Scientific Annals of Computer Science vol. 28 (1), 2018, pp. 141–156
doi: 10.7561/SACS.2018.1.141

On Partition Metric Space, Index Function, and Data Compression

Dan A. Simovici[1], Roman Sizov[1]

Abstract

We discuss a metric structure on the set of partitions of a finite set induced by the Gini index and two applications of this metric: the identification of determining sets for index functions using techniques that originate in machine learning, and a data compression algorithm.

Keywords: Gini index, Vapnik-Chervonenkis dimension, index function, determining set, compression

1 Introduction

The Gini index was developed as a measure of wealth inequality by the Italian statistician Corrado Gini [1, 2] and became increasingly important in machine learning. The Gini index is related but distinct from Shannon entropy (since it belongs to the same family of measures of diversity of probability distributions) and can be given an algebraic treatment that is useful in our context.

We discuss two rather distinct problems where the Gini index and a metric induced by this index on the set of partitions of a finite set prove to be useful, namely, the identification of determining sets for index functions, and a compression algorithm.

Index functions were introduced and studied by T. Sasao in a series of papers [3–10, 15, 16] and have multiple applications including terminal access controllers, IP address table lookup, packet filtering, memory patch and virus scan circuits, fault maps for memory, etc. In general, the number of variables

[1] University of Massachusetts Boston, Computer Science Department, Boston, USA, Emails: dsim@cs.umb.edu, rsizov@cs.umb.edu

is large and these functions do not depend effectively on all their variables. Therefore, identification of sets of minimal sets of variables on which such functions depend (known as determining sets) may lead to simplification of circuits that implement these functions. We investigated the identification of determining sets for index functions in a previous contribution and proposed an Apriori-like algorithm [17].

Let S be a finite set and let $\mathcal{P}(S)$ be the collection of its subsets. A partition of S is a collection π of pairwise disjoint, non-empty subsets of S, $\{B_1, \ldots, B_m\}$ such that $\bigcup_{i=1}^{m} B_i = S$. The sets $B_1, \ldots, B_m$ are the *blocks* of π. The set of partitions of S is denoted by $\mathsf{PART}(S)$.

A partial order relation is introduced on $\mathsf{PART}(S)$. For $\pi, \sigma \in \mathsf{PART}(S)$ we write $\pi \leqslant \sigma$ if each block of π is included in a block of σ. It is easy to see that this is equivalent to asking that each block of σ is a union of blocks of π. The largest partition in $\mathsf{PART}(S)$ is the one-block partition $\omega_S = \{S\}$; the smallest partition is $\alpha_S = \{\{x\} \mid x \in S\}$ that consists of singletons.

If $\pi, \sigma \in \mathsf{PART}(S)$, the partition $\pi \wedge \sigma$ is the partition of S that consists of sets of the form $B_i \cap C_j$, where $B_i \in \pi$, $C_j \in \sigma$, and $B_i \cap C_j \neq \emptyset$. Clearly, we have $\pi \wedge \sigma \leqslant \pi$, and $\pi \wedge \sigma \leqslant \sigma$. Also, $\rho \leqslant \pi$ and $\rho \leqslant \sigma$ if and only if $\rho \leqslant \pi \wedge \sigma$.

For $U, V \in \mathcal{P}(S)$ denote by $U \oplus V$ the symmetric difference of the sets U and V. We have

$$|U \oplus V| = |U| + |V| - 2|U \cap V|.$$

The mapping $d : \mathcal{P}(S)^2 \longrightarrow \mathbb{R}_{\geqslant 0}$ defined as $d(U, V) = |U \oplus V|$ is a metric on $\mathcal{P}(S)$. In other words, we have $d(U, V) = d(V, U)$, $d(U, V) = 0$ if and only if $U = V$, and $d(U, V) \leqslant d(U, W) + d(W, V)$ for every $U, V, W \in \mathcal{P}(S)$.

The paper is structured as follows. In Section 2 we discuss the metric space of partitions of finite sets. Then, in Section 3 we establish a link between the Vapnik-Chervonenkis dimension of collections of sets and the size of determining sets for index function. An algorithm for data compression based on the Gini index is presented in Section 4. Finally, we present our conclusions in Section 5.

2 The Metric Space of Set Partitions

For a partition $\pi \in \mathsf{PART}(S)$ let P_π be the equivalence relation defined by π that consists of all pairs $(x, y) \in S \times S$ such that x and y belong to the same block B_i of π. In other words, for $\pi = \{B_i \mid i \in I\}$ we have $P_\pi = \bigcup_{i \in I}(B_i \times B_i)$.

For $\pi, \sigma \in \mathsf{PART}(S)$ it is clear that $\pi = \sigma$ if and only if $P_\pi = P_\sigma$. For a finite set S we define a metric on $\mathsf{PART}(S)$ as

$$\delta(\pi, \sigma) = \frac{1}{|S|^2}|P_\pi \oplus P_\sigma| = \frac{1}{n^2}\left(|P_\pi| + |P_\sigma| - 2|P_\pi \cap P_\sigma|\right),$$

where "$\oplus$" denotes the symmetric difference of two sets and $n = |S|$.

