



HAL
open science

An algebraic approach to ensemble clustering

Frédéric Dumonceaux, Guillaume Raschia, Marc Gelgon

► **To cite this version:**

Frédéric Dumonceaux, Guillaume Raschia, Marc Gelgon. An algebraic approach to ensemble clustering. Int. Conf. on Pattern Recognition (ICPR'2014), Aug 2014, Stockholm, Sweden. pp.1301-1306. hal-00967283

HAL Id: hal-00967283

<https://hal.science/hal-00967283v1>

Submitted on 28 Oct 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

An algebraic approach to ensemble clustering

F. Dumonceaux G. Raschia M. Gelgon
 Laboratoire d'Informatique Nantes Atlantique
 LINA (UMR CNRS 6241)
 Nantes university, France.
 Email: firstname.lastname@univ-nantes.fr

Abstract—In clustering, consensus clustering aims at providing a single partition fitting a consensus from a set of independently generated. Common procedures, which are mainly statistical and graph-based, are recognized for their robustness and ability to scale-up. In this paper, we provide a complementary and original viewpoint over consensus clustering, by means of algebraic definitions which allow to ascertain the nature of available inferences in a systematic approach (e.g. a knowledge base). We found our approach on the lattice of partitions, for which we shall disclose how some operators can be added with the aim to express a formula representing the consensus. We show that adopting an incremental approach may assist to retain significant amount of aggregate data which fits well with the set of input clusterings. Beyond that ability to model formulae, we also note that its potential cannot be easily captured through such a logical system. It is due to the volatile nature of handling partitions which finally impacts on ability to draw some valuable conclusions.

I. INTRODUCTION

This paper deals with data clustering or, more exactly, with the partitions that result from several clustering operations on a given data set. It discusses the way by which these partitions may be combined, to provide a single partition, which better reflects the organization of data into meaningful sub-populations. This task is well known as *ensemble clustering* or *consensus clustering*. In this work, our only input is a set of partitions over a same data set (i.e. we disregard initial object features and the techniques that are used for providing the partitions).

This paper takes an original viewpoint, by treating the consensus task within an algebraic structure gathering all possible partitions over the data set : the *partition lattice*. More precisely, we contribute by addressing the following questions:

- we express the optimal combination of partitions, by means of algebraic operators on partitions ; we relate the algebraic operators and their associated semantics (i.e. a logical view)
- we discuss how such an expression may be efficiently computed. To this aim, we present the properties of the algebraic operators in the partition lattice, and provide experimental results on these operators, in terms of efficiency.

Because the algebraic framework is not as commonly used for ensemble clustering as the statistical or graph-based ones, the paper first provides (in Section 2) the necessary background.

The reasons we explore the algebraic way to ensemble clustering may be summarized as follows :

- semantics and expression of consensus : an algebraic presentation can help understand the semantics mechanisms involved when combining partitions and, as a consequence, legitimate that a partition be the consensus-based representative of input partitions;
- expression of consensus and its efficient computation : this work aims at contributing to the more general goal of providing efficient ways of expressing and computing the many machine learning tasks that manipulate partitions. Currently, despite their ubiquitous presence in classification tasks, partitions are unfortunately not first-class citizens of programming environments. As a result, these matters are usually bypassed through ad hoc encoding of partitions and of their combinations, thus ignoring potentially useful properties better achievable in the partition lattice.

Our main result is about the study of the formal conditions that allow for inferring from knowledge over aggregates within a partition through dedicated operators. Handling algebraically such items means that at any time, what is acknowledged as valid remains true whatever the applied operation and any further operand. Monotonicity property then labels such logical system while their calculus is accurately achieved by a dedicated algebra. As we shall point out, the partition lattice, algebra over partitions, has not it and as a direct consequence, it hinders reliability of any drawn inference. Nevertheless, it is worth noting that it is not an end in itself as we shall see in Section III. The one that follows, emphasises relevance of algebraic calculus for operational use.

Clustering a data set into sub-populations is a fundamental, widely studied research track, both as a general data analysis topic and in many applied contexts (bioinformatics [1], audio-visual data analysis [2],...). This task is much studied because there appears to be no technique seems to fit all problems, and even evaluation itself it is complex task, since there often is no clear ground truth. The number of clusters is often difficult to determine. Models and techniques proposed in the literature optimize various criteria for achieving various forms of intra-class homogeneity, and the search for these optima involves a large variety of algorithms. In many cases, several runs of a same (local) optimization algorithm would lead to different partitions of the data set.

