensional count (integer) arrays, yielding multidimensional fuzzy relations equipped with operations for projection, (context-dependent) extension, subsetting, and supersetting. Local measures of structure are obtained in terms of compression of support size (relative crisp nonspecificity) and uniformity (group-relative entropy) over the partial ordering of extended state distributions. The methodology is described as alternating, possibly heuristic, application of the database operations to an initial subsetted projection in search of areas of high local structure. An algorithmic approach in terms of paths through the $2^n$ projection space is available. Application to fraud detection for data mining in relational databases is described, as well as implementation in both back-end (program-driven) and front-end (user-driven) processes.
**Keywords:** Data mining, relational theory, possibility theory, entropy, nonspecificity.

## Contents

# 1 Introduction

Data mining methods can be broadly distinguished according to a variety of criteria:

- **Supervised** methods constructs models which identify data as either being similar or dissimilar to some training data, while **unsupervised** methods look for generic structure within databases.
- **Automatic** methods work algorithmically over large datasets to identify interesting regions, whereas **guided** methods work with human interaction using domain-specific knowledge and heuristics.
- Typology of the data is also very important, with various methods working well with **scalar** (cardinal), **ordinal**, or **nominal** (categorical) data.
- **Dimensionality reduction** is a recurring problem in data mining, with methods like clustering and regression introducing new, hybrid dimensional structures onto which data are projected.

In today's vast, heterogenous, high dimensional databases, there is a pressing need for methods which focus the attention of operators, researchers, and managers to relatively small areas (both dimensionally and cardinally) of special interest. Anomaly and fraud detection is an example of such an application, where the goal is to identify a usually very small number of "interesting" data records. A concurrent goal is to deploy methods which respect the original structure of the database, and don't distort its dimensionality too much.

In this paper we present the Data Exploration through Extension and Projection (DEEP) method for largely guided, largely unsupervised exploration of nominally-supported databases. DEEP is intended to find local areas of high structure, while respecting and retaining the original dimensionality of the database. This method is especially appropriate with either nominal data, low-cardinality numeric data, or binned scalar data.

The DEEP approach is rooted in ideas from a number of areas:

- The mathematical approach to General Systems Theory based on mathematical relations [15, 19].
- Modern database theory, especially relational and multidimensional databases [1, 7] as used in OnLine Analytical Processing (OLAP) [6].
- General Information Theory (GIT) [16], which attempts to represent information, uncertainty, and structure in systems using mathematical representations which extend beyond probability measures, including fuzzy sets, relations, and measures.

In particular, we are informed by Klir's General Systems Problem Solver approach to fuzzy-relational inductive modeling [15], and the strong relations that current researchers are making to relational database theory [20, 24] and statistical inference [3, 8]. See also preliminary work by the author [13].

In the first sections we take some care to develop fundamental concepts and notation. While this is mathematically simple material, and somewhat redundant with some well-known results, in this case it will be very valuable, for a number of reasons.

1. In our experience, there are some simple but problematic issues in the formal foundations of relational theory, as compared to working with real databases, or with statistical approaches for nominal data based on multi-dimensional contingency tables or histograms. In particular, what needs to be reconciled is the issue of having multiple data entries per possible state, resulting, for example, in duplicate entries in rows of a data table. Statistical methods address this, but relational theory usually handles this by establishing additional key fields as unique identifiers with functional (deterministic) relations to other fields in the table. We aim to make these issues very clear.
2. To accomplish this, we use somehwat idiosyncratic usage and notation, relying on vector and mathematical bag forms (in addition to set theory based representations) to make explicit use of duplicate entries. We thereby introduce some new concepts of a database bag, state distribution, and extension in context.
3. Finally, we need to introduce a number of ideas from GIT. In particular, a number of our representations and measures of information are based on possibility theory and possibilistic approaches to the representation of data.

One of our overall purposes is to critique the usual application of GIT methods. We hold that typical fuzzy applications based on subjective evaluations need to be complemented by data and measurement-based methods [10]. But at the same time, we hold that possibilistic methods, in particular, must be developed on the basis of random sets, that is, statistical collections of imprecise observations [4, 5, 14]. In this paper we will demonstrate that while measures of possibilistic information in crisp (non-weighted) sources can be very valuable, that the normal methods of deriving possibilistic information from converted probabilities are inadequate [12].

After the introduction of mathematical concepts and notation, we introduce structural measures over extensions (in context) of subsetted database projections. The DEEP method proper is then introduced in terms of the complementary database operations of projection/extension and subsetting/supersetting. These operations, when coupled with the structural measures, allow the identification of local regions of high structure, and a programmatic way for processes (human or automated) to manuever among these regions.

DEEP was initially developed as part of a data mining program to detect fraud in large databases of IRS tax return data. So finally, we explore some of the applications to fraud detection in this context, and also describe both back-end implementations for automatic cross-aggregation analysis, and front-end implementations for user-directed data exploration.

## 2 Mathematical Preliminaries

Denote the natural numbers $\mathbb{N}_M := \{1, 2, \ldots, M\}$ and $\mathbb{N} := \mathbb{N}_\infty$. Similarly, denote the whole numbers $\mathcal{W}_M := \{0, 1, 2, \ldots, M\}$ and $\mathcal{W} := \mathcal{W}_\infty$.

We wish to work with vector notation in a similar way to set notation, including concepts of membership, inclusion, intersection, etc. Denote a vector $\vec{x} := \langle x_1, x_2, \ldots, x_M \rangle = \langle x_i \rangle, M \in \mathcal{W}, i \in \mathcal{W}_M$. Define the length of $\vec{x}$ as $|\vec{x}| := M$. Define the empty vector as $\vec{\emptyset} := \langle\rangle$, such that $|\vec{\emptyset}| = 0$. For a fixed $i \in \mathbb{N}_M$, denote vector-element inclusion as $x \in \vec{x} := \exists i \in \mathbb{N}_M, x = x_i$; vector-element subtraction as

$$\vec{x} - x_i := \begin{cases} \langle x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_M \rangle, & x_i \in \vec{x} \\ \vec{x}, & \text{otherwise} \end{cases};$$

and given another vector $\vec{x}' := \langle x'_{i'} \rangle, i' \in \mathbb{N}_{M'}$ define vector-vector subtraction as $\vec{x} - \vec{x}' := (((\vec{x} - x'_1) - x'_2) - \cdots - x'_{M'})$. Define sub-vector inclusion as $\vec{x} \subseteq \vec{x}' := \forall x_i \in \vec{x}, x_i \in \vec{x}'$. Define vector intersection as $\vec{x} \cap \vec{x}' := \langle x_i \rangle$ such that $x_i \in \vec{x}$ and $x_i \in \vec{x}'$. Define a partition of a vector $\vec{x}$ as a new vector

$$\vec{x}' := \langle \vec{y}_j \rangle, 1 \leq j \leq |\vec{x}'| \leq |\vec{x}| \tag{1}$$

such that $\forall x \in \vec{x}, \exists! \vec{y}_j \in \vec{x}', x \in \vec{y}_j$.

### 2.1 Fuzzy Sets, Relations, and Measures

Assume a set of finite dimensions $\mathcal{X} := \{X_i\}, X_i := \{x^i_{j_i}\}$ for $1 \leq i \leq M, 1 \leq j_i \leq n_i = |X_i|$. Without ambiguity, denote $x^i \in X_i$. The overall **state space** is therefore $X := \times_{i=1}^n X_i$ with $n := |X| = \prod_{i=1}^M n_i$, and individual **state** vectors $\vec{x} := \left\langle x^i_{j_i} \right\rangle \in X$.

We now introduce some basic concepts of GIT [11, 17, 18, 22]. Assume a nonempty, finite set $\Omega = \{\omega_i\}, 1 \leq i \leq n$, and a non-empty subset $A \subseteq \Omega$. The **characteristic function** of $A$ is $\chi_A : \Omega \mapsto \{0, 1\}$ with

$$\chi_A(\omega_i) = \begin{cases} 1, & \omega_i \in A \\ 0, & \omega_i \notin A \end{cases}.$$

$\chi$ can be generalized to define a **membership function** of a **fuzzy set** $\widetilde{A} \widetilde{\subseteq} \Omega$ by $\mu_{\widetilde{A}} : \Omega \mapsto [0, 1]$, where $\mu_{\widetilde{A}}(\omega)$ is the degree or extent to which $\omega_i \in \widetilde{A}$. Each crisp set $A$ is therefore special fuzzy set $\widetilde{A}$ for which $\mu_{\widetilde{A}} = \chi_A$.

We can also identify fuzzy sets on the multidimensional universe $X$. A fuzzy subset of $X$ is called a **fuzzy relation** $\widetilde{R} \widetilde{\subseteq} X$. Thus a normal relation $R \subseteq X$ is also a special fuzzy relation.

A function $\nu: 2^\Omega \mapsto [0,1]$ is a **finite fuzzy measure** if $\nu(\emptyset) = 0$ and $\forall A, B \subseteq \Omega, A \subseteq B \rightarrow \nu(A) \leq \nu(B)$. The **trace** of a fuzzy measure $\nu$ is a function $q^\nu: \Omega \mapsto [0,1]$ where $\forall \omega_i \in \Omega, q^\nu(\omega_i) := \nu(\{\omega_i\})$. $q^\nu$ is also the membership function of a fuzzy set.

Given $\nu$ and $q^\nu$, if in addition there exists an operator function $\oplus: [0,1]^2 \mapsto [0,1]$, with $\oplus$ associative and commutative, such that $\forall A \subseteq \Omega, \nu(A) = \bigoplus_{\omega_i \in A} q^\nu(\omega_i)$, then $\oplus$ is said to be a **distribution operator** of $\nu$ and $q^\nu$ its **distribution**.

$f: \Omega \mapsto [0,1]$ is a **probability distribution** if $\sum_{i=1}^n f(\omega_i) = 1$. $f$ induces a **probability measure** $F: 2^\Omega \mapsto [0,1]$, where $\forall A \subseteq \Omega, F(A) := \sum_{\omega_i \in A} f(\omega_i)$. $F$ is a fuzzy measure with distribution operator $\oplus = +$, and $f = q^+$ its distribution. The traditional properties of probability measures hold, in particular $\forall A, B \subseteq \Omega, F(A \cup B) = F(A) + F(B) - F(A \cap B)$. Also, since $\Omega$ is finite, denote the probability distribution in vector form as $\vec{f} = \langle f_i \rangle := \langle f(\omega_i) \rangle$. The canonical measure of the information content of a probability distribution is the classical stochastic **entropy**

$$\mathbf{H}(\vec{f}) := -\sum_{i=1}^n f_i \log_2(f_i).$$

$\pi: \Omega \mapsto [0,1]$ is a **possibility distribution** if $\vee_{i=1}^n \pi(\omega_i) = 1$, where $\vee$ is the maximum operator. Similar to $f$, $\pi$ induces a **possibility measure** $\Pi: 2^\Omega \mapsto [0,1]$, where now $\forall A \subseteq \Omega, \Pi(A) = \bigvee_{\omega_i \in A} \pi(\omega_i)$. $\Pi$ is a fuzzy measure with distribution operator $\oplus = \vee$, and $\pi = q^\vee$. Unlike probability, possibility measure have the "maxitive" possibilistic property such that $\forall A, B \subseteq \Omega, \Pi(A \cup B) = \Pi(A) \vee \Pi(B)$. Again, since $\Omega$ is finite, denote the possibility distribution in vector form as $\vec{\pi} = \langle \pi_i \rangle := \langle \pi(\omega_i) \rangle$. In possibility theory the **nonspecificity** measure

$$\mathbf{N}(\vec{\pi}) := \sum_{i=2}^n \pi_i \log_2 \left( \frac{i}{i-1} \right) = \sum_{i=1}^n (\pi_i - \pi_{i+1}) \log_2(i),$$

where $\pi_{n+1} = 0$ by convention, is most well justified as the measure of the information content.

## 2.2 Distributions and Information Measures

In this method, we will use vectors of counts extensively. For the moment, define $\vec{c} := \langle c_i \rangle, 1 \leq i \leq n$, where $c_i \in \mathcal{W}$. We assume throughout that $\vec{c}$ is non-vacuous, in other words that $\vec{c} \neq \vec{\emptyset}$.

Fuzzy relational representations of data sets can be very valuable. There are a number of methods available to convert count vectors to fuzzy relations, and to convert amongst different fuzzy relational forms.

Classically, probabilities are derived as relative frequencies according to

$$f_i := \frac{c_i}{\sum_{i=1}^n c_i}, \tag{2}$$

although other formulae are available. Denote the tranform $f(\vec{c}) = \langle f_i \rangle$ according to (2).