The *Gini index* of the partition $\pi = \{B_1, \ldots, B_m\}$ is the number

$$\mathsf{gini}(\pi) = 1 - \frac{|P_\pi|}{n^2} = 1 - \sum_{i=1}^{m} \frac{|B_i|^2}{n^2},$$

that is, the relative number of pairs that do not inhabit the same block of the partition π.

The largest value of $\mathsf{gini}(\pi)$ for a partition in $\mathsf{PART}(S)$ that has m blocks is obtained when all blocks have equal sizes and equals $1 - \frac{1}{m}$ (when $n = |S|$ is a multiple of m). The least value is obtained when π consists of $m - 1$ blocks of size 1 and one block of size $n - m + 1$ and equals

$$1 - \frac{m-1}{n^2} - \frac{(n-m+1)^2}{n^2} = \frac{(m-1)(2n-m)}{n^2}.$$

Let now $\pi = \{B_1, \ldots, B_m\}, \sigma = \{C_1, \ldots, C_p\}$ be two partitions of a set S and let $\pi \wedge \sigma$ be the partition of S whose blocks are the non-empty intersection $B_i \cap C_j$ of blocks of π and σ. We have $P_{\pi\wedge\sigma} = P_\pi \cap P_\sigma$. Denote a block $B_i \cap C_j$ by D_{ij}. If $d_{ij} = |D_{ij}|$ we have $|B_i| = \sum_j |D_{ij}|$ and $|C_j| = \sum_i |D_{ij}|$. Therefore, $|P_{\pi\wedge\sigma}| = \sum_{i=1}^m \sum_{j=1}^p |D_{ij}|^2$, $|P_\pi| = \sum_{i=1}^m \left(\sum_{j=1}^p |D_{ij}|\right)^2$, and $|P_\sigma| = \sum_{j=1}^p \left(\sum_{i=1}^m |D_{ij}|\right)^2$. This allows us to write

$$\begin{aligned}
&\delta(\pi, \sigma) \\
&= \frac{1}{n^2} d(P_\pi, P_\sigma) = \frac{1}{n^2}\left(|P_\pi| + |P_\sigma| - 2|P_\pi \cap P_\sigma|\right) \\
&= \frac{1}{n^2}\left(\sum_{i=1}^m \left(\sum_{j=1}^p |D_{ij}|\right)^2 \right. \\
&\quad \left. + \sum_{j=1}^p \left(\sum_{i=1}^m |D_{ij}|\right)^2 - 2\sum_{i=1}^m \sum_{j=1}^p |D_{ij}|^2\right).
\end{aligned}$$

In terms of the gini function $\delta(\pi,\sigma)$ can be written as

$$\begin{aligned}\delta(\pi,\sigma) &= 1-\mathsf{gini}(\pi)+1-\mathsf{gini}(\sigma)-2(1-\mathsf{gini}(\pi\wedge\sigma))\\ &= 2\mathsf{gini}(\pi\wedge\sigma)-\mathsf{gini}(\pi)-\mathsf{gini}(\sigma).\end{aligned}$$

Furthermore, we have

$$\mathsf{gini}(\pi)=\delta(\pi,\omega_S)=1-\frac{1}{n}-\delta(\pi,\alpha_S).$$

Example 2.1 In the case of two-block partitions of a set T with $|T|=n$ the distance has a very simple form. Suppose that $\pi=\{B_0,B_1\}$, $\sigma=\{C_0,C_1\}$ and $\pi\wedge\sigma=\{D_{00},D_{01},D_{10},D_{11}\}$. Let D be the matrix

$$D=\begin{pmatrix} d_{00} & d_{01}\\ d_{10} & d_{11}\end{pmatrix},$$

where $d_{ij}=|D_{ij}|$ for $i,j=0,1$. The distance $\delta(\pi,\sigma)$ is

$$\begin{aligned}\delta(\pi,\sigma) &= \frac{1}{n^2}\big((d_{00}+d_{01})^2+(d_{10}+d_{11})^2\\ &\quad +(d_{00}+d_{10})^2+(d_{01}+d_{11})^2\\ &\quad -2(d_{00}^2+d_{01}^2+d_{10}^2+d_{11}^2)\big)\\ &= \frac{2}{n^2}(d_{00}d_{01}+d_{11}d_{01}+d_{00}d_{10}+d_{11}d_{10})\\ &= \frac{2}{n^2}(d_{00}+d_{11})(d_{01}+d_{10}). \qquad (1)\end{aligned}$$

□

3 Determining Sets for Index Functions

Let $X=\{x_1,\ldots,x_m\}$ be a finite set of symbols called *attributes.* A set $\mathrm{Dom}(x_i)$ referred to as the *domain* of x_i is attached to each attribute x_i, and a *table* having the heading X is defined as a pair $T=(X,R)$, where R, the content of the table is a relation on $\prod_{i=1}^{m}\mathrm{Dom}(x_i)$. The members of R are the *tuples* or the rows of the table. The *weight* of T is the number of tuples, $w(T)=|R|$. Note that the tables defined as above *do not contain duplicate rows.*

We adopt the relational database theory notation, where subsets of table headings are denoted as strings.