For these reasons, the last decade saw the rise of a new research track, *ensemble clustering*, which builds criteria and techniques for aggregation multiple partitions into a single,

better partition. The computation of these input partitions may have been trying to optimized antagonistic criteria. As a side remark, our paper does not cover *alternative clustering*, an extension of ensemble clustering for producing multiple, non-redundant partitions.

There are two main alternative criteria when building a consensus partition :

- background knowledge, provided by an expert and related to the applicative field producing the data set, may help formulate assumption on the partition generation mechanism and, hence, on the respective likelihoods of various possible partitions;
- the optimal partition may be defined and sought as a trade-off among the input partitions (e.g. [3], [4]).

An algebraic presentation of the problem aims at contributing to the understanding of the semantics of mechanisms involved when combining partitions. In social choice theory, a consensus is defined as an arrangement or combination of a set of profiles describing individuals, such that this combination represents as faithfully as possible as many profiles as possible.

In our situation, input partitions are individual profiles and one may consider that there exist a *preference relation* and that seeking a consensus partition is equivalent to seeking a partition which maximally *compatible* which the input partition set. A proper mixing procedure is then applied to *reconcile* the various “point of view”. A main purpose of this paradigm is that it can supply constraints on decision procedures that organize the way by which partitions may be combined. In this regard, it is also interesting to note that well-known median procedure originates from Kemeny-Young voting procedure [5].

II. ABOUT PARTITION LATTICE

In this section, we review basics on order theory and lattices such that we are able in the next section to build upon those formal background and propose new operators to achieve consensus.

A. Preliminaries

1) *Posets and lattices*: A partially ordered set, usually coined *poset*, (E, \leq) is a set E where elements may be comparable by a relation \leq having the following properties:

- 1) $x \leq x$ (reflexivity)
- 2) $x \leq y$ and $y \leq x \Rightarrow x = y$ (antisymmetry)
- 3) $x \leq y$ and $y \leq z \Rightarrow x \leq z$ (transitivity)

where x, y, z are elements in E .

In the following, we assume all the considered posets have an upper bound denoted \top and a lower bound \perp . A proper order relation $<$ can be easily derived from \leq when discarding reflexivity. We then denote by \prec the *cover* relation for (E, \leq) such that $x \prec z$ iff there does not exist any element $y \in E$ such that $x < y < z$. Relation \leq is then the transitive and reflexive closure of \prec in E .

To follow on posets, we denote by $\sup(x, y) \triangleq \{z \mid x \leq z, y \leq z\}$ the upper bound of x and y in E , and $\inf(x, y) \triangleq$

$\{z \mid z \leq x, z \leq y\}$. Then, if available, we define the lowest upper bound and the greatest lower bound of those sets:

$$\begin{cases} x \vee y \triangleq \inf \sup(x, y) \\ x \wedge y \triangleq \sup \inf(x, y) \end{cases}$$

Operators \vee and \wedge are both *idempotent*, *commutative* and *associative*. And when they are defined for any pair $(x, y) \in E^2$, then (E, \vee, \wedge) is a well-known algebraic structure called *lattice*. (E, \vee) and (E, \wedge) are then both *semi-lattices*.

The following equivalence supports lattice’s axioms:

$$x \leq y \Leftrightarrow \begin{cases} x \vee y = y \\ x \wedge y = x \end{cases}$$

Finally, we denote by *filter* a closed subset of a lattice under the \wedge operator, and conversely, an *ideal* is a closed subset of E under \vee . $\uparrow(p) \triangleq \{x \mid p \leq x\}$ is the *principal filter* generated by the *principal element* p , and mainly consists in range $[p, \top]$. The dual notion of $\uparrow(p)$ is the *principal ideal* $\downarrow(p) \triangleq \{x \mid x \leq p\}$ generated by p .