The standard method to derive a possibility distribution from either a count or a probability distribution is according to

$$\pi_i = \frac{f_i}{\bigvee_{i=1}^n f_i} = \frac{c_i}{\bigvee_{i=1}^n c_i}. \tag{3}$$

Similarly denote the tranforms $\pi(\vec{c}) = \langle \pi_i \rangle$ and $\pi(\vec{f}) = \langle \pi_i \rangle$ according to (3).

For possibilities, there are also many alternate formulae available, with different justifications [12, 21]. Joslynin particular has criticized (3) as trying to represent essentially probabilistic information in an inappropriate possibilistic format [9]. We will demonstrate below that (3) is, indeed, inadequate, although exploration of other possibilistic formulae will await further work.

For $g \in \{c, f, \pi\}$, define the **core** and **support** as

$$\mathbf{C}(g) := \{\omega_i \in \Omega : g(\omega) \geq 1\}, \quad \mathbf{U}(g) := \{\omega_i \in \Omega : g(\omega) > 0\}.$$

Clearly $\mathbf{C}(g), \mathbf{U}(g) \neq \emptyset$, with $\mathbf{C}(g) \subseteq \mathbf{U}(g)$. If $\mathbf{C}(g) = \mathbf{U}(g)$, then either $g = \pi$ or $g$ is trivial with $|\vec{c}| = 1$. Further, then $g$ is a characteristic function $\chi$. Thus subsets have a natural expression as possibility distributions, but not as probability distributions.

It is well known that $\mathbf{H}$ is maximized at $\mathbf{H}(\vec{f}^*) = \log_2(|\vec{f}^*|)$ for the uniform probability distribution denoted $\vec{f}^* = \left\langle 1/|\vec{f}|^*, 1/|\vec{f}|^*, \ldots, 1/|\vec{f}|^* \right\rangle$; and miminzed at $\mathbf{H}(\vec{f}_*) = 0$ for any $\vec{f}_*$ such that $\exists! i, f_{*i} = 1$. Similarly, $\mathbf{N}$ is maximized at $\mathbf{N}(\vec{\pi}^*) = \log_2(|\vec{\pi}^*|)$ for the "uninformative" possibility distribution denoted $\vec{\pi}^* = \langle 1, 1, \ldots, 1 \rangle$, and minimized at $\mathbf{H}(\vec{\pi}_*) = 0$ for any $\vec{\pi}_*$ such that $\exists! i, \pi_{*i} = 1$. In general, if $\pi$ is crisp then $\mathbf{N}(\pi) = \log_2(|\mathbf{U}(\pi)|)$. Thus the classical information measure of the size of a subset $A \subseteq \Omega$, which is $\log_2(|A|)$, is essentially a (crisp) possibilistic measure of information $\mathbf{N}(\chi_{\mathbf{U}(A)})$.

Furthermore, by (2) and (3), $\pi(\vec{f}^*) = \vec{\pi}^*$, and for a uniform count vector denoted $\vec{c}_0 = \langle c_0, c_0, \ldots, c_0 \rangle$, for some $c_0 \in \mathbb{N}$, $f(\vec{c}_0) = \vec{f}^*$, $\pi(\vec{c}_0) = \vec{\pi}^*$. Note also that then $\mathbf{H}(f(\vec{c}_0)) = \mathbf{N}(\pi(\vec{c}_0)) = \log_2(|\mathbf{U}(\vec{c}_0)|)$.

# 3 Projections and Extensions of Relations

## 3.1 State Spaces

Assume a subset of indices $K \subseteq \mathbb{N}_M$, called a **projector**, and let $\neg K := \mathbb{N}_M - K$. Where possible without ambiguity, denote $K$ as a simple list of its elements, for example $K = 124 := K = \{1, 2, 4\}$, or $K = 10, 11, 12 := K = \{10, 11, 12\}$.

Define the **projection** of the space $X$ through $K$ as $X \downarrow K = X^K := \underset{k \in K}{\times} X_k$, which is the $K$ variables of $X$. Denoting $k' \in \neg K$, then $X \downarrow K$ has corresponding projected state vectors

$$\vec{x} \downarrow K = \vec{x}^K := \left\langle x^k_{j_k} \right\rangle = \vec{x} - \langle x_{k'} \rangle \in X \downarrow K,$$

which are the $K$ variables of the $\vec{x}$.

Note that $|\vec{x}^K| = |K| \leq M$. Also, if $\exists! i' \in \mathbb{N}_M, K = \{i'\}$, then $X \downarrow K$ is just the collection of "singleton" vectors $\left\{ \left\langle x^{i'}_{j_{i'}} \right\rangle \right\}$ for all $x^{i'}_{j_{i'}} \in X_{i'}$. Then just denote $X \downarrow K := X_{i'}$.

More generally, assume two projectors $K, K' \subseteq \mathbb{N}_M$. Then the projection of $X \downarrow K$ again through $K'$ is just $X \downarrow (K \cap K')$. Thus we identify $X = X \downarrow \mathbb{N}_M$, so that $X \downarrow K = X \downarrow (\mathbb{N}_M \cap K)$. Of course if $K \cap K' = \emptyset$ then the projection is empty.

Now consider $X \downarrow (K \cup K')$, and define this as the **extension** to $K'$ of $X \downarrow K$, denoted $(X \downarrow K) \uparrow K'$. Essentially this is just adding the $K'$ variables back in to $X \downarrow K$, so that

$$(X \downarrow K) \uparrow K' := X \downarrow (K \cup K') = (X \downarrow K') \uparrow K.$$

The extended vector

$$(\vec{x} \downarrow K) \uparrow K' := \vec{x} - \langle x_l \rangle, \qquad l \in \neg(K \cup K')$$

is just the sub-vector of $\vec{x}$ containing both the $K$ and the $K'$ variables. Here, if $K' \subseteq K$ then the extension is meaningless.

### 3.2 Relations

Now assume a relation $R \subseteq X$. Then the projection of the relation is very simply

$$R \downarrow K := \{\vec{x} \downarrow K : \vec{x} \in R\} = \{\vec{x}^K \in X \downarrow K : \vec{x}^K \uparrow \neg K \in R\} \subseteq X \downarrow K.$$

Relational extension is a bit more involved. Consider a relation $R \subseteq X$ and two projections $K, K'$. Then consider that we know $R \downarrow K \subseteq X \downarrow K$, and wish to construct the extension to $K'$. It is simple to define

$$(R \downarrow K) \uparrow K' := R \downarrow (K \cup K') = \{\vec{x}' \in (X \downarrow K) \uparrow K' : \vec{x}' \uparrow \neg(K \cup K') \in R\} \subseteq (X \downarrow K) \uparrow K'.$$

But note that this defines $(R \downarrow K) \uparrow K'$ in terms of $R$, the original relation. Therefore we call this **extension in context**. In practice, we might be given only $R \downarrow K$, which in general determines neither $R$ nor $(R \downarrow K) \uparrow K'$. Instead we would normally define only the **cylindrical extension**, denoted

$$(R \downarrow K) \hat{\uparrow} K' := (R \downarrow K) \times \bigtimes_{i \in K'} X_i.$$

In this paper we generally assume that it *is* possible to determine extensions in context.

In English, $X \downarrow K$ is just looking at $X$ without the $\neg K$ dimensions, and $(X \downarrow K) \uparrow K'$ is just adding back in the $K'$ dimensions. Similarly, $\vec{x} \downarrow K$ is just the $\vec{x}$ vector with the $\neg K$ variables removed, and $(\vec{x} \downarrow K) \uparrow K'$ is just adding back in the $K'$ variables. Similarly, the extension in context $(R \downarrow K) \uparrow K'$ is just the distinct vectors from $R$ containing the $K$ variables, and $(R \downarrow K) \uparrow K'$ adds back in the $K'$ variables. By contrast, the cylindrical extension $(R \downarrow K) \hat{\uparrow} K'$ assumes that each $K$-vector of $R \downarrow K$ is actually present in the extension together with *all* the *possible* $K'$-vectors.

More particularly, assume two projections $K, K'$ with $K \subset K'$. Given a particular state $\vec{x}^K \in X \downarrow K$, then define the **state distribution** of $\vec{x}^K$ to $K'$ as

$$\vec{x}^K \Uparrow K' := \{\vec{x}^{K'} \supseteq \vec{x}^K\} = \{\vec{x}^{K'} : \vec{x}^{K'} \downarrow K = \vec{x}^K\} = \{\vec{x}^{K'} : \vec{x}^K \uparrow K' = \vec{x}^{K'}\}. \tag{4}$$

$\vec{x}^K \Uparrow K'$ is thus the set of superstates of $K'$ actually present in $R$ which project through $K$ to yield $\vec{x}^K$.

### 3.3 Example

As an example, let there be $M = 3$ dimensions, with $X_1 := \{a, b, c\}, X_2 := \{x, y, z\}$, and $X_3 := \{\alpha, \beta\}$. Then the state space $X$ is all the ordered triples of $X_1 \times X_2 \times X_3$, and each state $\vec{x}$ is one of those triples.

Assume three projectors $K = \{1\}, K' = \{1, 2\}, K'' = \{1, 3\}$, so that $K' \supset K \subset K''$. The projection of $X$ through $K$ is $X \downarrow K = X_1$, and the extension back to $K'$ is $(X \downarrow K) \uparrow K' = X_1 \times X_2 = X \downarrow 12 = X \downarrow K \cup K'$.

Assume a relation $R := \{\langle a, x, \alpha\rangle, \langle a, y, \alpha\rangle, \langle b, y, \beta\rangle\}$. Then its projection to $K'$ is $R \downarrow 12 = \{x, y\} \subseteq X_2$. The extension of $R \downarrow K$ in context to $K''$ is $(R \downarrow 1) \uparrow 13 = R \downarrow 13 = \{\langle a, \alpha\rangle, \langle b, \beta\rangle\} \subseteq X_1 \times X_3$. But note that the cylyndircal extension is

$$R \downarrow 1) \hat{\uparrow} 3 = (R \downarrow 1) \times X_3 = \{\langle a, \alpha\rangle, \langle a, \beta\rangle, \langle b, \alpha\rangle, \langle b, \beta\rangle\}.$$

Fix a state $\vec{x} = \langle a, x, \alpha\rangle \in R$. Then the projection through $K$ is $\vec{x}^1 = \langle a\rangle$, and the state distribution to $K'$ is $\vec{x}^1 \Uparrow 12 = \{\langle a, x\rangle, \langle a, y\rangle\}$.

# 4 Databases

Define a **data record** as a collection $\vec{\mathbf{X}} := \langle \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N \rangle = \langle \vec{x}_j \rangle, j \in \mathbb{N}_N$, a vector of **observations** $\vec{x}_j \in X$. $\vec{\mathbf{X}}$ is usually represented tabularly as a matrix with $N$ rows (data items) and $M$ columns (fields, dimensions). But since $\vec{\mathbf{X}}$ is a vector, it may have possibly duplicated observations $\vec{x}_i, \vec{x}_{i'} \in \vec{\mathbf{X}}, \vec{x}_i \neq \vec{x}_{i'}$. So it is always possible to define a corresponding multidimensional bag (an unordered collection of multidimensional vectors with duplicates [23]), and from there a multidimensional set (a relation).

Given $\vec{\mathbf{X}}$, define:

- A counting function $c \colon X \mapsto \mathcal{W}_{|\vec{\mathbf{X}}|}$, where $c(\vec{x})$ is the number of times that each record type $\vec{x} \in X$ has been observed in $\vec{\mathbf{X}}$; and the vector form $\vec{c}$;
- The bag $B := \{\langle \vec{x}, c(\vec{x}) \rangle\}$;
- A **support relation** $R \subseteq X$, where

$$\vec{x} \in R \leftrightarrow c(\vec{x}) > 0; \tag{5}$$

- And probability and possibility distributions $f(\vec{c})$ and $\pi(\vec{c})$ given by (2) and (3).

In English, the support relation $R$ consists of those record types which have been observed at least once in the collection $\vec{\mathbf{X}}$. It is also the bag $B$ stripped of both zero count records and all other count information. Note that $\sum_{\vec{x} \in X} c(\vec{x}) = N$.

While the specific information encoded in each of these structures is different, their supports are all equivalent to the support relation.

**Corollary 6** $\mathbf{U}(c) = \mathbf{U}(f) = \mathbf{U}(\pi) = \mathbf{U}(\chi_R) = R$

**Proof:** Follows from inspection of (2), (3), and (5). ■

Also, note that in typical information science applications, it is common to deal with scalar variables, where, for example, $X_i \subseteq \mathbb{R}$. In this case, it would be very rare for $\exists \vec{x} \in X, c(\vec{x}) > 1$. Our method would not be appropriate for such situations.

However, given a scalar variable, constructing a binned, intervally coarsened representation results in an ordinal dimension. Furthermore, scalar vectors are typically represented in relatively low cardinality sets. For example, in financial applications, a quantity of money might be represented as an integer in $[0, 10^6]$. Under these circumstances, this method would be highly appropriate.