Table 1: Tabular Representation of a Partial Function

x_1	x_2	x_3	x_4	x_5	x_6	x_7	y
0	0	1	0	0	1	1	1
1	0	0	0	1	0	1	2
1	0	1	1	0	0	1	3
0	0	1	1	1	1	0	4
1	1	0	0	1	1	0	5
0	1	1	0	1	0	1	6
1	1	0	1	0	1	1	7
0	0	0	1	1	0	0	8
0	1	1	1	0	1	1	9

If $t = (a_1, \ldots, a_m)$ is a tuple in T, the restriction of t that consists of components that correspond to the attributes $Y = x_{i_1} \cdots x_{i_k}$ is denoted by $t[Y] = (a_{i_1}, \ldots, a_{i_k})$ and is referred to as the *projection* of t on Y.

Let $\mathbf{k}$ be the finite set $\{0, 1, \ldots, k-1\}$. The number k is referred to as the *radix* of the set $\mathbf{k}$. A *k-table* is a table $T = (X, R)$ with $\operatorname{Dom} x_i = \mathbf{k}$ for $1 \leqslant i \leqslant m$.

Consider a set of n different binary vectors of m bits referred to as *registered vectors*. An *index generation function* or, more briefly, an *index function* assigns to every registered vector a unique integer from 1 to n. A circuit implementing the index function produces a value k if its input matches the k^{th} registered vector, and 0 otherwise. The number n is the weight of the index generation function. Thus, an index generation function represents a mapping: $f; \{0,1\}^m \longrightarrow \{0, 1, \ldots, n\}$.

An *index table* is a table that describes an index function and is defined as a pair $T = (x_1 \cdots x_m y, R)$, where $\operatorname{Dom}(x_i) = \mathbf{2}$ and $\operatorname{Dom}(y) = \{1, \ldots, n\}$, where $n = w(T)$. Thus, an index table is a table whose attributes are binary with the exception of the *index attribute* y that is an n-ary attribute, where $n = w(T)$.

Example 3.1 In Table 1 we show an $(\mathbf{2}, \mathbf{9})$-index table that contains nine tuples in $\mathbf{2}^7 \times \mathbf{9}$:

For instance, $t_5 = (0, 1, 1, 0, 1, 0, 1, 6)$. □

If the index table T has the heading $x_1 \ldots x_n y$, then T defines a collection $\mathcal{C}_T$ of subsets of the set $X = x_1 \ldots x_n$ by interpreting the rows of T as characteristic vectors of these subsets.

Example 3.2 For the table T given in Example 3.1 the collection $\mathcal{C}_T$ consists of the following sets:

$$\begin{aligned}&C_0 = \{x_3, x_6, x_7\}, C_1 = \{x_1, x_5, x_7\},\\ &C_2 = \{x_1, x_3, x_4, x_7\}, C_3 = \{x_3, x_4, x_5, x_6\}\\ &C_4 = \{x_1, x_2, x_5, x_6\}, C_5 = \{x_2, x_3, x_5, x_7\},\\ &C_6 = \{x_1, x_2, x_4, x_6, x_7\}, C_7 = \{x_4, x_5\},\\ &C_8 = \{x_2, x_3, x_4, x_6, x_7\}.\end{aligned}$$

□

We use next the Vapnik-Chervonenkis dimension of a collection of sets. This characteristic property of collection of sets is of fundamental importance for machine learning and data mining [13]. A collection $\mathcal{C}$ of subsets of a set X *shatters a subset* U *of* X if

$$\mathcal{P}(U) = \{C \cap U \mid C \in \mathcal{C}\}.$$

The family of sets shattered by $\mathcal{C}$ is denoted by $\mathsf{SH}(\mathcal{C})$. The size of the largest set in $\mathsf{SH}(\mathcal{C})$ is the *Vapnik-Chervonenkis dimension* $\mathsf{VC}(\mathcal{C})$ of the collection $\mathcal{C}$.

For $k, m \in \mathbb{N}$ and $k \leqslant m$ let $\phi(m, k) = \sum_{i=0}^{k} \binom{m}{i}$. If $|X| = m$, there are $\phi(m, k)$ subsets of a set X that contain at most k elements.

The Sauer-Shelah theorem [11, 14] stipulates that if $\mathcal{C}$ is a collection of subsets of X such that $|\mathcal{C}| > \phi(m, k-1) = \sum_{i=0}^{k-1} \binom{m}{i}$, then X contains a set U with $|U| \geqslant k$ that is shattered by $\mathcal{C}$. In other words, for such a collection $\mathsf{VC}(\mathcal{C}) \geqslant k$.