2) *A logical perspective*: Properties of the \leq order relation are similar to those from the *entailment* mechanism in any propositional logic (L, \vdash) . The logical consequence (or entailment) reveals true statements in L , called *conclusion*, from other ones, called *premises*, by a deduction mechanism \vdash . In other words, given propositional formula $\phi, \psi \in L$, $\phi \vdash \psi$ states that “ ψ logically follows from ϕ ” or “ ϕ proves ψ ”. From this entailment mechanism, we derive equivalences between logical statements when both $\phi \vdash \psi$ and $\psi \vdash \phi$ hold.

The above *equivalence relation* supports, by the way of the quotient space L / \sim_L , a set of logical statements that make the *Lindenbaum-Tarski algebra* and consequently, it defines a lattice. Then, it allows to map a propositional calculus to an algebra such as the entailment becomes the underlying partial order relation of the lattice¹.

Moreover, deducing all new facts by \vdash from a given premiss in the logical framework, is analogous to achieving the reflexive and transitive closure of the element in the premiss. To this end, it is required to compute principal filter $\uparrow(p)$ of the principal element p in the premiss. By definition, for all x, y in $\uparrow(p)$, we have both $p \leq x$ and $p \leq y$. As an immediate consequence, and given the definition of the (\wedge) operator, property $p \leq x \wedge y$ holds, i.e. $x \wedge y$ belongs to $\uparrow(p)$.

B. Focus on the partition lattice and its algebra

We denote by Ω the finite *ground set* of a partition thereafter mapped to $\mathbb{N}_n = \{1..n\}$. A cluster $c \subseteq \Omega$ is then the short representation of the pairwise associations (up to the equivalence relation) between elements in c . A set partition on Ω is an union of clusters that are both *collectively exhaustive* and *mutually exclusive*. Two special partitions are $\top = \{\Omega\}$ and $\perp = \{\{e\}_{e \in \Omega}\}$. For ease of reading we may not use the set standard writing of partitions but rather ‘|’ as a separator between clusters, such like $P = 12|3|456|78|9$ on \mathbb{N}_9 .

The set of partitions Π_Ω defined on the same ground set Ω is equipped with the refinement relation, defined as follows:

$$P \leq Q \text{ iff } \forall p \in P, \exists q \in Q, p \subseteq q$$

¹We usually call *algebraisation* the process of mapping a class of logical systems with shared connectors and features to a calculus in an algebra.

For instance, given $P = 12|3|45|6$ and $Q = 123|456$, then P is *finer* than Q such that every pairwise association of elements in P is preserved in Q . And one may state that Q “is deduced” from P wherever P holds, either as an assumption in a problem or as an actual output from any processing. Moreover, the refinement relation \leq is independent from the labeling of the ground set Ω . Indeed, given any permutation σ over Ω and two partitions P, Q in Π_Ω , if $P \leq Q$ holds, then $P_\sigma \leq Q_\sigma$ holds as well.

Algebraic operations in the set partition lattice $(\Pi_\Omega, \vee, \wedge)$ are then interpreted as follows:

- $P \wedge Q$ outputs partition M (coined *meet*) such that every cluster of M is *simultaneously* contained both in P and Q ;
- $P \vee Q$ outputs partition J (coined *join*) such that every cluster in P or Q is contained in J .

For instance, given $P = 12|345|67$ and $Q = 123|45|67$, then $P \wedge Q = 12|3|45|67$ and $P \vee Q = 12345|67$.

From the logical point of view, P and Q are assumed to be the *premises* of a usual reasoning mechanism with regular *introduction* rules of connectives. Then, it allows to compute $P \wedge Q$ and $P \vee Q$. Also, the statement $P \wedge Q \leq P \vee Q$ is tautologic.

C. Logical algebraic aspects of the calculus over partition

Given $P = 12|3|4$, $Q = 1|23|4$ and $R = 13|2|4$, it becomes clear that these partitions are all *atomic* propositions, *i.e.* there exists no other partitions refining them, except \perp . The following relationship then occurs:

$$P \vee Q = P \vee R = 123|4$$

Semidistributivity laws (see [6]) for (\vee, \wedge) are defined by:

$$\begin{cases} x \vee y = x \vee z \Rightarrow x \vee y = x \vee (y \wedge z) & (SD_\vee) \\ x \wedge y = x \wedge z \Rightarrow x \wedge y = x \wedge (y \wedge z) & (SD_\wedge) \end{cases}$$