## 4.1 Projection and State Distribution

Given a data record $\vec{\mathbf{X}}$ and a projector $K$, then it is natural to define:

- The projection of $\vec{\mathbf{X}}$ through $K$ as $\vec{\mathbf{X}} \downarrow K := \langle \vec{x}_j \downarrow K \rangle$;
- The counting function $c^K \colon X \downarrow K \mapsto \mathcal{W}_{|\vec{\mathbf{X}} \downarrow K|}$, where $c^K(\vec{x}^K)$ is the number of times that each projected record type $\vec{x}^K \in X \downarrow K$ has been observed in the projected data record $\vec{\mathbf{X}} \downarrow K$; and vector form $\vec{c}^K$;
- The bag $B \downarrow K := \left\{ \left\langle \vec{x}^K, c^K(\vec{x}^K) \right\rangle \right\}$;

- The relation $R^K = R \downarrow K \subseteq X \downarrow K := \{\vec{x}^k : c^K(\vec{x}^K) > 0\}$; and
- Probability and possibility distributions $f^K := f(\vec{c}^K)$ and $\pi^K := \pi(\vec{c}^K)$ also given by (2) and (3).

We recognize projections of probabilities and counts as producing marginal distributions. Marginal counts and probabilities are both additive over their state distributions, and thus this additivity is preserved for probabilities derived from projected counts. On the other hand, marginal possibility distributions are defined using the maximum operator. And while this maximality is preserved for support relations (whose characteristic functions are, afer all, possibility distributions) which are derived from additively projected counts, for general possibility distributions this is not the case. This is a fundamental critique of possibility distributions derived from (3).

**Theorem 7** Assume $\vec{\mathbf{X}}$ and $K \subset K'$. Then $\forall \vec{x}^K \in X \downarrow K$

1. $c^K(\vec{x}^K) = \sum_{\vec{y}^{K'} \in \vec{x}^K \Uparrow K'} c^{K'}(\vec{y}^{K'})$
2. $f^K(\vec{x}^K) = \sum_{\vec{y}^{K'} \in \vec{x}^K \Uparrow K'} f^{K'}(\vec{y}^{K'})$
3. $\chi_R^K(\vec{x}^K) = \bigvee_{\vec{y}^{K'} \in \vec{x}^K \Uparrow K'} \chi_R^{K'}(\vec{y}^{K'})$
4. $\pi^K(\vec{x}^K) \neq \bigvee_{\vec{x}^{K'} \in \vec{x}^K \Uparrow K'} \pi^{K'}(\vec{x}^{K'})$

**Proof:**

1. Since $K \subset K'$, therefore $\forall \vec{x} \in X \downarrow K, \vec{y} \in X \downarrow K', \vec{x} \subseteq \vec{y}$. Thus for each $\vec{x}^K$, for the observations $\vec{x}_j \in \vec{\mathbf{X}}$ for which $\vec{x}_j^K = \vec{x}^K$, it must also be that $\exists! \vec{y}^{K'} \in X \downarrow K', \vec{x}_j^{K'} \downarrow K = \vec{y}^{K'}$. Thus the counts $c^K(\vec{x}^K)$ must be completely accounted for in the counts of the $c^{K'}(\vec{y}^{K'})$, and the conclusion follows.

2. From (2) and 1. above,

$$f^K(\vec{x}^K) = \frac{c^K(\vec{x}^K)}{\sum_{\vec{z}^K \in X \downarrow K} c^K(\vec{z}^K)} = \frac{\sum_{\vec{y}^{K'} \in \vec{x}^K \Uparrow K'} c^{K'}(\vec{y}^{K'})}{\sum_{\vec{z}^K \in X \downarrow K} \sum_{\vec{y}^{K'} \in \vec{z}^K \Uparrow K'} c^{K'}(\vec{y}^{K'})} =$$

$$\sum_{\vec{y}^{K'} \in \vec{x}^K \Uparrow K'} \frac{c^{K'}(\vec{y}^{K'})}{\sum_{\vec{w}^{K'} \in X \downarrow K'} c^{K'}(\vec{w}^{K'})} = \sum_{\vec{y}^{K'} \in \vec{x}^K \Uparrow K'} f^{K'}(\vec{y}^{K'})$$

3. $\vec{x}^K \in R^K$ iff $\exists \vec{y}^{K'} \in (\vec{y}^K \Uparrow K'), \vec{y}^{K'} \in R^{K'}$, and so the conclusion follows.

4. For a counterexample, see (11) in Sec. 4.3.

■

## 4.2 Subsetting

Assume a data record $\vec{\mathbf{X}}$, and a subset of indices on the observations $Y \subseteq \mathbb{N}_N$. Then define the **subsetted data record** as $\vec{\mathbf{X}}(Y) := \left\langle \vec{x}_{l_j} \right\rangle \subseteq \vec{\mathbf{X}}, \forall l_j \in Y$. Where possible without ambiguity, denote this as $\vec{\mathbf{Y}}$.

Where projecting reduces the number of columns of the database, subsetting reduces the number of rows. These can be combined, considering the subsetted database $\vec{\mathbf{Y}}$ being projected through the $K$ dimensions. Thus denote a **database** proper as a system $\mathcal{D}(Y, K) := \vec{\mathbf{Y}} \downarrow K$, or just $\mathcal{D}$ without ambiguity.

Given a database $\mathcal{D}$, then it is natural to define:

- The counting function $c^{K,Y}: X \downarrow K \mapsto \mathcal{W}_{|\vec{\mathbf{Y}}\downarrow K|}$, where $c^{K,Y}(\vec{x}^K)$ is the number of times that each projected record type $\vec{x}^K \in X \downarrow K$ has been observed in the database $\vec{\mathbf{Y}} \downarrow K$; and vector form $\vec{c}^{K,Y}$;
- The bag $B^Y \downarrow K := \left\{\left\langle \vec{x}^K, c^{K,Y}(\vec{x}^K)\right\rangle\right\}$;
- The relation $R^{K,Y} \subseteq X \downarrow K := \{\vec{x}^k : c^{K,Y}(\vec{x}^K) > 0\}$; and
- Probability and possibility distributions $f^{K,Y} := f(\vec{c}^{K,Y})$ and $\pi^{K,Y} := \pi(\vec{c}^{K,Y})$ also given by (2) and (3).

Unlike projections, not only are probabilities additively normal, but possibilities are also maximally normal, over subsets.

**Theorem 8** $\forall \vec{x}^K \in \mathcal{D}$,

1. $c^{K,Y}(\vec{x}^K) = \begin{cases} c^K(\vec{x}^K), & \vec{x}^K \in \vec{\mathbf{Y}} \downarrow K \\ 0, & \text{otherwise} \end{cases}$.
2. $f^{K,Y}(\vec{x}^K) = \frac{f^K(\vec{x}^K)}{\sum_{\vec{y}^K \in \mathcal{D}} f^K(\vec{x}'^K)}$.
3. $\pi^{K,Y}(\vec{x}^K) = \frac{\pi^K(\vec{x}^K)}{\bigvee_{\vec{y}^K \in \mathcal{D}} \pi^K(\vec{x}'^K)}$.

**Proof:**

1. Obvious.
2. Because $\vec{x}^K \in \mathcal{D}$, from 1. above

$$f^{K,Y}(\vec{x}^K) = \frac{c^{K,Y}(\vec{x}^K)}{\sum_{\vec{y}^K \in X\downarrow K} c^{K,Y}(\vec{y}^K)} = \frac{c^K(\vec{x}^K)}{\sum_{\vec{y}\in\mathcal{D}} c^K(\vec{y}^K)} = \frac{\frac{c^K(\vec{x}^K)}{\sum_{\vec{z}^K\in\mathcal{D}} c^K(\vec{z}^K)}}{\sum_{\vec{y}^K\in\mathcal{D}} \frac{c^K(\vec{y}^K)}{\sum_{\vec{z}^K\in\mathcal{D}} c^K(\vec{z}^K)}} = \frac{f^K(\vec{x}^K)}{\sum_{\vec{y}^K\in\mathcal{D}} f^K(\vec{y}^K)}.$$

3. By similar reasoning,

$$\pi^{K,Y}(\vec{x}^K) = \frac{c^{K,Y}(\vec{x}^K)}{\bigvee_{\vec{y}^K \in X\downarrow K} c^{K,Y}(\vec{y}^K)} = \frac{c^K(\vec{x}^K)}{\bigvee_{\vec{y}\in\mathcal{D}} c^K(\vec{y}^K)} = \frac{\frac{c^K(\vec{x}^K)}{\bigvee_{\vec{z}^K\in\mathcal{D}} c^K(\vec{z}^K)}}{\bigvee_{\vec{y}^K\in\mathcal{D}} \frac{c^K(\vec{y}^K)}{\bigvee_{\vec{z}^K\in\mathcal{D}} c^K(\vec{z}^K)}} = \frac{f^K(\vec{x}^K)}{\bigvee_{\vec{y}^K\in\mathcal{D}} f^K(\vec{y}^K)}.$$

■

While $Y$ can be specified arbitrarily, in this method we do so only with respect to the states $x^K$ of some projection $X \downarrow K$. In particular, assume a data record $\vec{\mathbf{X}}$, two arbitrary projectors $K, K'$, and a particular state $\vec{x}^K$ in the $K$ space. Then define

$$[\vec{x}^K] := \{j \in \mathbb{N}_N : \vec{x}_j \downarrow K = \vec{x}^K\},$$

so that

$$\mathcal{D}([\vec{x}^K], K') = \left\langle \vec{x}_j^{K'} : \vec{x}_j^{K'} \downarrow K = \vec{x}^K \right\rangle \subseteq \vec{\mathbf{X}} \downarrow K'.$$

In other words, $\mathcal{D}([\vec{x}^K], K')$ is the database derived by considering the projection through the $K'$ variables of all the data vectors which have the value of $\vec{x}^K$ on the $K$ variables.

| | $\vec{x}$ | | | | | |
|---|---|---|---|---|---|---|
| $X_3$ | $X_2$ | $X_1$ | $c(\vec{x})$ | $f(\vec{x})$ | $\pi(\vec{x})$ | $\chi_R(\vec{x})$ |
| $\alpha$ | $x$ | $a$ | 1 | 0.02 | 0.05 | 1 |
| | | $b$ | 0 | 0.00 | 0.00 | 0 |
| | | $c$ | 8 | 0.16 | 0.40 | 1 |
| | $y$ | $a$ | 4 | 0.08 | 0.20 | 1 |
| | | $b$ | 3 | 0.06 | 0.15 | 1 |
| | | $c$ | 0 | 0.00 | 0.00 | 0 |
| $\beta$ | $x$ | $a$ | 8 | 0.16 | 0.40 | 1 |
| | | $b$ | 0 | 0.00 | 0.00 | 0 |
| | | $c$ | 20 | 0.40 | 1.00 | 1 |
| | $y$ | $a$ | 4 | 0.08 | 0.02 | 1 |
| | | $b$ | 2 | 0.04 | 0.10 | 1 |
| | | $c$ | 0 | 0.00 | 0.00 | 0 |

Table 1: Example database bag and distributions.

In particular, if $K \subset K'$, then $\mathcal{D}(\vec{x}^K, K')$ is the database derived by taking just the data records equal to $\vec{x}^K$ on the $K$ variables, and distributing them over the $K'$ variables. Just as (when $K \subset K'$, by (4)) the states $\vec{x}^K$ can be distributed to the $K'$ variables with the state distribution $\vec{x}^K \Uparrow K'$, so the $\vec{x}^K$ can be seen as partitioning the projected and subsetted data record $\vec{\mathbf{Y}} \downarrow K'$ into subvectors consisting of those data records $\vec{x}_j^{K'}$ equal to $\vec{x}^K$ on the $K$ variables. Thus they form a "disjoint cover" in the vector sense.

**Theorem 9** Assume a database $\mathcal{D}([\vec{x}^K], K')$, with $K \subset K'$, and let $\vec{y}^{K'} \in \vec{x}^K \Uparrow K'$. Then the vector of subsetted data records $\left\langle \vec{\mathbf{X}}([\vec{y}^{K'}]) \right\rangle$, paramaterized by the $\vec{y}^{K'}$, is a partition of the subsetted data record $\vec{\mathbf{X}}([\vec{x}^K])$.

**Proof:** Since $\vec{y}^{K'} \in \vec{x}^K \Uparrow K'$, therefore $\vec{y}^{K'} \downarrow K = \vec{x}^K$. Therefore $\forall \vec{x}_j \in \vec{\mathbf{X}}([\vec{x}^K]), \exists \vec{y}^{K'}, \vec{x}_j \in \vec{y}^{K'}$. Furthermore, this is unique, since each $\vec{y}^{K'}$ maps to a unique vector in $X \downarrow K'$. This satisfies (1). ■

**Corollary 10** Under the assumptions from (9), $c^K(\vec{x}^K) = \sum_{\vec{y}^{K'} \in \vec{x}^K \Uparrow K'} c^{K'}(\vec{y}^{K'})$.

