Maximal Closed Set and Half-Space Separations in Finite Closure Systems*

Florian Seiffarth SEIFFARTH@CS.UNI-BONN.DE
Dept. of Computer Science, University of Bonn, Germany

Tamás Horváth HORVATH@CS.UNI-BONN.DE
Dept. of Computer Science, University of Bonn, Germany
Fraunhofer IAIS and Fraunhofer Center for Machine Learning,
Schloß Birlinghoven, Sankt Augustin, Germany

Stefan Wrobel WROBEL@CS.UNI-BONN.DE
Dept. of Computer Science, University of Bonn, Germany
Fraunhofer IAIS and Fraunhofer Center for Machine Learning,
Schloß Birlinghoven, Sankt Augustin, Germany

Abstract

Several problems of artificial intelligence, such as predictive learning, formal concept analysis or inductive logic programming, can be viewed as a special case of half-space separation in abstract closure systems over finite ground sets. For the typical scenario that the closure system is given via a closure operator, we show that the half-space separation problem is NP-complete. As a first approach to overcome this negative result, we relax the problem to maximal closed set separation, give a greedy algorithm solving this problem with a linear number of closure operator calls, and show that this bound is sharp. For a second direction, we consider Kakutani closure systems and prove that they are algorithmically characterized by the greedy algorithm. As a first special case of the general problem setting, we consider Kakutani closure systems over graphs, generalize a fundamental characterization result based on the Pasch axiom to graph structured partitioning of finite sets, and give a sufficient condition for this kind of closures systems in terms of graph minors. For a second case, we then focus on closure systems over finite lattices, give an improved adaptation of the greedy algorithm for this special case, and present two applications concerning formal concept and subsumption lattices. We also report some experimental results to demonstrate the practical usefulness of our algorithm.

1. Introduction

Several problems of *artificial intelligence* (AI), such as predictive learning, formal concept analysis or inductive logic programming, can be viewed as a special case of *half-space separation* in abstract closure systems over finite ground sets. The theory of binary separation in $\mathbb{R}^d$ by hyperplanes goes back at least to the pioneer work of Rosenblatt (1958) on perceptron learning in the late fifties. Since then several deep results have been published on this topic, including Vapnik and his co-workers seminal paper on support vector machines (Boser, Guyon, & Vapnik, 1992). Given two finite subsets of $\mathbb{R}^d$, the problem

∗. A short version of this paper has appeared in the post-conference proceedings of *Machine Learning and Knowledge Discovery in Databases - European Conference*, ECML PKDD 2019 (see https://link.springer.com/chapter/10.1007%2F978-3-030-46150-8_2).

whether or not they are separable by a hyperplane can be decided by checking the disjointness of their convex hulls. The correctness of this generic method for $\mathbb{R}^d$ follows from the result of Kakutani (1937) that any two disjoint convex sets in $\mathbb{R}^d$ are always separable by a hyperplane.

While hyperplane separability in $\mathbb{R}^d$ is a well-studied problem, its adaptation to non-standard data, such as graphs and other relational and algebraic structures has received less attention in artificial intelligence. This is somewhat surprising, as several problems in AI, including e.g. machine learning, formal concept analysis, and inductive logic programming (see Section 3 for some illustrative examples), can be viewed as special cases of half-space separation in finite closure systems. In contrast, several results concerning different formal properties of *abstract* half-spaces over finite domains have been published in *geometry* and *theoretical computer science* (see, e.g., Chepoi, 1994; Ellis, 1952; Kubiś, 2002; van de Vel, 1984). Using the fact that the family of convex hulls in $\mathbb{R}^d$ forms a *closure system*, the underlying idea of adapting hyperplane separation in $\mathbb{R}^d$ to arbitrary finite sets E is to consider some semantically meaningful closure system $\mathcal{C}$ over E. A subset H of E is considered as an *abstract* half-space, if H and its complement both belong to $\mathcal{C}$. In this field of research there is a distinguished focus on characterization results of *Kakutani closure systems* (see, e.g., Chepoi, 1994; van de Vel, 1993). This kind of closure systems satisfy the property that any two subsets of the ground set are half-space separable in the closure system if and only if their closures are disjoint.