Yet, $P \vee (Q \wedge R) = P \vee \perp \neq 123|4$ infringes (SD_\vee) . That means the partition $123|4$ provides several decomposition under (\vee) , hence there is no single and minimal representation in terms of atomic partitions such as $\forall P \in \Pi_\Omega$:

$$P = \bigvee_{a \in A} a = \bigvee_{b \in B} b \Rightarrow P = \bigvee_{(a,b) \in A \times B} (a \wedge b)$$

Whenever this property is valid in an algebraic structure, this allows for rewriting every element as an *invariant* set of atoms. Otherwise, this entails that a particular element can be equipped with many different explanations. From a logical viewpoint, any partition is the shortened representation of a sequence of aggregation operations. Apart from its maximal representation (see below) which links all clustered elements together, several *models* are then available to provide slightly different *semantics* of the same element.

A non-trivial partition, say $P = 1234|567$, as such is unable then to track down the right ordered sequence of atomic facts that led to it. Looking at first cluster of P , the next two clustering scenarii, $\langle (1,3); (1,4); (2,4) \rangle$ and $\langle (2,3); (1,2); (3,4) \rangle$ are equally plausible while fully decorrelated!

Atoms set of a lattice E are defined by $\mathcal{A}(E) \triangleq \{x \in E \mid \perp \prec x\}$ and similarly coatoms $\mathcal{A}^c(E)$ are elements covered by \top . The Birkhoff’s representation theorem (see [7]) allows for describing elements of a distributive lattice as a set of atomic ones, *i.e.* prime under (\vee) operator and give its *basis*.

$$\begin{cases} x \vee y \Leftrightarrow A_x \cup A_y \\ x \wedge y \Leftrightarrow A_x \cap A_y \end{cases}$$

It then generates the lattice $(\mathcal{A}(E), \subseteq)$ which inherits all algebraic properties of the original lattice (E, \leq) .

Theorem The partition lattice is not distributive [8]

Since the proper representation of a cluster is an equivalence relation (reflexive, symmetric, transitive) over Ω , transitive closure of a set is not preserved under application of set-theoretic union operator. The following formula returns a maximal representation of a whole partition in terms of atoms, though this drops unnecessarily symmetric pair of elements.

$$A_x \triangleq I(x) \cap \mathcal{A}(E)$$

and each atom in A_x subsume x by the order relation.

So, given any non-atomic partition P and its set-theoretic representation A_P , there exists a great number of $B \subsetneq A_P$, each one is probably the *original* model of the partition. The current paradox is that calculus is in principle feasible through set-theoretic operations over Ω^2 even though it depends on assigned model to “read” a partition. Also, achieving a (\vee) operation in the set representation has critical side effects: given two non-atoms P, Q and assuming w.l.o.g. they have several equivalent models, then $A_{P \vee Q} = A_P \cup A_Q \cup X$ where X can be any other set-written partition.

Let’s look back to a logical formalism: if one makes a negated statement, for instance $\neg R$, and from P and Q , inferring $P \wedge Q$, and then $P \vee Q$. But if $A_{\neg R} \subset A_{P \vee Q}$, we are likely to discard this inference. Finally, the partition resulting from ‘ P and Q ’ is true but ‘ P or Q ’ is not!

A solution is then to promote another kind of inference. *Abductive reasoning* is a (non-monotonic) logical framework which roughly addresses the issue of inferring the “best” model from the set of those which all draw the same conclusion, according to some context. This may be used to strip away any ambiguity, namely to reduce side effects when applying (\vee) . Our consensus function presented in the next section, shall merely carry out this principle.

Up to now, we can safely affirm that we are trying to deal with an odd logical system in comparison with the well-established classical propositional logic whose its algebraic counterpart are distributive lattices. Moreover, provided extra operators in such lattices, namely the *implication* (\rightarrow) and the *difference* $(-)$, are defined as *Galois connection* and their computation depend on, (\wedge) and (\vee) :

$$\begin{cases} z \wedge x \leq y \iff z \leq x \rightarrow y \\ x \leq y \vee z \iff x - y \leq z \end{cases}$$

where $x \rightarrow y$ refers to greatest *subtractive* proposition allowing to infer y from x . In a way similar, $x - y$ refers to the smallest *additive* proposition allowing to infer y from x . Algebraically speaking, z is defined as the relative pseudo-complement and there are two for each operators (\vee) and (\wedge) .