Note that in order to shatter a d-element set U a collection $\mathcal{C}$ must contain at least 2^d sets. Therefore, $\mathsf{VC}(\mathcal{C}) = d$ implies $2^d \leqslant |\mathcal{C}|$.

If $\mathsf{VC}(\mathcal{C}) = d$, no set with more than d elements is shattered by $\mathcal{C}$. Therefore, if $\mathcal{P}_d(X)$ is the family of subsets of X that contain d or fewer elements, $\mathsf{SH}(\mathcal{C}) \subseteq \mathcal{P}_d(X)$, hence

$$2^d \leqslant |\mathcal{C}| \leqslant \sum_{i=0}^{d} \binom{m}{i}. \tag{2}$$

Theorem 3.3 *If $\mathcal{C}$ is a collection of subsets of set X with $|X| = m$ and there exist $k, \ell \in \mathbb{N}$ such that $\phi(m, k-1) < |\mathcal{C}| < 2^\ell$, then $k \leqslant \mathsf{VC}(\mathcal{C}) < \ell$.*

Proof: Suppose that $\mathcal{C}$ is a collection of subsets of a set X with $|X| = m$ such that

$$\phi(m, k-1) < |\mathcal{C}| < 2^\ell, \tag{3}$$

where $\ell = \lceil \log_2(|\mathcal{C}| + 1) \rceil$. The first inequality implies $\mathsf{VC}(\mathcal{C}) \geqslant k$ by the Sauer-Shelah theorem. The second inequality yields $\mathsf{VC}(\mathcal{C}) < \ell$ because $\mathcal{C}$ must contain at least 2^ℓ sets in order to shatter a set of size ℓ. Thus, Inequalities (3) imply $k \leqslant \mathsf{VC}(\mathcal{C}) < \ell$. □

Example 3.4 Let $X = \{x_1, x_2, x_3\}$ and let T_1, T_2 be the tables shown below:

T_1

x_1	x_2	x_3
1	0	0
0	1	0
1	1	0
1	0	1

T_2

x_1	x_2	x_3
1	0	0
0	1	0
0	0	1
1	1	0

Let $\mathcal{C}_{T_1}, \mathcal{C}_{T_2}$ be the collections of sets defined by these tables. We have $|\mathcal{C}_{T_1}| = |\mathcal{C}_{T_2}| = 4$ and, therefore,

$$\phi(3, 1) = 4 = |\mathcal{C}| < \phi(3, 2) = 7.$$

We have $\mathsf{VC}(\mathcal{C}_{T_1}) = 1$ because $\mathcal{C}_{T_1}$ shatters all one-element subsets but does not shatter any larger sets, and $\mathsf{VC}(\mathcal{C}_{T_2}) = 2$ because $\mathcal{C}_{T_2}$ shatters the set $\{x_1, x_2\}$. □

Example 3.5 The table T from Example 3.1 contains 9 tuples, so we have $m = 7$ and $|\mathcal{C}_T| = 9$. Since

$$\phi(7, 1) \leqslant 9 < 2^4,$$

by Theorem 3.3, we may conclude that $\mathsf{VC}(\mathcal{C}) \in \{2, 3\}$. An inspection of the table shows that there exists a set of three attributes that is shattered by $\mathcal{C}$. For example, one such set is $\{x_2, x_3, x_4\}$, hence $\mathsf{VC}(\mathcal{C}) = 3$. □

Suppose that all tuples of a $(\mathbf{2}, \mathbf{n})$-table T are distinct. This allows us to define a multi-valued partial injective function $f_T : \mathbf{2}^m \longrightarrow \{0, \ldots, n-1\}$ with binary inputs and multivalued output. The set of all such partial functions is denoted by $\mathsf{PF}(\mathbf{2}^m, \mathbf{n})$.

The registered vectors of an index table T whose heading is $\{x_1, \ldots, x_m, y\}$ can be regarded as the characteristic vectors of certain subsets of the set $X = \{x_1, \ldots, x_m\}$ as shown next. Namely, if $t = (a_1, \ldots, a_m, b)$ is a tuple, its corresponding subset is $C_b = \{x_i \mid a_i = 1 \text{ for } 1 \leqslant i \leqslant m\}$.

If a subset U of the heading X with $|U| = k$ is shattered by $\mathcal{C}_T$, then $T[U]$ contains all binary equivalents of numbers between 0 and $2^k - 1$ (and some values may be repeated).

Definition 3.6 Let $f : \mathbf{2}^m \longrightarrow \mathbf{n}$ be an index function described by the index table $T_f = (X, R)$, where $X = x_1 \cdots x_m y$. A set of variables $V = \{x_{i_1}, \ldots, x_{i_p}\}$ is a *determining set* for f if $|\mathcal{C}_V| = |\mathcal{C}_{T_f}|$. □

In other words, if $V = \{x_{i_1}, \ldots, x_{i_p}\}$ is a determining set for the index function f, then the projection $T_f[x_{i_1} \cdots x_{i_p} y]$ is also an index table.