Utilizing the results of other research fields (Chepoi, 1994; Ellis, 1952; Kubiś, 2002; van de Vel, 1984), in this work we study the *algorithmic* aspects of half-space separation in finite closure systems and discuss some potential AI *applications* of this problem. We assume that the closure systems are given implicitly via a *closure operator*. This assumption is justified by the fact that their cardinality can be exponential in that of the domain. We regard the closure operator as an *oracle* (or black box) which returns the closure for any subset of the domain in *unit time*. We show that under these assumptions, it is NP-complete to decide whether two subsets of the ground set are half-space separable in the underlying abstract closure system. We therefore relax the problem to *maximal* closed set[1] separation. That is, we are interested in finding two closed sets that are disjoint, contain the two input sets, and have no supersets in the closure system w.r.t. these properties.

For this relaxed problem we give an efficient *greedy* algorithm and show that it is *optimal* w.r.t. the number of closure operator calls in the worst-case. As a second approach to resolve the negative complexity result above, we then focus on Kakutani closure systems. We first show that any algorithm deciding whether a closure system is Kakutani requires exponentially many closure operator calls in the worst-case. Still, Kakutani closure systems remain highly interesting because there are various closure systems that are known to be Kakutani. We also prove that our greedy algorithm provides an algorithmic characterization of Kakutani closure systems. This implies that for such systems, the output is always a partitioning of the domain containing the input sets if and only if their closures are disjoint.

Regarding potential applications in AI, we consider closure systems over *graphs* and *lattices*. In particular, for graphs we consider the closure operator over vertices induced by shortest paths (Farber & Jamison, 1986) and generalize first a fundamental characterization

1. Throughout this work we consistently use the nomenclature "closed sets" by noting that "convex" and "closed" are synonyms by the standard terminology of this field.

result of this kind of Kakutani closure systems that is based on the *Pasch axiom* (Chepoi, 1994) to graph structured partitioning of finite sets. Potential practical applications of this more general result include *graph clustering* (see, e.g., Schaeffer, 2007) and *graph partitioning* (see, e.g., Buluc, Meyerhenke, Safro, Sanders, & Schulz, 2016). Although the Pasch axiom allows for a polynomial time naive algorithm for deciding whether a closure system over graphs is Kakutani or not, the algorithm is practically infeasible even for small graphs. As a second result concerning graphs, we therefore show that closure systems over graphs induced by shortest paths are Kakutani if they do not contain the bipartite clique $K_{2,3}$ as a minor. The converse of this claim is, however, not true. This result, together with the characterization result of Chartrand and Harary (1967) immediately implies that closures systems over *outerplanar* graphs and hence, over trees are Kakutani.

Regarding *lattices*, we present an adaptation of the greedy algorithm which calculates a disjoint maximal ideal and filter that contain the input two sets. It has several algorithmic advantages over the original greedy algorithm. In particular, it utilizes the facts that the current *ideal* and *filter* in each iteration can be represented by its *supremum* and *infimum*, respectively. Furthermore, their disjointness can be decided by comparing these two elements. For the special case that the elements of the lattice are subsets of some finite ground set (e.g., concept lattices Ganter, Stumme, & Wille, 2005), the number of closure operator calls is only quadratic in the cardinality of the domain, implying an exponential speed-up over the original greedy algorithm. In addition to these results, we also show that the adapted algorithm preserves the characterization property of the generic greedy algorithm, i.e., it provides an *algorithmic* characterization of Kakutani closure systems over finite lattices. This result is somewhat orthogonal to the characterization given in terms of *distributivity* (see, e.g., Kubiś, 2002).

Besides the results discussed above, we present some illustrative *experimental* results obtained by the greedy algorithm in Kakutani and non-Kakutani closure systems over finite subsets of $\mathbb{R}^d$ as well as over graphs. The results of our systematic evaluations on various artificial datasets clearly demonstrate that surprisingly high accuracy and coverage values can be obtained even for non-Kakutani closure systems. Since our primary goal was to study the predictive performance of our *general* purpose greedy algorithm, we deliberately have *not* exploited any domain specific properties in these experiments. Accordingly, we have not compared our results to those of the state-of-the-art domain specific algorithms.