III. CONSENSUS PARTITION

The consensus partition we seek should be obtained through basic combinations of input partitions on the lattice, so that is as close as possible to these input partitions. Besides, the efforts required to transform each input partition into this consensus partition should be as similar as possible, in terms of atomic operations.

To the extent that a median element has to be drawn through basic combinations on the lattice, a self-evident way is to seek an element being enough close to its contributors. Also, effort to transform each of these into its median should require a similar number of atomic operations. This is possible whenever the underlying lattice is distributive and also uniquely complemented.

Corollary the partition lattice is not uniquely complemented [8]

A typical example is the powerset lattice $(2^\Omega, \subseteq)$, *i.e.* the lattice of all subsets of a set Ω , that is distributive. There is a bijection between its atoms set $\mathcal{A}(2^\Omega)$, *i.e.* singletons $\{1\}, \{2\}, \{3\}, \dots$, and its coatoms set $\mathcal{A}^c(2^\Omega)$, *i.e.* $\Omega \setminus \{1\}, \Omega \setminus \{2\}, \Omega \setminus \{3\}$, figuring a natural matching between complemented elements.

Conversely, the relative complement of a partition is never achievable regardless of the chosen operator. Apart from trivial cases, there are always several solutions from which one can select a complement, as a suitable algebraic criterion is missing to properly pick a single complement.

From there, there are two alternative solutions:

- explicitly model the set of *contexts* in which a partition is valid;
- “refine” the original definition of extra operators based on (\vee, \wedge) .

For simplicity sake, we retain the second solution. Yet, we discuss below the first approach, on the basis of the Urquhart’s representation theorem, broadening the scope of Birkhoff one to non-distributive lattice [9]

1) *First approach*: The lack of the distributivity axiom implies the vanishing of the duality between conjunctive and disjunctive formulae over lattice elements figured by De Morgan’s inference rules:

$$P \vee Q = \neg(\neg(P \wedge Q))$$

Given that an atomic partition P and its filter $\uparrow(P)$, that together form the subset of Π_Ω which contains all deductible partitions from P , what are its complemented elements in the set $\Pi_\Omega - \uparrow(P)$? Since P is *true*, then any partition $Q \in \Pi_\Omega - \uparrow(P)$ should be *false* since they are orthogonal, *i.e.* $P \wedge Q = \perp$. Then, every element in the closed set $\downarrow(Q)$ is in turn false. Moreover, $\uparrow(P) \cap \downarrow(Q) = \emptyset$ is trivially consistent although $\uparrow(P) \cup \downarrow(Q) \subsetneq \Pi_\Omega$.

It is then suitable to pick every coatom, elements relatively complemented to P , as maximal base points to generate ideals to entirely span Π_Ω . In our case, coatoms are binary partitions that enable enumerating any partition Q that *negates* P . From there, we aim at designing the opposite duality to somewhat

“negate” an already negated partition and capture De Morgan’s rules, following this pattern:

$$P \vee Q = g(f(P \wedge Q))$$

where f and g express two sorts of negation-like operations. Once that duality has been designed, it is possible to build, through each negation, a pair of implication operators $(\rightarrow, \leftarrow)$ and a couple of difference operators $(\setminus, /)$.

2) *Second approach*: As it would appear impossible to build a single solution for one or other equation defining extra operators, we choose to narrow down the solution domain to a subset with the right properties.

The difference operator is commonly defined as:

$$P - Q = \inf\{R \mid P \leq Q \vee R\}$$

This requires computing $P \vee \bigwedge_{s \in S} s$, where S denotes the set of solutions $\{R \mid P \leq Q \vee R\}$. This does not enable choosing a single solution, due to the lack of the distributivity property. A straightforward remedy is to add a further clause enforcing that $R \in 2^P$.

The following properties hold:

$$P \vee Q = \left\{ \begin{array}{l} P \vee (Q - P) \\ Q \vee (P - Q) \end{array} \right\} = (P - Q) \vee (Q - P)$$

The returned partition is a boolean function (0-1) over clusters of P . Preserved clusters are those which are not a strict subset of a cluster of Q . The operation then wholly covers aggregates over both operands and has the same identities that its counterpart in 2^Ω .