**Proof:** Follows immediately from (9). ■

### 4.3 Example

To introduce a similar example to that from Sec. 3.3, let let $N = 50$ and $M = 3$, with $X_1 := \{a, b, c\}, X_2 := \{x, y\}$, and $X_3 := \{\alpha, \beta\}$. Assume a data record $\vec{\mathbf{X}}$ is given, but it is very difficult to show this explicitly. Instead, we derive $\forall \vec{x} \in X, c(\vec{x})$, and use the more efficiently represented database bag $B := \{\langle \vec{x}, c(\vec{x}) \rangle\}$, along with $f(\vec{c}), \pi(\vec{c})$, and $\chi_R(\vec{x})$.

The overall bag and distribution functions are shown in Tab. 1. The projections $\forall \emptyset \neq K \subset \{X_1, X_2, X_3\}$ are then shown in Tab. 2.

As an example of (7), let $K = \{2\}, K' = \{2, 3\}$. Then

$$13 = c^2(\langle y \rangle) = \sum_{\vec{x}^{23} \in \{\langle y,\alpha \rangle, \langle y,\beta \rangle\}} c^{23}(\vec{x}^{23}) = 7 + 6$$

| $\vec{x}^{12}$ | | | | | |
|---|---|---|---|---|---|
| $X_2$ | $X_1$ | $c^{12}$ | $f^{12}$ | $\pi^{12}$ | $\chi_R^{12}$ |
| $x$ | $a$ | 9 | 0.18 | 0.32 | 1 |
| | $b$ | 0 | 0.00 | 0.00 | 0 |
| | $c$ | 28 | 0.56 | 1.00 | 1 |
| $y$ | $a$ | 8 | 0.16 | 0.29 | 1 |
| | $b$ | 5 | 0.10 | 0.18 | 1 |
| | $c$ | 0 | 0.00 | 0.00 | 0 |

| $\vec{x}^{13}$ | | | | | |
|---|---|---|---|---|---|
| $X_3$ | $X_1$ | $c^{13}$ | $f^{13}$ | $\pi^{13}$ | $\chi_R^{13}$ |
| $\alpha$ | $a$ | 5 | 0.10 | 0.25 | 1 |
| | $b$ | 3 | 0.06 | 0.15 | 1 |
| | $c$ | 8 | 0.16 | 0.40 | 1 |
| $\beta$ | $a$ | 12 | 0.24 | 0.60 | 1 |
| | $b$ | 2 | 0.04 | 0.10 | 1 |
| | $c$ | 20 | 0.40 | 1.00 | 1 |

| $\vec{x}^{23}$ | | | | | |
|---|---|---|---|---|---|
| $X_3$ | $X_2$ | $c^{23}$ | $f^{23}$ | $\pi^{23}$ | $\chi_R^{23}$ |
| $\alpha$ | $x$ | 9 | 0.18 | 0.32 | 1 |
| | $y$ | 7 | 0.14 | 0.25 | 1 |
| $\beta$ | $x$ | 28 | 0.56 | 1.00 | 1 |
| | $y$ | 6 | 0.12 | 0.21 | 1 |

| $\vec{x}^{1}$ | | | | |
|---|---|---|---|---|
| $X_1$ | $c^1$ | $f^1$ | $\pi^1$ | $\chi_R^1$ |
| $a$ | 17 | 0.34 | 0.61 | 1 |
| $b$ | 5 | 0.10 | 0.18 | 1 |
| $c$ | 28 | 0.56 | 1.00 | 1 |

| $\vec{x}^{2}$ | | | | |
|---|---|---|---|---|
| $X_2$ | $c^2$ | $f^2$ | $\pi^2$ | $\chi_R^2$ |
| $x$ | 37 | 0.74 | 1.00 | 1 |
| $y$ | 13 | 0.26 | 0.35 | 1 |

| $\vec{x}^{3}$ | | | | |
|---|---|---|---|---|
| $X_3$ | $c^3$ | $f^3$ | $\pi^3$ | $\chi_R^3$ |
| $\alpha$ | 16 | 0.32 | 0.47 | 1 |
| $\beta$ | 34 | 0.68 | 1.00 | 1 |

Table 2: Projected database bags.

$$0.26 = f^2(\langle y \rangle) = \sum_{\vec{x}^{23} \in \{\langle y,\alpha \rangle, \langle y,\beta \rangle\}} f^{23}(\vec{x}^{23}) = 0.14 + 0.12,$$

but

$$0.35 = \pi^2(\langle y \rangle) = \frac{c^2(\langle y \rangle)}{\bigvee_{\vec{x}' \in X \downarrow 23} c^{23}(\vec{x}')} = \frac{13}{37 \vee 13} \neq \bigvee_{\vec{x}^{23} \in \{\langle y,\alpha \rangle, \langle y,\beta \rangle\}} \pi^{23}(\vec{x}^{23}) = 0.25 \vee 0.21 = 0.25. \quad (11)$$

Consider the $\vec{x}^K$ in $\vec{\mathbf{X}} \downarrow K = \{\langle x \rangle, \langle y \rangle\}$. The bags corresponding to the $\mathcal{D}([\vec{x}^K], K')$ are given in Tab. 3. As an example of (9),

$$c^K(\langle x \rangle) = \sum_{\vec{y}^{K'} \in \langle x \rangle \Uparrow K'} c^{K'}(\vec{y}^{K'}) = \sum_{\vec{y}^{K'} \in \{\langle \alpha,x \rangle, \langle \beta,x \rangle\}} c^{K'}(\vec{y}^{K'}) = 9 + 28.$$

# 5 Structural Measures Over Extended Subsets

Assume a database $\mathcal{D}$, a state $\vec{x}^K \in X \downarrow K$, and a new projector $K' \supset K$. Then $\vec{x}^K \Uparrow K'$ represents the actual division within the database of the $c^K(\vec{x}^K)$ data points equal to $\vec{x}^K$ on the $K$ variables to all the vectors $\vec{x}^{K'} \in R^{K'}$ containing the original vector $\vec{x}^K$.

## 5.1 Count Distributions

In particular, for a fixed $\vec{x}^K$, let $C := c^K(\vec{x}^K)$ be the number of records in $\vec{\mathbf{X}}$ equal to $\vec{x}^K$ over the $K$ variables. We are concerned with how these $C$ data records are distributed over the superstates of $\vec{x}^K \Uparrow K'$. Construct the **count distribution** $\vec{C}(\vec{x}^K, K') := \langle C_l \rangle$, where $1 \leq l \leq \gamma := |\vec{C}(\vec{x}^K, K')| \leq$

| | $\vec{x}^{23}$ | | | | |
|---|---|---|---|---|---|
| $X_3$ | $X_2$ | $c^{23,\langle x\rangle}$ | $f^{23,\langle x\rangle}$ | $\pi^{23,\langle x\rangle}$ | $\chi_R^{23,\langle x\rangle}$ |
| $\alpha$ | $x$ | 9 | 0.24 | 0.32 | 1 |
| $\beta$ | $x$ | 28 | 0.76 | 1.00 | 1 |
| | $\vec{x}^{23}$ | | | | |
| $X_3$ | $X_2$ | $c^{23,\langle x\rangle}$ | $f^{23,\langle x\rangle}$ | $\pi^{23,\langle x\rangle}$ | $\chi_R^{23,\langle x\rangle}$ |
| $\alpha$ | $y$ | 7 | 0.54 | 1.00 | 1 |
| $\beta$ | $y$ | 6 | 0.46 | 0.86 | 1 |

Table 3: Subsetted database bags.

$C$, where $\forall C_l \in \vec{C}, \exists \vec{x}^{K'} \in \vec{x}^K \Uparrow K'$ such that $C_l := c^{K'}(\vec{x}^{K'})$ and $C_l > 0$. Where possible without ambiguity, denote $\vec{C} := \vec{C}(\vec{x}^K, K')$.

In other words, $\vec{C}$ is constructed just as a collection of all the positive counts of the superstates in the state distribution of $\vec{x}^K$. Since $\vec{C}$ is determined only to a permutation, if we need to determine $\vec{C}$ uniquely, we will choose it to be decreasing. Given $\vec{C}$, then $\vec{f}$ and $\vec{\pi}$ are available from (2) and (3).

We are therefore concerning ourselves now with a combinatorial problem: given $C$ elements, how can they be divided up over some new category? From (9), we know that $\sum_{l=1}^{\gamma} C_l = C$. But otherwise, in general there are two extremes. On the one hand, they can all be in one state of the new category, so that $\vec{C} = \langle C \rangle$, and $\gamma = 1$. On the other, they can all be in different states of the new category, so that $\vec{C} = \langle 1, 1, \ldots, 1 \rangle$, and $\gamma = C$. Note that $\gamma = |\vec{C}| \in \mathbb{N}_C$ is only *bounded* by $C$, and not *determined* by it.

This problem is equivalent to grouping elements of the partition lattice of $C$ items homomorphically by their size [2, pp. 15-16]. The resulting structure is a poset with unique maximal and minimal elements $\vec{C}^* := \langle C \rangle$ and $\vec{C}_* := \langle 1, 1, \ldots, 1 \rangle$ respectively. Denote this poset as $\Gamma(C)$ for a given $C$, with the partial ordering $\leq$. $\Gamma(8)$ is shown in Fig. 1.

### 5.2 Structural Measures

We are interested in determining the amount of structure represented by a given count distribution $\vec{C} \in \Gamma(C)$ of some initial group of $C$ records. Begin by making a number of explicit assumptions:

1. There are enough states in the $K'$ space that the $C$ observations could be reasonably spread out over them. In particular, $|X \downarrow K'| \gg C$.

2. One would normally expect observations to be reasonably well spread over the $K'$ states.

Given these assumptions, then we can we recognize $\vec{C}^*$ as having maximal structure: all the data points are grouped into the same value in the new category, which is *not* expected. Similarly, we recognize $\vec{C}_*$ as having minimal structure: all the data points are in distinct categories, which *is* expected. So we wish to order the various $\vec{C} \in \Gamma(C)$ by their amount of structure, denoting this as a new ordering $\preceq$.

We need to consider how $\leq$ compares to $\preceq$. In particular, we have $\vec{C}_* \preceq \vec{C}^*$, and also $\vec{C}_* \leq \vec{C}^*$. But from examining the structure of $\Gamma$, it seems unlikely that $\preceq$ should be a linear ordering. As an example, let $C = 8$, and consider

$$\vec{C}_1 = \langle 4, 4 \rangle , \quad \vec{C}_2 = \langle 7, 1 \rangle , \quad \vec{C}_3 = \langle 4, 2, 1, 1 \rangle .$$