The rest of the paper is organized as follows. In Section 2 we collect the necessary notions and fix the notation. In Section 3 we define the problem settings, discuss some potential AI applications, and study the complexity issues of half-space and maximal closed set separation in abstract closure systems over finite domains. Section 4 is devoted to Kakutani closure systems. In Sections 5 and 6 we present applications to closure systems over graphs and lattices, respectively. Our experimental results are reported in Section 7. Finally, in Section 8 we formulate some problems for further research.

2. Preliminaries

In this section we collect the necessary notions and notation for set and closure systems (for good references on closure systems and separation axioms see, e.g., Chepoi, 1994; Davey & Priestley, 2002; van de Vel, 1993).

Closure Systems For a set E, 2^E denotes the power set of E. A *set system* over a ground set E is a pair $(E, \mathcal{C})$ with $\mathcal{C} \subseteq 2^E$; $(E, \mathcal{C})$ is a *closure system* if it fulfills the following properties:

- $E \in \mathcal{C}$ and
- $X \cap Y \in \mathcal{C}$ for all $X, Y \in \mathcal{C}$.

Throughout this paper by closure systems we always mean closure systems over *finite* ground sets (i.e., $|E| < \infty$). It is a well-known fact (see, e.g., Davey & Priestley, 2002) that any closure system can be defined by a *closure operator*, i.e., function $\rho : 2^E \to 2^E$ satisfying the following properties for all $X, Y \subseteq E$:

- $X \subseteq \rho(X)$, (*extensivity*)
- $\rho(X) \subseteq \rho(Y)$ whenever $X \subseteq Y$, (*monotonicity*)
- $\rho(\rho(X)) = \rho(X)$. (*idempotency*)

For a closure system $(E, \mathcal{C})$, the corresponding closure operator ρ is defined by

$$\rho(X) = \bigcap\{C \in \mathcal{C} : X \subseteq C\}$$

for all $X \subseteq E$. Conversely, for a closure operator ρ over E the corresponding closure system, denoted $(E, \mathcal{C}_\rho)$, is defined by the family of its *fixed points*, i.e.,

$$\mathcal{C}_\rho = \{X \subseteq E : \rho(X) = X\}.$$

Depending on the context we sometimes omit the underlying closure operator from the notation and denote the closure system simply by $(E, \mathcal{C})$. The elements of $\mathcal{C}_\rho$ of a closure system $(E, \mathcal{C}_\rho)$ will be referred to as *closed* or *convex* sets. This latter terminology is justified by the fact that closed sets generalize several properties of *convex hulls* in $\mathbb{R}^d$. As a straightforward example, for any finite set $E \subset \mathbb{R}^d$, the set system $(E, \mathcal{C}_\alpha)$ with $\alpha : 2^E \to 2^E$ defined by

$$\alpha : X \mapsto \mathrm{conv}(X) \cap E \tag{1}$$

for all $X \subseteq E$ is a closure system, where $\mathrm{conv}(X)$ denotes the convex hull of X in $\mathbb{R}^d$. We will refer to this type of closure systems as *α-closure systems*.

Separation in Closure Systems We now turn to the generalization of binary separation in $\mathbb{R}^d$ by hyperplanes to that in *abstract* closure systems (for a detailed introduction into this topic see, e.g., van de Vel, 1993). In the context of *machine learning*, one of the most relevant and natural questions concerning closure systems $(E, \mathcal{C})$ is whether two subsets of E are separable in $\mathcal{C}$, or not. To state the formal problem definition, we follow the generalization of half-spaces in Euclidean spaces to closure systems from Chepoi (1994). More precisely, let $(E, \mathcal{C})$ be a closure system. Then $H \subseteq E$ is called a *half-space* in $\mathcal{C}$ if both H and its complement, denoted H^c, are closed (i.e., $H, H^c \in \mathcal{C}$). Note that H^c is also a half-space by definition. Two sets $A, B \subseteq E$ are *half-space separable* if there is a half-space $H \in \mathcal{C}$ such that $A \subseteq H$ and $B \subseteq H^c$; H and H^c together form a *half-space separation* of A and B. The following property will be used many times in what follows:

Proposition 1. *Let* $(E, \mathcal{C}_\rho)$ *be a closure system,* $H, H^c \in \mathcal{C}$*, and* $A, B \subseteq E$*. Then* H *and* H^c *form a half-space separation of* A *and* B *if and only if they form a half-space separation of* $\rho(A)$ *and* $\rho(B)$*.*

Proof. The "if" direction is immediate by the extensivity of ρ. The "only-if" direction follows from the fact that for any $S \subseteq E$ and $C \in \mathcal{C}_\rho$ with $S \subseteq C$ we have $\rho(S) \subseteq \rho(C) = C$ by the monotonicity and idempotency of ρ. □

Throughout this paper we will be concerned with half-space separation of *non-empty* subsets of the ground set. Proposition 2 below provides a necessary condition for this problem. Its proof is immediate from the property that $\mathcal{C}$ is closed under intersection.

Proposition 2. *Let* $(E, \mathcal{C})$ *be a closure system and* A, B *be non-empty subsets of* E *that are half-space separable in* $\mathcal{C}$*. Then* $\rho(\emptyset) = \emptyset$*.*

Notice that half-space separability in abstract closure systems does not preserve all natural properties of that in $\mathbb{R}^d$. For example, for any two *finite* subsets of $\mathbb{R}^d$ it always holds that they are half-space separable if and only if their convex hulls[2] are disjoint. In contrast, as we show in Section 3.1 below, this equivalence does not hold for finite closure systems in general.

3. Half-Space and Maximal Closed Set Separation in Closure Systems

Our goal in this work is to investigate the algorithmic aspects of half-space and maximal closed set separations in abstract closure systems. To motivate these two general problem settings, below we have selected three illustrative problems from different fields of AI. We show that they all deal with maximal or half-space separation in abstract closure systems.

(i) (*machine learning*) Our first example is concerned with predictive learning in *graphs*. More precisely, suppose all vertices of a graph are colored by one of two colors, say red and blue, but the colors are known only for a subset of the vertices. The task is to predict the unknown color for *all* uncolored vertices. Clearly, if we have no further information about the problem, then there is no chance to improve the predictive performance of random guessing. Suppose we are provided with the additional knowledge about the fully colored graph that for all pairs of monochromatic vertices, all shortest paths connecting them are also monochromatic. Then, utilizing the folklore result that this kind of "convexity" gives rise to a closure operator (see, e.g., Farber & Jamison, 1986), the problem above can be regarded as a special case of half-space separation in the corresponding abstract closure system. If the convexity property holds for one of the two colors only, then the problem becomes a special case of the maximal closed set separation problem, independently whether the color is known, or not.

(ii) (*formal concept analysis*) For the second application example, consider the following problem concerning *formal concepts* (Ganter et al., 2005): Given disjoint sets $\mathcal{C}_1$ and

2. Notice that the function mapping any subset of $\mathbb{R}^d$ to its convex hull is a closure operator.

$\mathcal{C}_2$ of concepts, find two concepts C_1 and C_2 such that C_1 *generalizes*[3] all concepts in $\mathcal{C}_1$, but no concept in $\mathcal{C}_2$ and C_2 *specializes* all concepts in $\mathcal{C}_2$, but no concept in $\mathcal{C}_1$, or vice versa. Furthermore, C_1, C_2 need to be *maximal* with respect to this property. If there are no such C_1 and C_2, then the algorithm is required to return the answer "No". That is, we are interested in finding two maximal "meta-concepts" (i.e., which represent sets of concepts) separating $\mathcal{C}_1$ from $\mathcal{C}_2$. For this problem, one can consider the closure system formed by the set of *maximal sublattices* of the concept lattice and regard the problem as finding a maximal closed set separation of C_1 and C_2 in that system.