There results a similar pattern for the implication operator. By applying the same clause we obtain:

$$P \rightarrow Q = Q - P$$

Using them concurrently is then unnecessary. Given the partition lattice equipped with the implication operator $(\Pi_\Omega, \vee, \wedge, \rightarrow)$, the following properties hold:

- $R \rightarrow (P \vee Q) = (R \rightarrow P) \vee (R \rightarrow Q)$
- $R \rightarrow (Q \rightarrow P) = (Q \rightarrow P) \wedge (R \rightarrow P)$

The second law states (by associativity over \wedge) that iterated application of several implications over P is independent of the order in which they are performed and ensures the parallelization of the computation.

In the partial order context, the following properties hold:

$$P \leq Q \Rightarrow \left\{ \begin{array}{l} R \rightarrow P \leq R \rightarrow Q \\ P \rightarrow R \geq Q \rightarrow R \end{array} \right.$$

This means that implication is an order-preserving operation, whenever they are placed as conclusions and order-reversing if placed as premises.

We shall now apply this operator as a pre-filter on the given set of partitions to be processed and build a consensus partition by applying several algebraic combination over the input partitions.

Arrow [10] describes a set of three axiomatic properties that should be applied by any consensus method. He proved

that these criteria cannot be achieved at the same time (often referred to as the Arrow's paradox). It is widely agreed to consider a relaxed viewpoint on one property or even to infringe one or several ones. These criteria are the following:

- Independence of irrelevant alternatives: two elements $x, y \in \Omega$ must be clustered regardless of other elements in $\Omega - \{x, y\}$ and such clustering process of outlined position on *its* couple by the input partitions in \mathcal{P} ;
- Pareto optimality: if all partitions *unanimously* agree on a couple $x, y \in \Omega$, then it should be also the case in the resulting partition;
- Non-dictatorship: the resulting partition must take into account all the features expressed in every partition and not from a specific subset $\mathcal{P}' \subset \mathcal{P}$, disclaiming aggregates in the partitions from $\mathcal{P} - \mathcal{P}'$.

Let a consensus function is denoted $f : \Pi_{\Omega}^n \rightarrow \Pi_{\Omega}$. A straightforward function which fulfils these prerequisites is the unanimity function:

$$u(\mathcal{P}) =_{def} \bigwedge_{P_i \in \mathcal{P}} P_i$$

Yet, this is likely to return uninformative or inconclusive results as it does not save enough aggregates. Thus, these properties must be relaxed to improve the result. The following function achieves a better trade-off, as it unionizes the set of consensus unanimously computed for each formed *majority* on \mathcal{P} :

$$m(\mathcal{P}) =_{def} \bigvee_{\{S \in 2^{\mathcal{P}} \mid |S| \geq |\mathcal{P}|/2\}} S$$

Our proposal consists in, first, computing, a set \mathcal{P}' so that each included partition is the filtered version of its counterpart in \mathcal{P} . Then, applying implication on each couple of partitions wholly records maximal aggregates, whenever one is strictly included in a bigger one. Denoting \mathcal{P} as a family of sets, \mathcal{P} naturally generates an hypergraph \mathcal{H} whose (hyper)edges are clusters. Given that \mathcal{H}' is processed from \mathcal{P}' , it takes the form of a Sperner family where connectivity depends on the preservation of every edge/cluster:

$$\forall e, f \in \mathcal{H}', e \not\subset f, f \not\subset e$$

Moreover, this preprocessing step thus achieves an abductive inference, providing for each partition a minimal model for the whole set, according to our will to preserve *all* aggregates which appear in at least one partition.

Preserved aggregates in the consensus is the outcome of performing binary intersection between each filtered partition and is then equivalent to computing the intersection between each edge and union those ones in a single partition. Let it be denoted $alg(\mathcal{P})$ (for *algebraic*) and denoted as follows :

$$\begin{aligned} alg(\mathcal{P}) &=_{def} \bigvee_{\{(x,y) \in 2^{\mathcal{P}'}\}} (x \wedge y) \\ \mathcal{P}' &= \{\bigwedge_{\{y_i \in P-x\}} (y_i \rightarrow x) \mid x \in \mathcal{P}\} \end{aligned}$$

Let us illustrate this on a toy example. Given the set of partitions \mathcal{P} :

- $P_1 = 147|2|356$
- $P_2 = 1234|57|6$
- $P_3 = 126|3|47|5$
- $P_4 = 123|4567$
- $P_5 = 124|35|67$

and the multiset $\mathcal{P}_m = \mathcal{P} \cup \{P_1\}$.