Theorem 3.7 *A minimal determining set for an index function f contains a maximal set of attributes that is shattered by $\mathcal{C}_{T_f}$.*

Proof: Let V be a minimal determining set for the index function f. Note that V does not contain an attribute x who has constant values (1 or 0) in T_f for, otherwise, we would be able to drop x and the set $V - \{x\}$ would still be a determining set.

Thus, if $x \in V$, both 0 and 1 are present under x and the set $\{x\}$ is shattered by $\mathcal{C}_{T_f}$. This shows that V contains sets that are shattered by $\mathcal{C}_{T_f}$. Since $\mathcal{P}(V)$ is finite, it is immediate that there are maximal subsets of V that are shattered by $\mathcal{C}_{T_f}$. □

Observe that the projection of the table T_f on W need not contain distinct values, so $\mathsf{gini}(\pi_W) < 1 - \frac{1}{w(T)}$. By systematically expanding a set W that is shattered, that is, by adding to W subsets L of $X - W$ it is possible to reach a determining set $V = WL$. Thus, the maximum size of a shattered set by $\mathcal{C}$ offers a lower bound for the size of determining sets and allows avoiding a search of the entire collection of subsets of X.

These considerations suggest the Algorithm 3.1 for identifying determining sets.

Example 3.8 In Example 3.5 we have shown that the Vapnik-Chervonenkis dimension of the collection of sets introduced in Example 3.1 is 3. Therefore any determining set for the index function specified must include at least 3 variables. The computation of the Gini index for three-variable subsets shown below indicates that none of these sets has the Gini index of $0.8889 \approx 1 - \frac{1}{9}$, but there are several such sets (shown in bold characters) that have a maximum value of 0.8642.

```
Algorithm 3.1: Identification of Determining Sets
  Input  : Index table $T = (X, R)$
  Output : Collection of Determining Sets $\mathsf{DS}(f)$ for the Index
           Function Represented by $T$
begin
    set $m = |X|$;
    set $\mathsf{DS}(f) = \emptyset$, $d_1 = 0$, $d_2 = 0$;
    while not $(|\mathcal{C}| > \sum_{i=0}^{d_1} \binom{m}{i})$ do
        $d_1 + +$;
    end
    while not $(|\mathcal{C}| < 2^{d_2})$ do
        $d_2 + +$;
    end
    foreach $W \in \mathcal{P}(X)$ with $d_1 \leq |W| \leq d_2$ and maximal Gini index
     do
        if $W$ is shattered by $\mathcal{C}$ then
            foreach $L \in \mathcal{P}(X - W)$ do
                if $\mathit{gini}(\pi_{W \cup L}) \geq 1 - \frac{1}{m}$ then
                    add $W \cup L$ to $\mathsf{DS}(f)$
                end
            end
        end
    end
end
```

$x_1x_2x_3$: 0.8148	$x_1x_2x_4$: **0.8642**	$x_1x_2x_5$: **0.8642**	$x_1x_2x_6$: 0.8148
$x_1x_2x_7$: 0.8148	$x_1x_3x_4$: 0.8148	$x_1x_3x_5$: 0.8148	$x_1x_3x_6$: 0.7901
$x_1x_3x_7$: 0.7901	$x_1x_4x_5$: 0.8148	$x_1x_4x_6$: **0.8642**	$x_1x_4x_7$: 0.8148
$x_1x_5x_6$: 0.8395	$x_1x_5x_7$: 0.8148	$x_1x_6x_7$: 0.8395	$x_2x_3x_4$: **0.8642**
$x_2x_3x_5$: 0.8395	$x_2x_3x_6$: 0.8148	$x_2x_3x_7$: 0.8395	$x_2x_4x_5$: 0.8148
$x_2x_4x_6$: 0.8395	$x_2x_4x_7$: 0.8148	$x_2x_5x_6$: 0.8395	$x_2x_5x_7$: 0.8148
$x_2x_6x_7$: 0.8395	$x_3x_4x_5$: 0.8395	$x_3x_4x_6$: **0.8642**	$x_3x_4x_7$: 0.8395
$x_3x_5x_6$: 0.8395	$x_3x_5x_7$: 0.7901	$x_3x_6x_7$: 0.8395	$x_4x_5x_6$: 0.8395
$x_4x_5x_7$: 0.7654	$x_4x_6x_7$: 0.8395	$x_5x_6x_7$: 0.7654	

These sets can be extended to a determining set. In the next table, the extensions of the sets of size 3 that are determining sets are shown in bold characters:

$x_1x_2x_3x_4$: **0.8889** $x_1x_2x_3x_5$: **0.8889** $x_1x_2x_3x_6$: 0.8395 $x_1x_2x_3x_7$: 0.8642
$x_1x_2x_4x_5$: 0.8642 $x_1x_2x_4x_6$: **0.8889** $x_1x_2x_4x_7$: 0.8642 $x_1x_2x_5x_6$: **0.8889**
$x_1x_2x_5x_7$: 0.8642 $x_1x_2x_6x_7$: 0.8642 $x_1x_3x_4x_5$: 0.8642 $x_1x_3x_4x_6$: 0.8642
$x_1x_3x_4x_7$: 0.8642 ; $x_1x_3x_5x_6$: 0.8642 $x_1x_3x_5x_7$: 0.8642 $x_1x_3x_6x_7$: 0.8642
$x_1x_4x_5x_6$: **0.8889** $x_1x_4x_5x_7$: 0.8395 $x_1x_4x_6x_7$: **0.8889** $x_1x_5x_6x_7$: 0.8642
$x_2x_3x_4x_5$: **0.8889** $x_2x_3x_4x_6$: **0.8889** $x_2x_3x_4x_7$: **0.8889** $x_2x_3x_5x_6$: 0.8642
$x_2x_3x_5x_7$: 0.8642 $x_2x_3x_6x_7$: 0.8642 $x_2x_4x_5x_6$: 0.8642 $x_2x_4x_5x_7$: 0.8395
$x_2x_4x_6x_7$: 0.8642 $x_2x_5x_6x_7$: 0.8642 $x_3x_4x_5x_6$: **0.8889** $x_3x_4x_5x_7$: 0.8642
$x_3x_4x_6x_7$: **0.8889** $x_3x_5x_6x_7$: 0.8642 $x_4x_5x_6x_7$: 0.8395

□

4 The Gini-based metric and data compression

A set of attributes U of a table $T = (X, R)$ generates a partition π_U of the set of rows R whose blocks consists of tuples that have the same projection on U.

Example 4.1 For the table introduced in Example 3.1 the partition of R induced by x_1x_2 is $\pi_{x_1x_2} = \{\{t_0, t_3, t_7\}, \{t_1, t_2\}, \{t_4, t_6\}, \{t_5, t_8\}\}$. □

This allows us to identify sets of attributes whose partitions are close in the sense of this distance. Formula (1) from Example 2.1 suggests that when $\delta(\pi_x, \pi_{x'})$ is small the columns corresponding to x and x' are rather similar. This allows encoding values that occur in the projection $T[xx']$ using a single value that belongs to a higher radix.

A hierarchical clustering algorithm produces a hierarchical system of clusters (also known as a *dendrogram*) as a tree. Cutting this tree at a certain height generates a clustering that groups together attributes that may be encoded together.

Example 4.2 For the table given in Example 3.1 the mutual distances between attributes are given in the following table:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	0.0000	0.4938	0.3457	0.4938	0.4938	0.4938	0.4938
x_2	0.4938	0.0000	0.4938	0.4938	0.4938	0.4444	0.4938
x_3	0.3457	0.4938	0.0000	0.4938	0.4444	0.4938	0.4444
x_4	0.4938	0.4938	0.4938	0.0000	0.4444	0.4938	0.4938
x_5	0.4938	0.4938	0.4444	0.4444	0.0000	0.4444	0.3457
x_6	0.4938	0.4444	0.4938	0.4938	0.4444	0.0000	0.4938
x_7	0.4938	0.4938	0.4444	0.4938	0.3457	0.4938	0.0000

Applying a single-link hierarchical clustering produces the dendrogram shown in Figure 1.

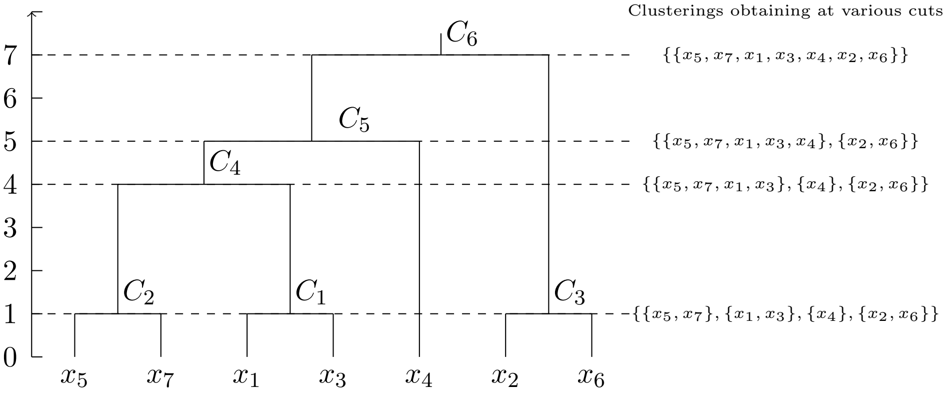
Clusterings obtaining at various cuts
7
6
5
4
3
2
1
0
C_6
C_5
C_4
C_2
C_1
C_3
$\{\{x_5, x_7, x_1, x_3, x_4, x_2, x_6\}\}$
$\{\{x_5, x_7, x_1, x_3, x_4\}, \{x_2, x_6\}\}$
$\{\{x_5, x_7, x_1, x_3\}, \{x_4\}, \{x_2, x_6\}\}$
$\{\{x_5, x_7\}, \{x_1, x_3\}, \{x_4\}, \{x_2, x_6\}\}$
x_5 x_7 x_1 x_3 x_4 x_2 x_6

Figure 1: Dendrogram of Single-Link Clustering

□

Algorithm 4.1 takes a dataset and a cutting height as an input and produces a compressed dataset together with a mapping file.