(iii) (*inductive logic programming*) Our third motivating example deals with generalization and specialization of *first-order clauses*. More precisely, one of the most common problems in inductive logic programming (ILP) (Nienhuys-Cheng & Wolf, 1997) is defined as follows: Given a set of positive and a set of negative first-order clauses[4], return a first-order clause that subsumes (or generalizes) all positive and none of the negative clauses, if such a clause exists; otherwise return "No". It follows from the seminal results of Plotkin (1970) that such a clause exists if and only if the *least general generalization* of the positive clauses does not subsume any of the negative ones. Equivalently, the algorithm is required to return the supremum of the *smallest ideal* in the subsumption lattice that contains all positive examples if it is disjoint with the set of negative clause; o/w the answer "No". In contrast to this classical problem setting, our algorithm selects a finite sublattice of the subsumption lattice spanned by certain specialization and generalization of the input clauses and consider the set system defined by the set of ideals and filters of this lattice. Since it is a closure system, the separation of the two clause sets above can be regarded as another special case of the maximal closed set separation problem. Regarding the solution of the two problems, there are two crucial differences. While the traditional ILP problem above treats the positive and negative examples *asymmetrically* (i.e., considers the smallest ideal containing the *positive* examples), in the maximal closed set separation problem the two clause sets are regarded *symmetrically* (i.e., it allows the solution to consist of a generalization of the *negative* and a specialization of the *positive* examples as well). Thus, as we show in Section 6, the maximal closed set separation problem can have a solution also for such problem instances where there is no consistent hypothesis according to the traditional ILP problem setting.

3.1 Half-Space Separation

In this section we formulate some results concerning the computational complexity of the following decision problem:

Half-Space Separation (HSS) Problem: *Given* a closure system $(E, \mathcal{C}_\rho)$ with $|E| < \infty$ via ρ and non-empty subsets $A, B \subseteq E$, *decide* whether A and B are half-space separable in $\mathcal{C}_\rho$, or not.

3. A concept C generalizes (resp. specializes) a concept C' if its set of objects contains (resp. is contained by) that of C'.

4. In case of logic programming, clauses are restricted to program clauses, where the body may contain negative literals as well (cf. Lloyd, 1987).

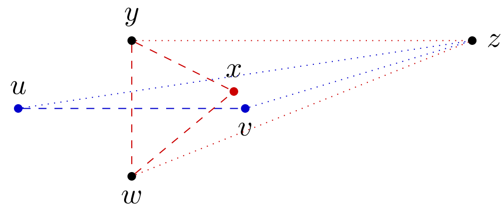


Figure 1: Example of a point configuration in $\mathbb{R}^2$, where adding z to either of the closed sets $\{x, y, w\}$ and $\{u, v\}$ would violate the disjointness condition.

For algebraic reason we disregard the degenerate case of $A = \emptyset$ or $B = \emptyset$. Furthermore, similarly to the infinite closure system over $\mathbb{R}^d$ defined by the family of all convex hulls in $\mathbb{R}^d$, we suppose that the (abstract) closure system is given *implicitly*. More precisely, we assume that $(E, \mathcal{C}_\rho)$ is given by the corresponding closure operator ρ, which returns $\rho(X)$ for any $X \subseteq E$ in *unit* time. Accordingly, we characterize the complexity of the algorithms by the number of closure operator calls they require. The assumption that $\mathcal{C}_\rho$ is given implicitly (or intensionally) is natural, as $|\mathcal{C}_\rho|$ can be exponential in $|E|$.

Clearly, the answer for an instance of the HSS problem is always "No" whenever $\rho(A) \cap \rho(B) \neq \emptyset$. However, as shown in the example below, the converse of the implication is not true, i.e., the disjointness of the closures of A and B does not imply their half-space separability in $\mathcal{C}$.

Example 1. *Consider the set $E \subset \mathbb{R}^2$ consisting of the seven points in Fig. 1 and the α-closure system $(E, \mathcal{C}_\alpha)$ defined in (1). Though $\{u, v\}$ and $\{x, y, w\}$ are both closed (i.e., belong to $\mathcal{C}_\alpha$) and disjoint, they are not half-space separable in $\mathcal{C}_\alpha$, as z can be added to neither of the sets without violating the disjointness property of half-space separation.*