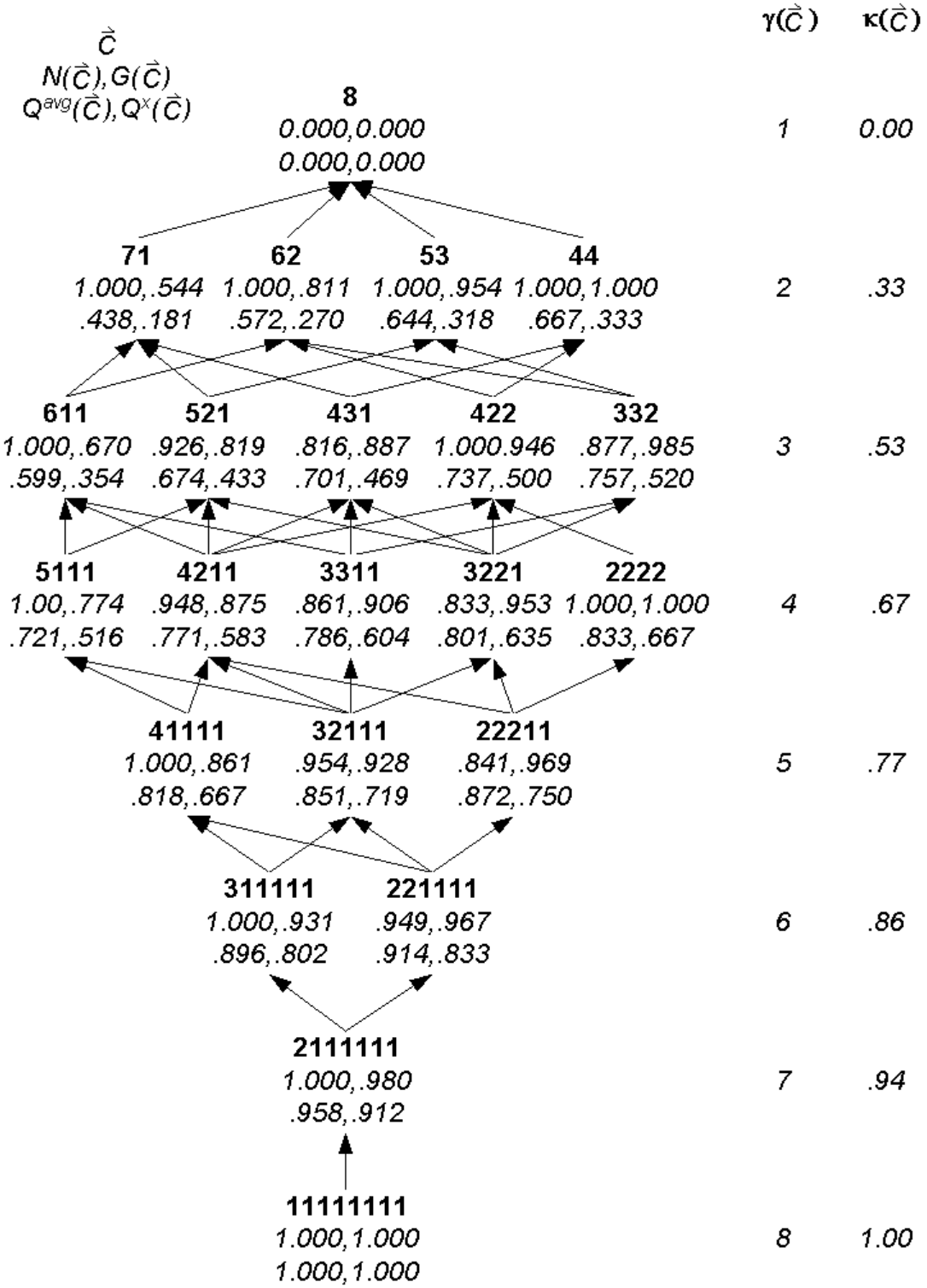


Figure 1: $\Gamma(8)$. Left side: Each node in the poset, showing $\vec{C}, N(\vec{C}), G(\vec{C}), Q^{\text{avg}}(\vec{C})$, and $Q^{\times}(\vec{C})$ according to the key at the upper left. Right side: $\gamma(\vec{C})$ and $\kappa(\vec{C})$ for each level of $\Gamma(8)$.

Clearly $\vec{C}_1 \preceq \vec{C}_2$, because while for both the $C$ counts are split into two groups, they would be expected to be evenly divded as in $\vec{C}_1$, and not as in $\vec{C}_2$. Similarly $\vec{C}_1 \preceq \vec{C}_3$, since we expect the $C$ to be divided into more, rather than fewer, categories. But in both these cases, the sets are noncomparable by $\leq$. But $\vec{C}_2$ and $\vec{C}_3$ appear non-comparable by $\preceq$, since here both factors come into play, despite the fact that $\vec{C}_3 \leq \vec{C}_2$.

Thus we assert that:

1. $\preceq$ is relatively independent of $\leq$.
2. We can recognize in the posets a two dimensional structure. On the one hand, $\Gamma(C)$ has $C$ levels, each corresponding to a different $\gamma \in \mathbb{N}_C$. On the other hand, for a given $\gamma$, there are a variety of possible states, ranging from the maximally to the minimally uniform.

We therefore introduce two independent summary statistics to measure the structure of a given $\vec{C}$:

**Group-Relative Entropy:** Notice that within each $\gamma$ level of $\Gamma(C)$, $\vec{C}$ varies from maximally uniform on the right, to maximally non-uniform on the left. The first measure of $\Gamma(C)$ measure this as an entropy. $f(\vec{C})$, considered as a function of $\vec{C}$, is a probability distribution whose length varies with $\gamma$ between the levels of $\Gamma(C)$. Therefore, we need to define

$$G(\vec{C}) := \frac{\mathbf{H}(f(\vec{C}))}{\log_2(\gamma)} = \frac{\mathbf{H}(f(\vec{C}))}{\log_2(|\vec{C}|)} \tag{12}$$

as a *group*-relative entropy. Note that since $\gamma$ varies from 1 to $C$, the normalizing factor is not a function of the initial count $C$. However, *within* each $\gamma$ level, $G(\vec{C})$ varies within $[0, 1]$, where by convention we assign $G(\vec{C}^*) := 0$. $G(\vec{C}) = 1$ for any uniform count distribution (in other words, once for each $\gamma \neq 1$ which is a factor of $C$).

**Compression:** The other measure of $\Gamma(C)$ is related strictly to the level $\gamma$. Define

$$\kappa(\vec{C}) := \frac{\log_2(\gamma)}{\log_2(C)} = \frac{\log_2(|\vec{C}|)}{\log_2\left(\sum_{l=1}^{|\vec{C}|} C_l\right)}$$

to measure the level $\gamma$ at which $\vec{C}$ exists within $\Gamma(C)$.

We interpret $\kappa(\vec{C})$ as the amount of "compression" which the $C$ elements undergo in their distribution. In other words, if we expect $\vec{C} = \vec{C}_*$, and $|\vec{C}_*| = C$, but $|\vec{C}|$ is actually $\gamma(\vec{C})$, then $\kappa(\vec{C})$ measures the amount of our surprise.

We can also interpret $\kappa(\vec{C})$ as a relative crisp nonspecificity. Let $(\vec{x}^K \Uparrow K')_*$ be any "maximal" state distribution of $\vec{x}^K$ to $K'$, so that $|(\vec{x}^K \Uparrow K')_*| = C$. Then

$$\kappa(\vec{C}) = \frac{\mathbf{N}(\chi_{\mathbf{U}(\vec{x}^K \Uparrow K')})}{\mathbf{N}(\chi_{\mathbf{U}((\vec{x}^K \Uparrow K')_*)})}.$$

Thus $\kappa$ measures the relative crisp nonspecificity of the size of the support of the grouped count distribution to the size of the support of the original count distribution.

Notice that we define $\kappa$ as a log ratio, rather than a simple ratio. The effect, as shown in Fig. 2, for $C = 100$, is to deemphasize smaller amounts of compression.

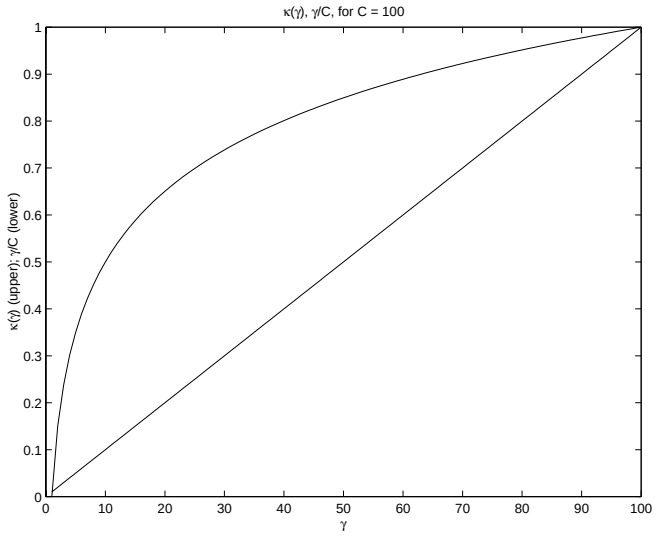


Figure 2: For $C = 100$: above: $\kappa(\vec{C})$; below: $\frac{\gamma}{C}$.

**Group-Relative Nonspecifcity:** Finally, we need to introduce a general (non-crisp) possibilistic measure. Similar to (12), the group-relative nonspecificity for $\pi$ determined by (3) is

$$N(\vec{C}) := \frac{\mathbf{N}(\pi(\vec{C}))}{\log_2(\gamma)}. \tag{13}$$

However, we will demonstrate below that $N$ has no natural interpretation in the context of $\Gamma(C)$, and will therefore deal generally only with $\kappa$ and $G$ below.

Both $\kappa$ and $G$ yield zero values for *high* structure and unity for *low* structure. This is coincident with classical information theory, where high information content or enropy indicates randomness, or lack of structure. Both also yield intermediate values for intermediate degrees of structure, but for structure of different kinds.

**Proposition 14**

1. $\forall \vec{C}, \kappa(\vec{C}), G(\vec{C}), N(\vec{C}) \in [0, 1]$.
2. $\kappa(\vec{C}) = 0 \leftrightarrow G(\vec{C}) = 0 \leftrightarrow N(\vec{C}) = 0 \leftrightarrow \vec{C} = \langle C \rangle$.
3. $\kappa$ is constant for fixed $\gamma$, and increases with $\gamma$.
4. $\kappa(\vec{C}) = 1 \leftrightarrow \vec{C} = \langle 1, 1, \ldots, 1 \rangle$.
5. For fixed $\gamma$, $G$ increases to the right of the diagram of $\Gamma$.
6. $G(\vec{C}) = 1 \leftrightarrow \exists c_0, \vec{C} = \vec{c}_0$.

While we asserted that $\preceq$ is at best a partial ordering with respect to $G$ and $\kappa$, it is not unreasonable to search for a single measure with a linear order over $\Gamma(C)$. Therefore define a hybrid **cross-aggregation measure** $Q^{\oplus}(\vec{C}) := \kappa(\vec{C}) \oplus G(\vec{C})$, where $\oplus$ is some appropriate operator, for example $\oplus \in \{+, \times, \wedge, \vee\}$, or average. Below we will work with $Q^{\text{avg}}$ and $Q^{\times}$.

The values of $\kappa, N, G, Q^{\text{avg}}$, and $Q^{\times}$ are shown for $\Gamma(8)$ in Fig. 1.

A display of $\kappa$ vs. $G$ for $\Gamma(8)$ is shown on the left side of Fig. 3. The solid line traces the $\vec{C}$ through $\Gamma(8)$, from left to right within each $\gamma$ level. Note the level structure of the $\gamma$ with constant $\kappa$ which is revealed, with $G$ moving uniformly within each level. A display of $\kappa$ vs. $N$ for $\Gamma(8)$ is shown on the right side of Fig. 3, with the same solid line trace. Note that the level now is much more confused, revealing no uniform structure.

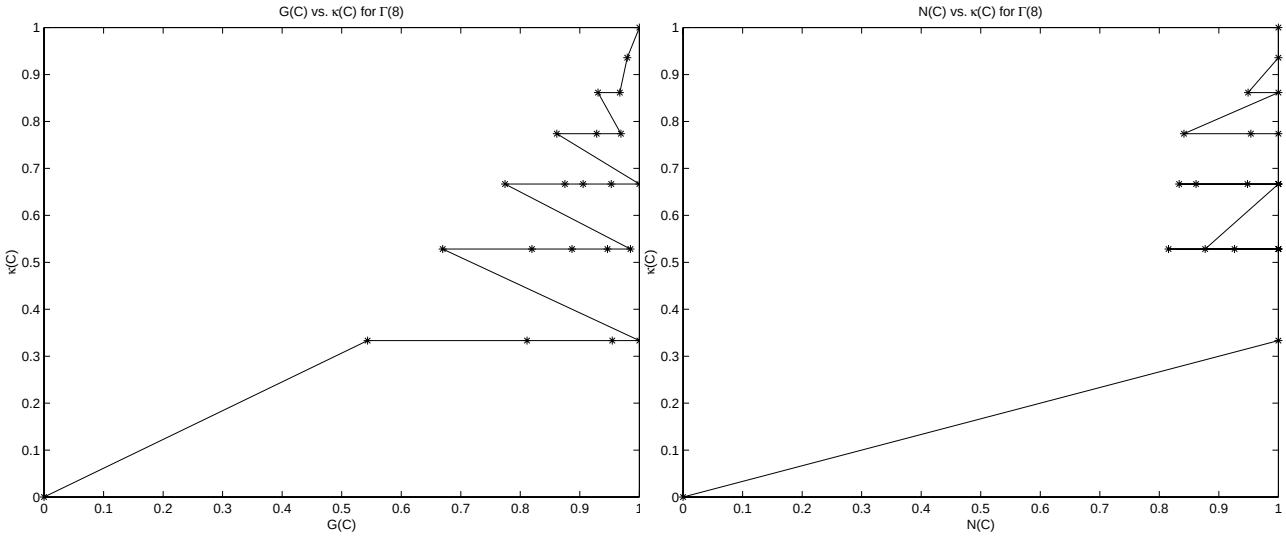


Figure 3: Points correspond to specific elements of $\Gamma(8)$. Left: $G(\vec{C})$ vs. $\kappa(\vec{C})$. Right: $N(\vec{C})$ vs. $\kappa(\vec{C})$.

Other interesting comparisons are shown in Fig. 4, which shows $Q^{\times}(\vec{C})$ and $Q^{\text{avg}}(\vec{C})$ against $\kappa(\vec{C})$ respectively. It is clear that $Q^{\times}$ is superior, although even in this case there no linear ordering results. In particular, $Q^{\times}(\langle 3,3,2\rangle) = .520 > Q^{\times}(\langle 5,1,1,1\rangle) = .516$, despite the fact that while $\langle 3,3,2\rangle$ is slightly more compressed that $\langle 5,1,1,1\rangle$, it is substantially more uniform.