Overall outcomes are summarized in the following table:

Partition set	$u(\cdot)$	$m(\cdot)$	$alg(\cdot)$
\mathcal{P}	\perp	12 3 4 5 6 7	12 3 47 56
\mathcal{P}_m	\perp	1247 3 5 6	1247 356

Fig. 1. Comparison of the trial results for each method

Regarding results for the first version of the dataset, our proposal *alg* performs clearly better than the majority function and gave also consistent results regarding the order relation. It also aggregates a greater number of elements and every cluster remain consistent with a subset from \mathcal{P} . It achieves similar results even with duplication of the first partition. It is worth noting that it is because no cluster shall be taken away, then P_1 is freely combinable with filtered clusters of remaining partitions and then we get $alg(\mathcal{P}_m) = p_1 \vee alg(\mathcal{P})$. Moreover, such a behaviour is potentially harmful if the number of duplicate partitions highly grows.

The major advantage of our proposal is that it is more permissive than the majority function and includes in the result more aggregates featured by a minority number of partition *without* discarding those agreed by the remaining ones.

IV. COMPUTATIONAL ISSUES

The partition combination operations described in the previous section call for an effective implementation, when instead of the toy example given above, we reach thousands/millions of clusters/elements in the partitions manipulated. Efficiency of the implementation depends on the data model used to store in-memory several partitions of a set and therefore query features we are interested in. Viewing partitions as a set of subsets of Ω , we essentially need to compute sequences of set-theoretic union and intersection over clusters of their respective operands.

We decide to manage and represent partitions through a system of representatives, each being deterministically chosen among their respective cluster so that an in-memory partition is a Ω -word where query the representative of a set is a $O(1)$ -time operation. Two elements are clustered together in the same partition if their share the same representative. Obviously, while a partition has a single representation, the latter is useless to deduce a particular partition since this mapping is surjective. For instance, $P = 123|45|6789$ is encoded by the Ω -word 111446666.

Performing operations $P \vee Q$ or $P \wedge Q$ generates a new Ω -word, serialized view of the resulting partition and encoding of the \in -relation over Ω . Disregarding local optimization during effective computation of an operator, performing generalized

application of each one, $\bigvee P_i$ and $\bigwedge P_i$, costs $(|\mathcal{P}| - 1) \times O(P \text{ op } Q)$ where $\text{op} \in \{\vee, \wedge\}$.

Considering for instance the majority function, given \mathcal{P} , we have to perform $o(2^{|\mathcal{P}|})$ operations. However, the set \mathcal{P} is likely to be much smaller than Ω . In that case, the overall cost of computing a formula depends more on the cost of a single one rather than the whole.

To characterize the computational cost involved by the present proposal, we report hereunder experimental results operating on randomly drawn partitions. Given two partitions P and Q , we assess the performance of increasing sequences of (\vee) operations. We generate a set of random partitions with a Chinese Restaurant Process (CRP) [11]. As a noteworthy property, the underlying distribution is said to be exchangeable, *i.e.* the probability of a partition only depends on block sizes. The expected number of blocks k grows as $O(\alpha \log n)$, where α is the scale parameter of the CRP. This mechanism is largely used in Bayesian statistical techniques for clustering.

Computation of the (\vee) operator is achieved by a dedicated algorithm based on a tailored version of the so-called Union-Find algorithm, related to connectivity problem in binary graph. We implemented it through the pointer machine model and the merge sets process is mimicked through pointers list defined over set of cluster representatives which enable fast rewriting by means of side effects.

Our protocol is based on incremental computation of an increasing sequence of (\vee) operations. We start from a random partition P_0 with $|\Omega| = 100\,000$ and having 20 clusters, then generate the set $\{P_0, P_1, \dots, P_5\}$ where P_{i+1} is got from P_i by applying hundreds of permutations, without discarding those already applied to generate P_i . Also, this process keeps unchanged the size of clusters and their number in the newly generated partition.