This difference to $\mathbb{R}^d$ makes, among others, the more general problem setting considered in this work computationally difficult, as shown in Theorem 2 below. The fact that the disjointness of $\rho(A)$ and $\rho(B)$ does not imply the half-space separability of A and B makes the HSS problem computationally intractable. To prove this negative complexity result, we adopt the definition of *convex* vertex sets of a graph defined by shortest paths (Farber & Jamison, 1986). More precisely, for an undirected graph $G = (V, E)$ we consider the set system $(V, \mathcal{C}_\gamma)$ with

$$V' \in \mathcal{C}_\gamma \iff \forall u, v \in V', \forall P \in \mathcal{S}_{u,v} : V(P) \subseteq V' \tag{2}$$

for all $V' \subseteq V$, where $\mathcal{S}_{u,v}$ denotes the set of shortest paths connecting u and v in G and $V(P)$ the set of vertices in P. Notice that $(V, \mathcal{C}_\gamma)$ is a closure system. This follows directly from the fact that the intersection of any two convex subsets of V is also convex, by noting that the empty set is also convex by definition. This type of closure systems will be referred to as γ-*closure systems* throughout this paper. Using the above definition of graph convexity, we consider the following problem:

Convex 2-Partitioning Problem: *Given* an undirected graph $G = (V, E)$, *decide* whether there is a *proper* partitioning of V into two convex sets.

Clearly, the condition on properness is necessary, as otherwise $\emptyset$ and V would always form a (trivial) solution. Note also the difference between the HSS and the Convex 2-Partitioning problems that the latter one is concerned with a property of G (i.e., has no additional input A, B). For the problem above, the following negative result has been shown by Artigas, Dantas, Dourado, and Szwarcfiter (2011):

Theorem 1. *The* Convex 2-Partitioning *problem is NP-complete.*

Using the concepts and the result above, we are ready to prove the following negative result:

Theorem 2. *The* HSS *problem is* NP*-complete.*

Proof. The problem is in NP because for any $A, B, H \subseteq E$, one can verify in time linear in $|E|$, whether H and H^c form a half-space separation of A and B in $\mathcal{C}$, or not. We prove the NP-hardness by reduction from the Convex 2-Partitioning problem defined above. Let $G = (V, E)$ be an instance of the Convex 2-Partitioning problem and γ be the closure operator corresponding to the closure system defined in (2). It holds that G has a proper convex 2-partitioning if and only if there are $u, v \in V$ with $u \neq v$ such that $\gamma(\{u\})$ and $\gamma(\{v\})$ are half-space separable in $(V, \mathcal{C}_\gamma)$. Indeed, if G has a proper convex 2-partitioning then there exist $u, v \in V$ belonging to different convex partitions. Since the two convex partitions are (complementary) half-spaces in $(V, \mathcal{C}_\gamma)$, $\{u\}$ and $\{v\}$ are half-space separable in $(V, \mathcal{C}_\gamma)$. Conversely, if there are $u, v \in V$ such that $\{u\}$ and $\{v\}$ are half-space separable in $(V, \mathcal{C}_\gamma)$, then the corresponding half-spaces form a proper convex 2-partitioning of G. Putting together, the Convex 2-Partition problem can be decided by solving the HSS problem for the input $(V, \mathcal{C}_\gamma)$, $A = \{u\}$, and $B = \{v\}$ for all $u, v \in V$. This completes the proof, as the number of vertex pairs is quadratic in the size of G. □

Theorem 2 immediately implies the following negative result on computing a closed set separation of *maximum* size. (Note that the case of $k = |E|$ in the corollary below corresponds to the HSS problem.)

Corollary 1. *Given a closure system $(E, \mathcal{C}_\rho)$ via ρ as in the HSS problem definition, non-empty subsets $A, B \subseteq E$, and an integer $k > 0$, it is NP-complete to decide whether there are disjoint closed sets $H_1, H_2 \in \mathcal{C}_\rho$ with $A \subseteq H_1$, $B \subseteq H_2$ such that $|H_1| + |H_2| \geq k$.*

The negative results above motivate us to relax the HSS problem.