But note that

$$Q^{\times}(\vec{C}) = G(\vec{C})\kappa(\vec{C}) = \frac{\mathbf{H}(f(\vec{C}))}{\log_2(C)},$$

which is the *non-group*, or normal, relative entropy. This is based on the *constant* normalization based on the assumption that $f$ should have the support of size $C$ (one position for each count of $\vec{C}$), rather than the variable $\gamma$ (one position for each group into which the counts of $\vec{C}$ are distributed).

### 5.3 Example

Continuing our example from above, let $K = \{3\}, K' = \{1,3\}$, and $\vec{x} = \langle\alpha\rangle$, so that $\vec{x}^K \Uparrow K' = \{\langle a,\alpha\rangle, \langle b,\alpha\rangle, \langle c,\alpha\rangle\}$. Then

$$\vec{C} = \langle 5,3,8\rangle, \qquad \vec{f} = \langle 0.31, 0.19, 0.50\rangle, \qquad \vec{\pi} = \langle 0.38, 0.63, 1.00\rangle$$

$$C = 16, \quad \gamma = 3, \quad \mathbf{H}(\vec{f}) = 1.48, \quad \mathbf{N}(\vec{\pi}) = 1.44$$

$$\kappa(\vec{C}) = 0.396, \quad G(\vec{C}) = 0.932, \quad N(\vec{C}) = 0.908, \quad Q^{\text{avg}}(\vec{C}) = 0.664, \quad Q^{\times}(\vec{C}) = 0.369.$$

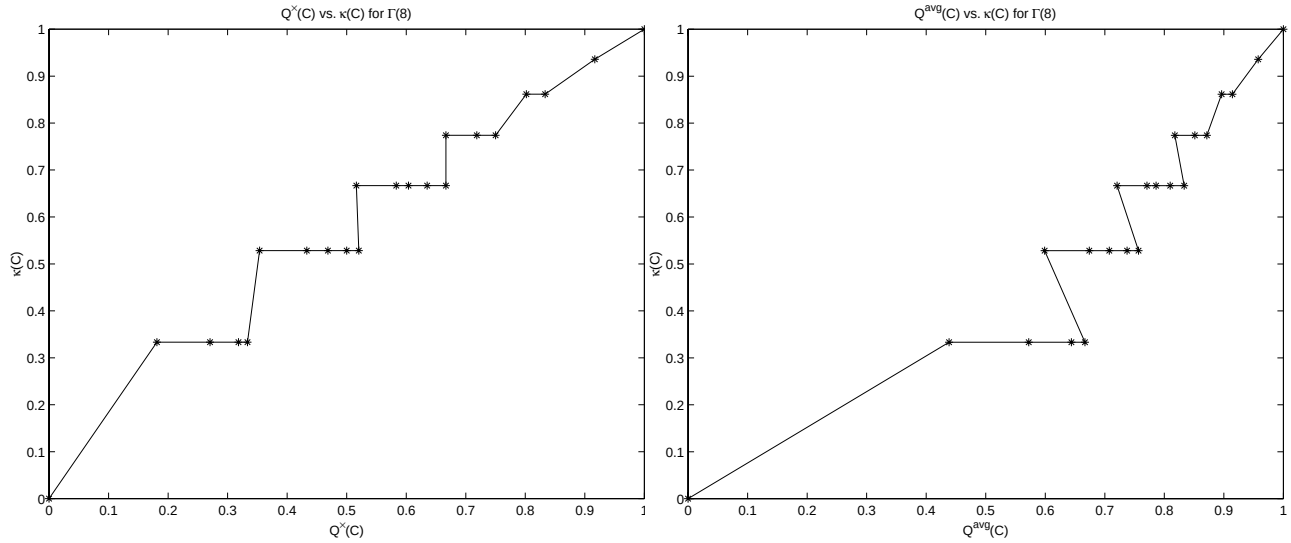


Figure 4: Points correspond to specific elements of $\Gamma(8)$. Left: $Q^{\times}(\vec{C})$ vs. $\kappa(\vec{C})$. Right: $Q^{\text{avg}}(\vec{C})$ vs. $\kappa(\vec{C})$.

# 6 Method

The proposed DEEP method works iteratively at two different levels: "dimensional exploration" of different *projections* of the database; and "cardinal exploration" of different *subsets* of the database. At each step, a database $\mathcal{D}^{t+1}$ is derived from the prior $\mathcal{D}^t$.

## 6.1 Database Operations

Given a database $\mathcal{D} = \mathcal{D}(Y, K)$, and a new projector $K'$, denote the extension of $\mathcal{D}$ to $K'$ as $\mathcal{D} \uparrow K' := \mathcal{D}(Y, K \cup K')$; and the projection through $K'$ as $\mathcal{D} \downarrow K' := \mathcal{D}(Y, K \cap K')$. Similarly, given a new subsetter $Y'$, denote the **restriction** of $\mathcal{D}$ to $Y'$ as $\mathcal{D} \cap Y' := \mathcal{D}(Y \cap Y', K)$; and the **expansion** of $\mathcal{D}$ to $Y'$ as $\mathcal{D} \cup Y' := \mathcal{D}(Y \cup Y', K)$.

Assume a data record $\vec{\mathbf{X}}$, and denote an initial database as $\mathcal{D}^0 := \mathcal{D}(\mathbb{N}_N, \mathbb{N}_M)$. Assume a sequence of time steps $t \in \mathcal{W}$ with an initial time $t = 0$. At each time $t$ assume the existence of a projector $K^t \subseteq \mathbb{N}_M$ and a subsetter $Y^t \subseteq \mathbb{N}_N$. Denote the database at time $t$ as $\mathcal{D}^t := \mathcal{D}(Y^t, K^t)$, or in bag form as

$$B^t := \left\{ \left\langle \vec{x}^{K^t, Y^t}, c^{K^t, Y^t}(\vec{x}^{K^t, Y^t}) \right\rangle \right\}.$$

**Dimensional Exploration:** It may be desirable to derive $\mathcal{D}^{t+1}$ by either projecting or extending $K^t$, leaving $Y^t$ unchanged. Assume a new projector $K'$. There are then three cases:

1. If $K \perp K' := K \cap K' = \emptyset$, then $\mathcal{D}^{t+1} = \mathcal{D}^t \uparrow K'$.
2. If $K \subseteq K'$, then $\mathcal{D}_{t+1} = \mathcal{D}^t \downarrow K'$.
3. Otherwise $K$ and $K'$ "properly overlap", in that $K \not\perp K'$, but also $K \not\subseteq K', K' \not\subseteq K$. Then $\mathcal{D}^{t+1}$ can be either $\mathcal{D}^t \downarrow K$ or $\mathcal{D}^t \uparrow K$.

**Cardinal Exploration:** It may be desirable to derive $\mathcal{D}^{t+1}$ by either restricting or expanding $Y^t$, leaving $K^t$ unchanged. Assume a new subsetter $Y'$. There are again three cases:

1. If $Y \perp Y'$, then $\mathcal{D}^{t+1} = \mathcal{D}^t \cup Y'$.
2. If $Y \subseteq Y'$, then $\mathcal{D}^{t+1} = \mathcal{D}^t \cap Y'$.
3. Otherwise $Y$ and $Y'$ properly overlap, so that $\mathcal{D}^{t+1}$ can be either $\mathcal{D}^t \cup Y'$ or $\mathcal{D}^t \cap Y'$.

At each time $t$, assume that either a subsequent projector $K^{t+1}$ or subsetter $Y^{t+1}$ is provided, and denote the operation performed on the prior database as one of $K^t \downarrow, K^t \uparrow, Y^t \subseteq, Y^t \supseteq$ for projection, extension, restriction, and expansion respectively.

### 6.2 General Method

We have now assembled the tools necessary to apply these concepts in a methodical way. The overall goal is to find localized regions of high structure, that is databases with specific subsets and projections which have low structural and/or cross-aggregation measures. Specifically what is required is to explore various $K$ and $Y$ (specified in terms of various $[\vec{x}^K]$) by applying the operations above.

The central idea is the following. Assume a particular database $\mathcal{D}^t$. We desire to find a particular subset $Y^{t+n} \subseteq Y^t$ of interest for some $n \in wholes$. If we are given a $K^{t+1}$, then for each projected state present in $\mathcal{D}^t$, construct its state distributions extended to $K^{t+1}$, and select that state with minimal structural or cross-aggregation measure. If we are not given a $K^{t+1}$, then it should be selected from all the individual dimensions not present in $K^t$ as that with minimal structural score.

In practice, we will operate with the restriction that $\forall t, K^t = \{k^t\}$ for some $k^t \in \mathbb{N}_M$. In other words, initially we will focus on a single variable, and then move to project or extend only by one variable at a time.

Furthermore, the various $K'$ and $[\vec{x}^K]$ can be derived from multiple sources, depending on the particular context. Sometimes heuristic or expert knowledge directs our attention to certain $K$ and $Y$. Other times a systematic or random search is required. Finally, there are many mixed cases. Furtheremore, expert domain knolwegde may also be needed to interpret the structural measures.

Based on various field typology, the natural distribution of the data, the semantics of the differing fields, and the adherence to the assumptions from Sec. 5.2, high or low values of various measures might either be or not be expected. Finally, implementations might be made in either the front-end of systems, where user selection of $K$ and $Y$ can be allowed, or in the back-end, where pre-programming will be required.

So depending on the particular context, the central idea above can be supplemented by iterated applications of extension and subsetting, or complemented by further actions supersetting and projection.

### 6.3 Examples

We proceed by showing a number of examples.

#### 6.3.1 Programmatic Example

First we illustrate the database operations in a simple example. Let $M = 2, X_1 = \{a, b, c\}, X_2 = \{x, y, z\}$. The contingency table of the $c(\vec{x})$ is shown in Fig. 5. Consider that the series of database operations shown in Tab. 4 is carried out, which is also illustrated in Fig. 5. For each step, Tab. 4 describes the operation, the operand, and the resulting database, in bag form. Note the programmatic nature of this example: given the series of $K^t, Y^t$, these steps can be carried out in an algorithmic manner.

| **Description** | **Operand** | **Result** |
|---|---|---|
| Initial state | | $\mathcal{D}^0 = \mathcal{D}(\mathbb{N}_N, \mathbb{N}_M)$ |
| Project to $X_1$ | $K^1 = \{1\}$ | $\mathcal{D}^1 = \mathcal{D}^0 \downarrow K^1 = \{\langle\langle x\rangle, 22\rangle, \langle\langle y\rangle, 10\rangle, \langle\langle z\rangle, 14\rangle\}$ |
| Subset to $\vec{x}^2 = \langle y\rangle$ | $Y^2 = [\langle y\rangle]$ | $\mathcal{D}^2 = \mathcal{D}^1 \cap Y^2 = \{\langle\langle y\rangle, 10\rangle\}$ |
| Extend back to $\{X_1, X_2\}$ | $K^3 = \{1, 2\}$ | $\mathcal{D}^3 = \mathcal{D}^2 \uparrow K^3 = \{\langle\langle a, y\rangle, 3\rangle, \langle\langle b, y\rangle, 7\rangle\}$ |
| Subset to $\vec{x}^{12} = \langle a, y\rangle$ | $Y^4 = [\langle a, y\rangle]$ | $\mathcal{D}^4 = \mathcal{D}^3 \cap Y^4 = \{\langle\langle a, y\rangle, 3\rangle\}$ |
| Project to $X_2$ | $K^5 = \{2\}$ | $\mathcal{D}^5 = \mathcal{D}^4 \downarrow K^5 = \{\langle\langle a\rangle, 3\rangle\}$ |
| Expand to records matching $\vec{x}^1 = \langle a\rangle$ | $Y^6 = [\langle a\rangle]$ | $\mathcal{D}^6 = \mathcal{D}^5 \cup K^6 = \{\langle\langle a\rangle, 9\rangle\}$ |

Table 4: A programmatic example.