The algorithm creates a matrix of Gini-based distances between the attributes of the dataset. The distance matrix is used to run single-linkage clustering algorithm. This hierarchy is at the provided cutting height and values of the attributes in each cluster are encoded into a new column with the name of the cluster.

This joining is based on the assumption that we have small number of unique projections of dataset transactions on a set of attributes in the same cluster. This is done by running a **group by** query on the dataset projected on the set of attributes and enumerating the results of the query. This enumeration and a mapping of sets of attributes to respective cluster names are saved in the *mapping file* of the output. The new clustered columns and the columns for unclustered attributes form the compressed dataset.

Our experiments involved the *mushroom* data set [12] in a binarized form. For every original column we have created as many binary columns as the number of unique values in the column. So, for instance, if column c_1 has had 3 values a, b, c, then 3 binary columns have been created c_1_a,c_1_b and c_1_c. Whenever c_1 has value a, the column c_1_a has value 1 and 0 otherwise, etc. The resulting dataset had 126 attributes, and after removing

Algorithm 4.1: Compression Algorithm

```
Input  : Dataset $T = (X, R)$ and cutting height $r$
Output : Compressed Dataset $T'$, Mapping File $M$
begin
    Let $distanceMatrix[|R|][|R|]$ be a matrix of distances between attributes;
    foreach pair of attributes $(x_k, x_\ell)$ do
        Calculate distance $d(x_k, x_\ell) = 2\ \mathsf{gini}(\pi_{x_k}, \pi_{x_\ell}) - \mathsf{gini}(\pi_{x_k}) - \mathsf{gini}(\pi_{x_\ell})$;
        Set $distanceMatrix[k][\ell] = d$;
        Set $distanceMatrix[\ell][k] = d$;
    end
    Run Single Linkage Clustering Algorithm for the set of attributes $X$ with the distance matrix $distanceMatrix[|R|][|R|]$;
    foreach cluster with height $\leq r$ that is strictly not included in any other cluster with height $\leq r$ do
        Join all the items from the cluster into a new column named as the cluster;
        Run group by query on the dataset $T$ for the attributes in the cluster;
        Enumerate rows in the query result;
        Save the new column in the output dataset $T'$ where the value for each row is based on the enumeration from the query;
        Save all the mappings to the file $M$;
    end
    Save all the columns which attributes are not included in the clusters into the output dataset $T'$ without change;
end
```

any column that had either all zeros or all ones the new dataset had 116 attributes left. The resulting file had size of $1,851\ KB$.

The original dataset had 22 attributes and 1 attribute for the class (poisonous/edible) and 8124 transactions. The class column was excluded.

We ran Algorithm 4.1 for several different cutting height values r. The dependency of sizes of the compressed files on the cutting height value r is shown in Figure 2. It can be readily seen that the best compression happens when the cutting height value is about 0.35.

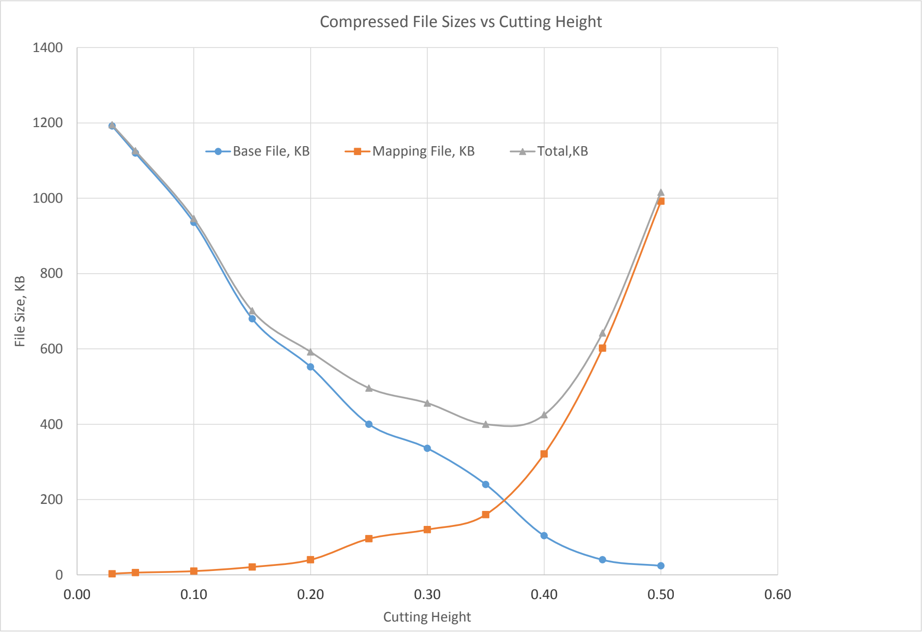

Figure 2: Compressed File Sizes vs Cutting Height

5 Conclusions

The Gini index that was developed for statistical purposes is a member of a broader family of diversity measures known as generalized entropies. We applied this index in conjunction with the Vapnik-Chervonenks dimension of collections of sets to develop an algorithm that seeks to identify determining sets for index function and provides a lower limit to the size of such sets. The relationship between determining sets and the Vapnik-Chervonenks dimension of the collection of sets defined by an index function suggests that this dimension is a good proxy for the complexity of index function, a further research goal to be explored.