3.2 Maximal Closed Set Separation

One way to overcome the negative results formulated in Theorem 2 and Corollary 1 is to weaken the condition on half-space separability in the HSS problem to the problem of *maximal* closed set separation:

Maximal Closed Set Separation (MCSS) Problem: *Given* a closure system $(E, \mathcal{C}_\rho)$ via ρ as in the HSS problem definition and non-empty subsets $A, B \subseteq E$, *find* two disjoint closed sets $H_1, H_2 \in \mathcal{C}_\rho$ with $A \subseteq H_1$ and $B \subseteq H_2$ such that there are no disjoint sets $H_1', H_2' \in \mathcal{C}_\rho$ with $H_1 \subseteq H_1'$ and $H_2 \subseteq H_2'$, where at least one of the containments is proper; o/w return "NO" (i.e., if such closed sets do not exist).

Algorithm 1: MAXIMAL CLOSED SET SEPARATION (MCSS)

Input: finite closure system $(E, \mathcal{C}_\rho)$ given by ρ and $A, B \subseteq E$ with $A, B \neq \emptyset$
Output: *maximal* disjoint closed sets $H_1, H_2 \in \mathcal{C}_\rho$ with $A \subseteq H_1$ and $B \subseteq H_2$ if $\rho(A) \cap \rho(B) = \emptyset$; "NO" o/w

1 $H_1 \leftarrow \rho(A)$, $H_2 \leftarrow \rho(B)$
2 **if** $H_1 \cap H_2 \neq \emptyset$ **then**
3   **return** "NO"
4 **end**
5 $F \leftarrow E \setminus (H_1 \cup H_2)$
6 **while** $F \neq \emptyset$ **do**
7   choose $e \in F$ and remove it from F
8   **if** $\rho(H_1 \cup \{e\}) \cap H_2 = \emptyset$ **then**
9         $H_1 \leftarrow \rho(H_1 \cup \{e\})$, $F \leftarrow F \setminus H_1$
10   **else if** $\rho(H_2 \cup \{e\}) \cap H_1 = \emptyset$ **then**
11         $H_2 \leftarrow \rho(H_2 \cup \{e\})$, $F \leftarrow F \setminus H_2$
12   **end**
13 **end**
14 **return** H_1, H_2

In this section we present Algorithm 1, that solves the MCSS problem and is optimal w.r.t. the worst-case number of closure operator calls. Algorithm 1 takes as input a closure system $(E, \mathcal{C}_\rho)$ over some finite ground set E, where $\mathcal{C}_\rho$ is given via the closure operator ρ, and non-empty sets $A, B \subseteq E$. If the closures of A and B are not disjoint, then it returns "NO" (cf. Lines 1–3). Otherwise, the algorithm tries to extend one of the largest closed sets $H_1 \supseteq A$ and $H_2 \supseteq B$ found so far consistently by an element $e \in F$, where $F = E \setminus (H_1 \cup H_2)$ is the set of potential generators. By consistency we mean that the closure of the extended set must be disjoint with the other (unextended) one (cf. Lines 8 and 10). Note that each element will be considered at most once for extension (cf. Line 5). If H_1 or H_2 could be extended, then F will correspondingly be updated (cf. Lines 9 and 11), by noting that e will be removed from F even in the case it does not result in an extension (cf. Line 5). The algorithm repeatedly iterates the above steps until F becomes empty; at this stage it returns H_1 and H_2 as a solution. We have the following result for Algorithm 1:

Theorem 3. *Algorithm 1 is correct and solves the MCSS problem by calling the closure operator at most* $2|E| - 2$ *times.*

Proof. The correctness is straightforward. The maximality of the output H_1 and H_2 follows from the monotonicity of ρ, as all elements e considered by the algorithm and not added earlier to one of the closed sets (cf. Lines 9 and 11) can be added later neither to H_1 nor to H_2 without violating the disjointedness.

Regarding the complexity, the algorithm calls initially the closure operator twice (cf. Line 1) and then at most twice per iteration (cf. Lines 9 and 11), giving the upper bound

$$2 \cdot |E \setminus (\rho(A) \cup \rho(B))| + 2 \ .$$

The claim then follows from the case that A and B are closed singletons. □

We stress that Algorithm 1 has access to $(E, \mathcal{C}_\rho)$ only via ρ, i.e., it does not utilize any domain specific properties. The following example shows that, under this assumption, the number of closure operator calls may depend on the order of A and B and on that of the elements selected in Line 7.