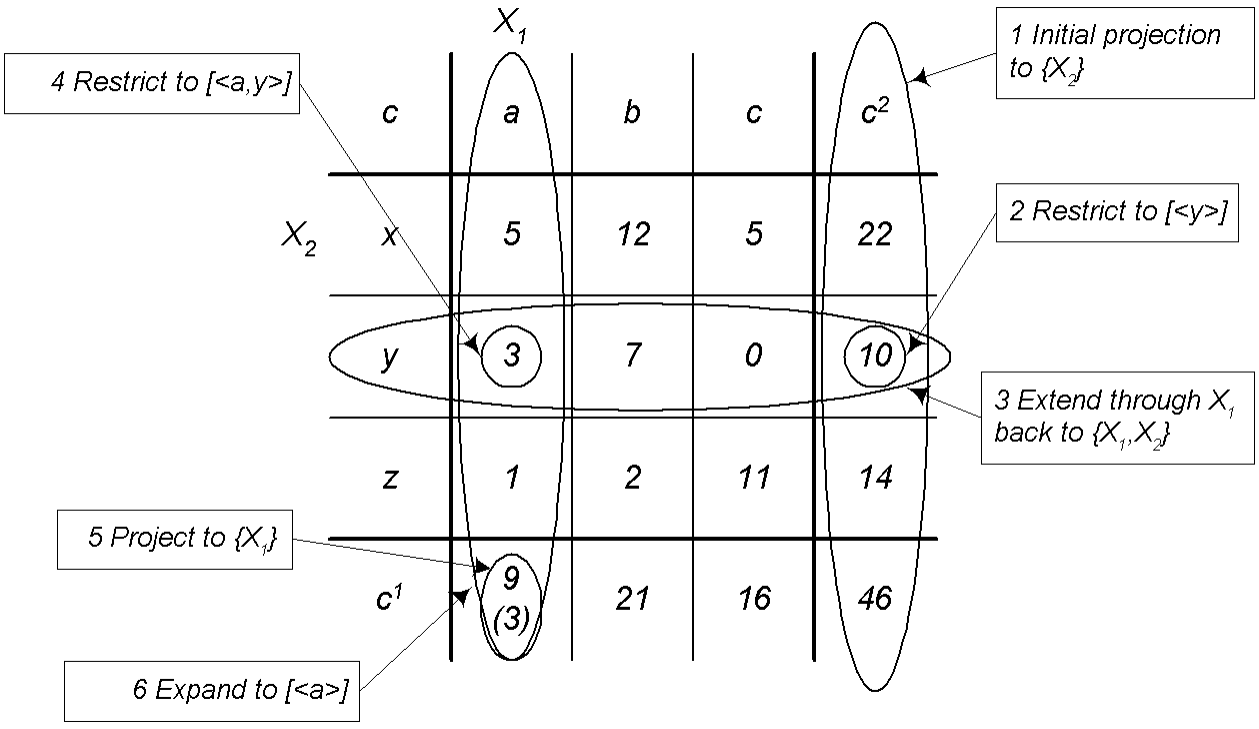


Figure 5: Illustration of the programmatic example.

| $i$ | $\vec{c}^i$ | $G(\vec{c}^i)$ | $\kappa(\vec{c}^i)$ | $Q^{\text{avg}}(\vec{c}^i)$ | $Q^{\times}(\vec{c}^i)$ |
|---|---|---|---|---|---|
| 1 | $\langle 17, 5, 28\rangle$ | 0.839 | 0.281 | 0.560 | 0.236 |
| 2 | $\langle 37, 13\rangle$ | 0.827 | 0.177 | 0.502 | 0.147 |
| 3 | $\langle 34, 16\rangle$ | 0.904 | 0.177 | 0.541 | 0.160 |

Table 5: Initial candidate projections of continuing example.

| $K'$ | $\vec{x}^K$ | $\vec{c}(\vec{x}^K \Uparrow K')$ | $G(\vec{c})$ | $\kappa(\vec{c})$ | $Q^{\text{avg}}(\vec{c})$ | $Q^{\times}(\vec{(c)})$ |
|---|---|---|---|---|---|---|
| $\{1\}$ | $\langle x\rangle$ | $\langle 9, 28\rangle$ | 0.800 | 0.192 | 0.496 | 0.154 |
| | $\langle y\rangle$ | $\langle 8, 5\rangle$ | 0.961 | 0.270 | 0.616 | 0.260 |
| $\{3\}$ | $\langle x\rangle$ | $\langle 9, 28\rangle$ | 0.800 | 0.192 | 0.496 | 0.154 |
| | $\langle y\rangle$ | $\langle 7, 6\rangle$ | 0.996 | 0.270 | 0.633 | 0.269 |

Table 6: Candidate subsets of continuing example.

### 6.3.2 Continuing Example

Our continuing example has been very useful so far, providing a simple illustration of these principles. However, at this point while it is still useful for simple calculations, it does not very accurately reflect real data. That is because:

- The size of the universe of discourse is small, with $n = |X| = 12$. Thus assumption 1 on page 13 is not met. This results in unexpectedly small values of $\kappa$.
- The items are relatively uniformly distributed, resulting in unexpectedly large values of $G$.

Nevertheless, for demonstration purposes it is still worth pursuing this example.

In this case, $\mathcal{D}^0$ is given in Tab. 1, and we proceed by letting the initial projection be for that variable with the minimum cross-aggregation measure, for some $Q^{\oplus}$:

$$K^1 := \arg\min_{X_i \in \mathcal{X}} Q^{\oplus}(\vec{c}^i), \qquad \mathcal{D}^1 := \mathcal{D}^0 \downarrow K^1.$$

The structural measures and the cross-aggregation measure for $\oplus$ = average and $\oplus = \times$ are shown in table Tab. 5. Based on this criteria, we would select $K^1 = \{2\}$, and then construct the extensions of $\vec{\mathbf{X}} \downarrow 2 = \{\langle x\rangle, \langle y\rangle\}$ to both $K' = \{1\}$ and $K' = \{3\}$. These are given in Tab. 6. This would then lead us to select $\vec{x}^1 = \langle x\rangle$ as the first subset.

### 6.3.3 Application Example

DEEP was deployed in the IRS fraud detection project at the Los Alamos National Laboratory. In this case we identified three scalar variables which domain experts indicated were of high interest, which we will denote here as $K = \{1, 2, 3\}$. These were actually integer amounts, and it was feasilble to treat them as nominal variables. We then subsetted groups of records where $G(c^K) = \kappa(c^K) \leq 0.1$, that is, where groups were identical, or nearly so, with respect to the $K$ variables.

While our domain experts indicated that these data points were of interest, and the subsets selected were very small with respect to the entire database, nevertheless there were still both far too many groups and far too many data points to be considered further. Instead, further variables denoted $K' = \{4, 5, 6, 7\}$ were identified, and the $\vec{x}^K$ distributed along the $i \in K'$ individually. Those $\vec{x}^K$ having $\min_{k' \in K'} M(c^{k'})$ were further selected, and proved to be very helpful to our clients.

# 7 Application to Fraud Detection

## 7.1 Supervised: "chaining"

Overlapping entities of orthogonal aggregators

Adjecent, linked entities of the same aggregator

## 7.2 Unsupervised

## 7.3 Variable selection

Preselected

User-guided

## 7.4 Relation to Ping-pong, etc.

# 8 Implementation

The DEEP methodology was deployed in the winter and spring of 1997 in the IRS Fraud Detection Project at the Los Alamos National Laboratory (LANL). This project operates on a large Oracle SQL database jointly between LANL the Cincinatti Service Center (CSC) of the IRS.

Implementation was made in a number of ways:

- In experimental Matlab code;
- Hard-coded into a back-end data analysis module;
- Within an interactive, graphical, data mining environment called VisTool.

As a generator of summary statistics, the DEEP method is, of course, highly sensitive to field selection, subset selection, data typing, and binning of scalar data. All of these must be specified by expert knowledge. This was done by *a priori* selection in the backend implementation, which is described first. The front end implementation is actually much more in keeping with the DEEP methodology, as it allows interactive, user-guided exploration of the spaces of projections and subsets. It will be described later, in the context of a proposed SQL extension to support generation of distributions and cross-aggregation statistics.

## 8.1 Backend Implementation

Referring again to the example described in Sec. 6.3.3, we are reminded that we used three scalar fields $K := \{1, 2, 3\}$, which were identified as nominal variables by being treated as integers. These were then supplemented by additional fields $K' := \{4, 5, 6, 7\}$, which were true nominals to begin with.

Code was written in the ProC SQL code-generation language to construct a hybrid C/SQL program for analysis of data as it entered the system. Inputted records were projected into $K$, and the bag $B^K$ constructed and stored in Oracle tables. For each entry ${}^k\vec{x} \in B^K$, the state distributions ${}^k\vec{x} \Uparrow K \cup \{i'\}$ were then constructed $\forall i' \in K'$. Denoting $\vec{x}^K := {}^k\vec{x}$, the values of

$$c^K(\vec{x}^K), \qquad \vec{C}^{K'} := \vec{C}(\vec{x}^K, K'), \qquad \kappa(\vec{C}^{K'}), \qquad G(\vec{C}^{K'}), \qquad Q^{\text{avg}}(\vec{C}^{K'}), \qquad Q^{\wedge}(\vec{C}^{K'})$$

were then also stored in the database.

In this trial, various experimental combinations of values were explored, in an attempt to find small subsets of records with a high level of interest. These groups were then automatically made available to the clients. Experimental variation included:

- Finding an appropriately low values of $Q^\wedge$ which would prove interesting.
- Experimenting with $K \subseteq \{1, 2, 3\}$.
- Experimenting with various ways in which $K' \cap K \neq \emptyset$, either by putting elements of $K$ in $K'$, or *vice versa*.

## 8.2 SQL Extension and VisTool Spreadsheets

We now describe the front-end implementation by introducing a proposed extension to SQL to allow the creation of distributions and cross-aggregation statistics on them. Complete extended queries are of the form:

SELECT [ $K^S$, ] [ $agg(K^A)$, ] [ DIST($K_1^D$), DIST($K_2^D$), ...]
FROM $T$
WHERE $C$
GROUP BY $K^G$

In the following subsections, we will build up this query form incrementally, at each stage showing the relation to our database notation, and how they are implemented in VisTool spreadsheets.

### 8.2.1 Non-Grouped Queries

There are one to one mappings among the non-extended, non-grouped SQL queries, our database notation, and VisTool spreadsheets.

Assume a query "SELECT $K^S$ FROM $T$ WHERE $C$", where

- $T = \{t\}$ is a set of table names, and its use in the query is to be interpeted as the insertion of that list of table names;
- dom($t$) is a list of the field names in $t$, corresponding to our dimensions $X_i$;
- The overall set of available fields is $\mathcal{X} := \cup_{t \in T} \text{dom}(t)$, with $M := |\mathcal{X}|$;
- Each of the $K^\cdot$ is a projector from $\mathcal{X}$, and their use in the query is to be interpreted as the insertion of the corresponding field names;
- $K^S$ is called the **selection projector**;
- Denote $K := K^S$ (this will be useful in another way below);
- $C$ is a condition of the form $exp(K^C)$, where:
  - *exp* is some appropriate expression; and
  - $K^C \subseteq \mathbb{N}_M$.

Construct $\mathcal{D} = \mathcal{D}(Y, K) = \vec{\mathbf{Y}} \downarrow K$ as follows:

- The data record is $\vec{\mathbf{X}} = \langle \vec{x}_i \rangle$, where each $\vec{x}_i$ is a line of the output of the query "SELECT * FROM $T$";

- $M$ is the result of the query "SELECT COUNT(*) FROM $T$";
- $Y$ is the set of indices in $\mathbb{N}_M$ of those rows whose $\vec{x}_j$ satisfy $C$, and $\vec{\mathbf{Y}}$ is the corresponding sub-vector of $\vec{\mathbf{X}}$;
- And $K$ is defined above.

Construct the VisTool spreadsheet as follows:

- There will be one column for each $i \in K^S$;
- There will be one row for each $j \in Y$;
- The $\vec{x}_j$ are displayed in the rows.

Call this form of spreadsheet a **detailed spreadsheet**.

### 8.2.2 Grouped Queries

Now extend this to a standard grouped query of the form:

SELECT $K^S$, $agg(K^A)$
FROM $T$
WHERE $C$
GROUP BY $K^G$

where now:

- $agg$ is an aggregation function, such as SUM, AVG, MIN, MAX, etc.;
- $K^A$ is called the **aggregation projector**, with $K^S \cup K^A \neq \emptyset$;
- $K^G$ is called the **grouping projector**, with $K^G \supseteq K^S$;
- And now denote $K := K^S \cup K^G = K^G$.

$\mathcal{D}$ is modified appropriately as $K$ is here.

For the VisTool spreadsheet, call any spreadsheet generated by a GROUP BY expression an **aggregated spreadsheet**, constructed as follows:

- Where a detailed spreadsheet is defined on the data record $\vec{\mathbf{Y}}$, and thus has one row for each $j \in Y$, an aggregated spreadsheet is defined on the data bag $B^Y \downarrow K$, and thus has one row for each $k \in \mathbb{N}_Q$, where we are reminded that in the data bag

$$1 \leq k \leq Q := |B^Y \downarrow K| = |R^{K,Y}| \leq N.$$

  Thus one line of output is produced for each distinct tuple $\vec{x}^K$ of the projector $K$ in the join of the $t \in T$.