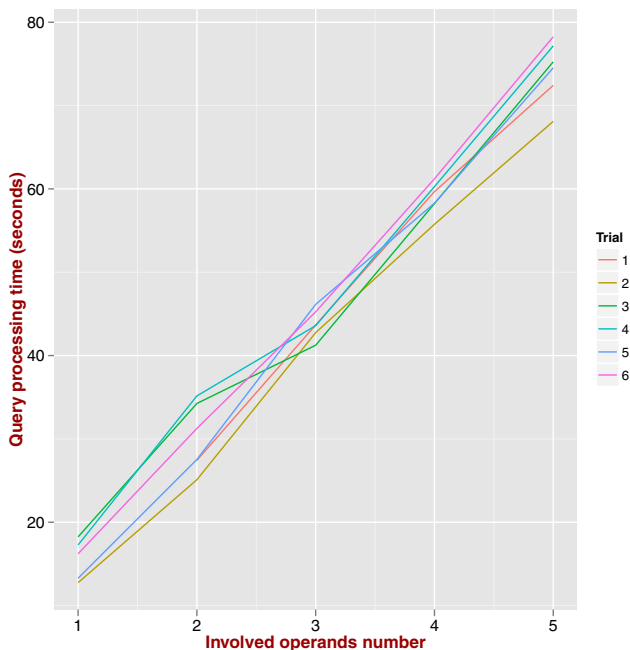


Fig. 2. Several trials for generalized application of (\vee) over $\{P_0, \dots, P_5\}$

Above, we achieved several trial on the same dataset, starting with $P_0 \vee P_1$ and ending with $\bigvee_{0 \leq i \leq 5} P_i$. These results suggest that computation of operators involved in our proposal is not prohibitive and provides grounds for expecting scaling up algebraic formulations of techniques manipulating partitions intensively to large-size partitions.

V. CONCLUSION

In this paper, we have developed a constructivist approach within the context of defining new algebraic operators according to those natural of the partition lattice structure and for which there is no formal framework describing any expected behaviour when combining them or the semantic explanation of the duality binding them.

We have argued for the use of algebraic formulae to achieve a consensus partition. We also briefly outlined a formal framework without making usage of the distributivity, which is a key axiom in the Lindenbaum-Tarski approach.

Explicitly identifying hypothetical contexts in which expressed aggregates within a partition are likely to be true seems a promising way. Indeed, its achievement proceeds from a normalisation step resulting in a loss of valuable information, susceptible to specify the initial context. Then, abductive reasoning would gain ability to outline interrelation between several partitions by means of contexts validating or disproving any particular partition, eventually improving the partition combination process.

REFERENCES

- [1] X. Hu and I. Yoo, "Cluster ensemble and its applications in gene expression analysis," in *Proceedings of the second conference on Asia-Pacific bioinformatics-Volume 29*. Australian Computer Society, Inc., 2004, pp. 297–302.
- [2] Y. Hong, S. Kwong, Y. Chang, and Q. Ren, "Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm," *Pattern Recognition*, vol. 41, no. 9, pp. 2742–2756, 2008.
- [3] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *The Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2003.
- [4] A. Topchy, A. K. Jain, and W. Punch, "Clustering ensembles: Models of consensus and weak partitions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 12, pp. 1866–1881, 2005.
- [5] J. G. Kemeny, "Mathematics without numbers," *Daedalus*, vol. 88, no. 4, pp. 577–591, 1959.
- [6] Y. Rav, "Semiprime ideals in general lattices," *Journal of Pure and Applied Algebra*, vol. 56, no. 2, pp. 105 – 118, 1989. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0022404989901400>
- [7] G. Birkhoff, "Rings of sets," *Duke Mathematical Journal*, vol. 3, pp. 443–454, 1937.
- [8] O. Ore, "Theory of equivalence relations," *Duke Math. J.*, vol. 9, no. 3, pp. 573–627, 1942.
- [9] A. Urquhart, "A topological representation theory for lattices," *Algebra Universalis*, vol. 8, no. 1, pp. 45–58, 1978.
- [10] K. J. Arrow, *Social Choice and Individual Values*. New York, NY: John Wiley and Sons, 1951.
- [11] S. Goldwater, T. L. Griffiths, and M. Johnson, "Producing power-law distributions and damping word frequencies with two-stage language models," *J. Mach. Learn. Res.*, vol. 12, pp. 2335–2382, Jul. 2011. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2021026.2021075>