The metric space generated by the Gini index on the set of partitions was used to develop a data compression algorithm starting from a clustering algorithm applied to table attributes. This compression is achieved by grouping together attributes that have similar value distributions.

It would be interesting to examine the use of other types of entropies (e.g. Shannon's entropy) for solving these problems.

References

[1] C. Gini. Sulla Misura della Concentrazione e della Variabilita dei Caratteri. *Transactions of the Real Istituto Veneto di Scienze, Lettere ed Arti*, LIII:1203, 1914. doi:10.1007/bf02858128.

[2] C. Gini. Measurement of Inequality of Incomes. *The Economic Journal*, 31(121):124–126, 1921. doi:10.2307/2223319.

[3] T. Sasao. *Switching Theory for Logic Synthesis*. Kluwer Academic Publishers, 1999. doi:10.1007/978-1-4615-5139-3.

[4] T. Sasao. On the Number of Dependent Variables for Incompletely Specified Multiple-Valued Functions. In *Proceedings of the 30th International Symposium on Multiple-Valued Logic*, pages 91–97, Los Alamitos, CA, 2000. Computer Society Press. doi:10.1109/ismvl.2000.848605.

[5] T. Sasao. Proceedings of the 35th International Symposium for Multiple-Valued Logic. In *Radix Converters: Complexity and Implementation by LUT Cascades*, pages 256–263, Los Alamitos, CA, 2005. Computer Society Press. doi:10.1109/ismvl.2005.50.

[6] T. Sasao. A Design Method of Address Generators Using Hash Memories. In *International Workshop on Logic and Synthesis*, pages 102–109, 2006.

[7] T. Sasao. On the Number of Variables to Represent Sparse Logic Functions. In *17th International Workshop on Logic and Synthesis (IWLS-2008)*, pages 233–239, Lake Tahoe, California, USA, 2008. IEEE-CS.

[8] T. Sasao. On the Number of Variables to Represent Sparse Logic Functions. In *2008 IEEE/ACM International Conference on Computer Aided Design*, pages 45–51, Los Alamitos, CA, 2008. Computer Society Press. doi:10.1109/iccad.2008.4681550.

[9] T. Sasao. A Fast Updatable Implementation of Index Generation Functions Using Multiple IGUs. *IEICE Transactions*, 100-D(8):1574–1582, 2017. doi:10.1587/transinf.2016lop0001.

[10] T. Sasao. A Linear Decomposition of Index Generation Functions: Optimization Using Autocorrelation Functions. *Multiple-Valued Logic and Soft Computing*, 28(1):105–127, 2017.

[11] N. Sauer. On the Density of Families of Sets. *Journal of Combinatorial Theory (A)*, 13:145–147, 1972. doi:10.1016/0097-3165(72)90019-2.

[12] J. S. Schlimmer. UCI Machine Learning Repository, 2013.

[13] D. A. Simovici and C. Djeraba. *Mathematical Tools for Data Mining – Set Theory, Partial Orders, Combinatorics*. Springer-Verlag, London, second edition, 2008.

[14] S. Shelah. A Combinatorial Problem; Stability and Order for Models and Theories in Infinitary Languages. *Pacific Journal of Mathematics*, 41:247–261, 1972. doi:10.2140/pjm.1972.41.247.

[15] T. Sasao and M. Matsuura. An Implementation of an Address Generator Using Hash Memories. In *10th EUROMICRO Conference on Digital System Design, Architectures, Methods and Tools, DSD-2007*, pages 69–76, Los Alamitos, CA, 2007. Computer Society Press. doi:10.1109/dsd.2007.4341452.

[16] T. Sasao, T. Nakamura, and M. Matsuura. Representation of Incompletely Specified Index Generation Functions Using Minimal Number of Compound Variables. In *12th EUROMICRO Conference*

on Digital System Design, Architectures, Methods and Tools, DSD-2009, pages 765–772, Patras, Greece, 2009. Computer Society Press. doi:10.1109/DSD.2009.214.

[17] D. A. Simovici, D. Pletea, and R. Vetro. Information-Theoretical Mining of Determining Sets for Partially Defined Functions. In *Proceedings of ISMVL*, pages 294–299, Barcelona, 2010. IEEE Computer Society. doi:10.1109/ismvl.2010.61.