- In the aggregated spreadsheet, there is one column for each $i \in K^S$ and distinctly for each $i \in K^A$, where the aggregated data are displayed.

| $\vec{x}^{23}$ | | | | | | |
|---|---|---|---|---|---|---|
| $X_3$ | $X_2$ | $c^{23}$ | $\vec{C}$ | $\kappa(\vec{C})$ | $G(\vec{C})$ | $Q^{\times}(\vec{C})$ |
| $\alpha$ | $x$ | 9 | $\langle 8,1\rangle$ | 0.316 | 0.503 | 0.159 |
| | $y$ | 7 | $\langle 4,3\rangle$ | 0.356 | 0.985 | 0.351 |
| $\beta$ | $x$ | 28 | $\langle 20,8\rangle$ | 0.208 | 0.863 | 0.180 |
| | $y$ | 6 | $\langle 4,2\rangle$ | 0.387 | 0.918 | 0.355 |

Table 7: Example extended SQL output.

### 8.2.3 Extended, Grouped Queries

Finally, continue to an extended group query using a proposed extension to SQL to allow the creation of distributions and cross-aggregation statistics on them. The extension is of the form:

SELECT $K^S$, $agg(K^A)$, DIST($K_1^D$), DIST($K_2^D$), ...
FROM $T$
WHERE $C$
GROUP BY $K^G$

where the $K_i^D$ are projectors called **extenders**, and $\forall K_i^D \cap K^S = \emptyset$. Denote $\mathcal{K} := \bigcup K_i^D$. An **extended aggregated spreadsheet** is constructed by showing on each line, and $\forall K' \in \mathcal{K}$, information for

$$c^K(\vec{x}^K), \qquad \vec{C}^{K'} := \vec{C}(\vec{x}^K, K'), \qquad \kappa(\vec{C}^{K'}), \qquad G(\vec{C}^{K'}), \qquad Q^{\times}(\vec{C}^{K'}),$$

### 8.2.4 Examples

Considering the example in Sec. 4.3, let $t$ be the table shown in Tab. 1, and consider the query:

SELECT $X_2, X_3$, DIST($X_1$)
FROM $t$
GROUP BY $X_2, X_3$

Output displayed in the VisTool spreadsheet would be as shown in Tab. 7. According to $\kappa$ alone, the state $\vec{x}^{2,3} = \langle x, \beta\rangle \in X \downarrow \{2,3\}$ would be selected as that with the most structure, whereas according to both $G$ and $Q^{\times}$, $\langle x, \alpha\rangle$ would be.

Then consider the query

SELECT $X_2$, DIST($X_1$), DIST($X_3$)
FROM $t$
WHERE $X_2 = y$
GROUP BY $X_2$

with VisTool extended aggregate spreadsheet output as shown in Tab. 8. These results simply reflect those from Sec. 6.3.2, as shown in Tab. 6. In this case, we are restricted to an initial subset and projector, and consider which extension to move to. They have identical $\kappa$, so resorting to $G$ and $Q^{\times}$, we should select a new projector $K' = \{1,2\}$ based on $Q^{\times}(\vec{C}^{1,2}) = 0.260 < Q^{\times}(\vec{C}^{2,3}) = 0.269$.

| $\vec{x}^2$ / $X_2$ | $c^2$ | $\vec{C}^{12}$ | $\kappa(\vec{C}^{12})$ | $G(\vec{C}^{12})$ | $Q^{\times}(\vec{C}^{12})$ | $\vec{C}^{23}$ | $\kappa(\vec{C}^{23})$ | $G(\vec{C}^{23})$ | $Q^{\times}(\vec{C}^{23})$ |
|---|---|---|---|---|---|---|---|---|---|
| $y$ | 13 | $\langle 8, 5\rangle$ | 0.270 | 0.961 | 0.260 | $\langle 7, 6\rangle$ | 0.270 | 0.996 | 0.269 |

Table 8: Example extended SQL output.

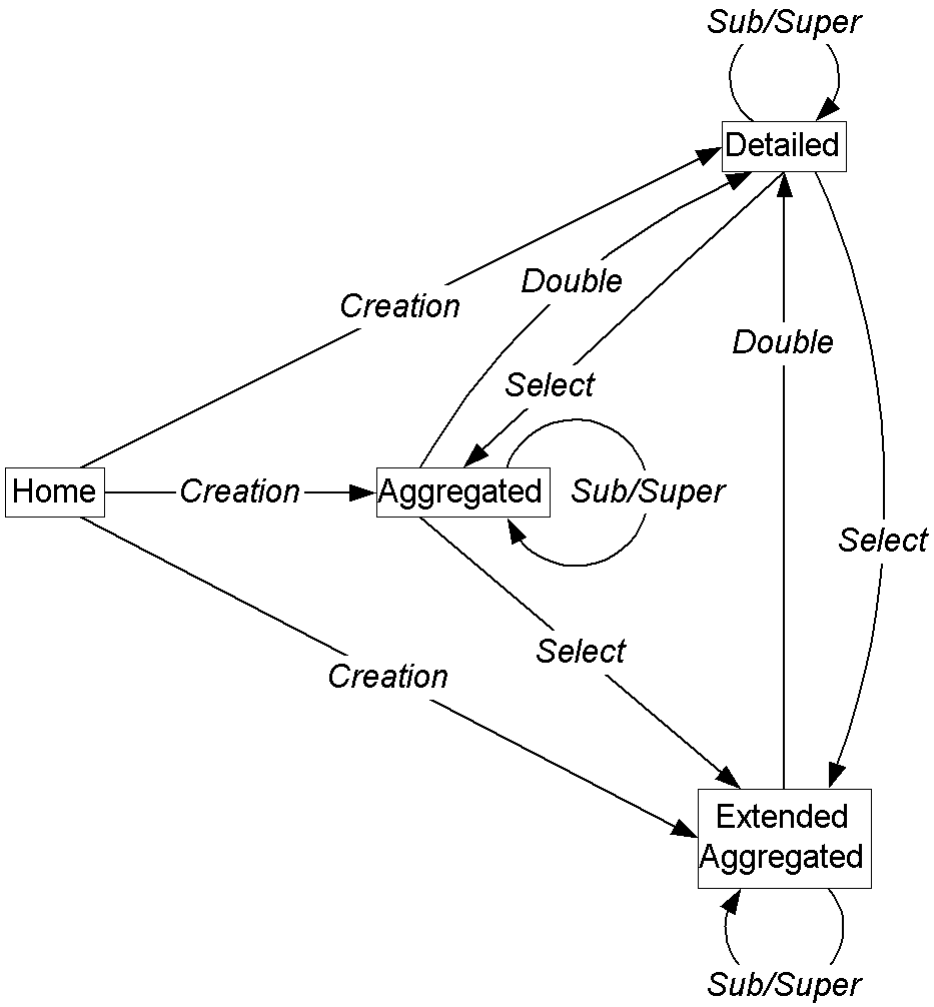


Figure 6: Transitions among VisTool spreadsheets.

### 8.3 Front-End Capabilities

We now describe the actual methods of operating with VisTool spreadsheets, and the transitions among them, as shown in Fig. 6. Boxes indicate types of spreadsheets, or **home**, indicating creation of a new spreadsheet from the top level.

We now describe the possible transitions in detail:

**Creation:** From home, $K^S$ must be specified in the preset constructor, and $Y$ in the condition constructor ($Y$ could be empty).

- **To Extended Aggregated:** If $K^G$ is specified in the preset constructor, then $K^A$ may optionally be. If the $K_i^D$ are also specified, then an extended aggregated spreadsheet is produced.
- **To Aggregated:** If $K^G$ is specified, but the $K_i^D$ are not.
- **To Detailed:** If $K^G$ is not specified.

**Subsetting:** In general, selecting a collection of rows from any spreadsheet produces a new $Y' \subseteq Y$.

- **Subsetting Proper:** A new spreadsheet of the same type can be created, reflecting now just those rows in $Y'$.
- **Double-Clicking:** From either of the aggregated spreadsheets, double-clicking produces a detailed spreadsheet reflecting $Y'$.

**"Selection":** From given spreadsheet, columns may be selected to indicate $K^G, K^A$, or the $K_i^D$.

- **Detailed to Aggregated:** From a detailed spreadsheet, if columns are selected from $K^S$ to indicate $K^G$, and optionally $K^A$.
- **Detailed to Extended Aggregated:** From a detailed spreadsheet, if columns are selected from $K^S$ to indicate $K^G$, optionally $K^A$, and the $K_i^D$.
- **Aggregated to Extended Aggregated:** From an aggregated spreadsheet, if columns are selected to indicate $K^G$ the $K_i^D$.

## References


[1] Agarwal, Sameet; Agrawal, Rakesh; and Deshpande, PM et al.: (1996) "On the Computation of Multidimensonal Aggregates", in: *Proc. 22nd VLDB Conference*, Bombay

[2] Birkhoff, Garrett: (1940) *Lattice Theory*, v. **25**, Am. Math. Soc., Providence RI, 3rd edition

[3] Gouw, Deky and Jones, Bush: (1996) "The Interaction Concept of K-Systems Theory", *Int. J. General Systems*, v. **24**:1-2, pp. 163-169

[4] Goodman, IR; Mahler, Ronald PS; and Nguyen, HT: (1997) *Mathematics of Data Fusion*, Kluwer

[5] Goutsias, J; Mahler, Ronald PS; and Nguyen, HT, eds.: (1998) *Random Sets: Theory and Applications*, Springer-Verlag

[6] Gray, Jim; Chaudhuri, Surajit; and Bosworth, A et al.: (1997) "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals", *Data Mining and Knowledge Discovery*, v. **1**, pp. 29-53

[7] Gyssens, Marc and Lakshmanan, Laks VS: (1997) "A Foundation for Multi-Dimensional Databases", in: *Proc. 23rd VLDB Conf.*, pp. 106-115

[8] Jones, Bush: (1986) "$K$-Systems versus Classical Multivariate Systems", *Int. J. General Systems*, v. **12**, pp. 1-6

[9] Joslyn, Cliff: (1993) "Possibilistic Semantics and Measurement Methods in Complex Systems", in: *Proc. 2nd Int. Symposium on Uncertainty Modeling and Analysis*, ed. Bilal Ayyub, pp. 208-215, IEEE Computer Soc.

[10] Joslyn, Cliff: (1995) "In Support of an Independent Possibility Theory", in: *Foundations and Applications of Possibility Theory*, ed. G de Cooman et al., pp. 152-164, World Scientific, Singapore

[11] Joslyn, Cliff: (1996) "Aggregation and Completion of Random Sets with Distributional Fuzzy Measures", *Int. J. of Uncertainty, Fuzziness, and Knowledge-Based Systems*, v. **4**:4, pp. 307-329

[12] Joslyn, Cliff: (1997) "Distributional Representations of Random Interval Measurements", in: *Uncertainty Analysis in Engineering and the Sciences*, ed. Bilal Ayyub and Madan Gupta, pp. 37-52, Kluwer

[13] Joslyn, Cliff: (1997) "Towards General Information Theoretical Representations of Databases Problems", in: *Proc. 1997 Conf. of the IEEE Society for Systems, Man and Cybernetics*, v. **2**, pp. 1662-1667

[14] Joslyn, Cliff: (1997) "Measurement of Possibilistic Histograms from Interval Data", *Int. J. General Systems*, v. **26**:1-2, pp. 9-33

[15] Klir, George: (1985) *Architecture of Systems Problem Solving*, Plenum, New York

[16] Klir, George: (1991) "Measures and Principles of Uncertainty and Information: Recent Developments", in: *Information Dynamics*, ed. H. Atmanspacher, pp. 1-14, Plenum Press, New York

[17] Klir, George and Folger, Tina: (1987) *Fuzzy Sets, Uncertainty, and Information*, Prentice Hall

[18] Klir, George and Harmanec, David: (1996) "Generalized Information Theory: Recent Developments", *Kybernetes*, v. **25**:7-8, pp. 50-67

[19] Madden, RF and Ashby, Ross: (1972) "On Identification of Many-Dimensional Relations", *Int. J. Systems Science*, v. **3**, pp. 343-356

[20] Pittarelli, Michael: (1994) "An Algebra for Probabilistic Databases", *IEEE Trans. on Knowledge and Data Engineering*, v. **6**:2, pp. 293-303

[21] Sudkamp, Thomas: (1992) "On Probability-Possibility Transformations", *Fuzzy Sets and Systems*, v. **51**, pp. 73-81

[22] Wang, Zhenyuan and Klir, George J: (1992) *Fuzzy Measure Theory*, Plenum Press, New York

[23] Yager, Ronald R: (1986) "On the Theory of Bags", *Int. J. of General Systems*, v. **13**:1, pp. 23-38

[24] Zwick, Martin and Shu, Hui: (1995) "Set-Theoretic Reconstructibility of Elementary Cellular Automata", *Advances in Systems Science and Applications*, v. **spcl**