Example 2. *Let $(E, \mathcal{C}_\rho)$ be the closure system with $E = \{1, 2, \ldots, n\}$ for some $n > 1$ integer and with the corresponding closure operator defined by $\rho : X \mapsto \{x \in E : \min X \leq x \leq \max X\}$ for all $X \subseteq E$. Consider first the case that $A = \{2\}$, $B = \{1\}$, and n has been chosen in Line 7 for the first iteration. For this case, the algorithm terminates after the first iteration returning the closed half-spaces $H_1 = \{2, \ldots, n\}$ and $H_2 = \{1\}$, and calling the closure operator together three times.*

Now consider the case that $A = \{1\}$, $B = \{2\}$, and the elements in Line 7 are processed in the order $3, 4, \ldots, n$. One can easily check that the algorithm returns the same two half-spaces after $n - 2$ iterations. In each iteration it calls the closure operator twice, giving together the worst-case upper bound $2n - 2$ claimed in Theorem 3.

Though the example above may suggest that Algorithm 1 is not optimal, the worst-case bound stated in Theorem 3 is in fact the best possible, regardless of the order of the elements in Line 7. To obtain this result, we first show the following lemma.

Lemma 1. *There exists no algorithm solving the MCSS problem calling the closure operator less than $2|E| - 2$ times in the worst-case.*

Proof. Suppose for contradiction that there is an algorithm $\mathfrak{A}$ solving the MCSS problem with strictly less than $2|E| - 2$ closure operator calls for *all* problem instances. Consider the closure system $(E, \mathcal{C}_\rho)$ with $E = \{e_1, e_2, \ldots, e_n\}$ for some $n > 2$ and with the closure operator ρ defined by

$$\rho(X) = \begin{cases} X & \text{if } X \in \{\emptyset, \{e_1\}, \{e_2\}\} \\ E & \text{o/w} . \end{cases}$$

By condition, $\mathfrak{A}$ returns the only solution $\{e_1\}$ and $\{e_2\}$ of the MCSS problem for the input $\{e_1\}$ and $\{e_2\}$ with at most $2|E| - 3$ closure operator calls. We claim that $\mathfrak{A}$ needs to calculate the closure for both input sets. Indeed, suppose the closure of one of them, say $\{e_1\}$, has not been computed. Then $\mathfrak{A}$ is incorrect, as it would return the same output for the closure system above and for $(E, \mathcal{C}_\rho \setminus \{e_1\})$. Thus, $\mathfrak{A}$ can calculate the closure for at most $2|E| - 5$ further subsets of E. This implies that the closure has not been considered by $\mathfrak{A}$ for at least one of the sets $\{e_1, e_3\}, \ldots, \{e_1, e_n\}, \{e_2, e_3\}, \ldots, \{e_2, e_n\}$, say for $\{e_1, e_3\}$. But then $\mathfrak{A}$ returns the same output for $(E, \mathcal{C}_\rho)$ and for the closure system $(E, \mathcal{C}_\rho \cup \{e_1, e_3\})$, contradicting its correctness. □

Theorem 4. *Algorithm 1 is optimal w.r.t. the worst-case number of closure operator calls.*

Proof. It is immediate from Theorem 3 and Lemma 1. □

In Section 4 we consider *Kakutani* closure systems, a special kind of closure systems, for which a half-space separation always exists if the closures of the input sets are disjoint. We will show that for this type of closure systems, Algorithm 1 provides an algorithmic characterization and solves the HSS problem correctly and efficiently.

4. Kakutani Closure Systems

A natural way to overcome the negative result stated in Theorem 2 is to consider closure systems in which *any* two disjoint closed sets are half-space separable. More precisely, for a closure operator ρ over a ground set E, the corresponding closure system $(E, \mathcal{C}_\rho)$ is *Kakutani*[5] if it fulfills the S_4 *separation axiom* defined as follows: For all $A, B \subseteq E$,

$$A \text{ and } B \text{ are half-space separable in } (E, \mathcal{C}_\rho) \iff \rho(A) \cap \rho(B) = \emptyset$$

(for a good reference on closure systems satisfying the S_4 separation property see, e.g., Chepoi, 1994). By Proposition 1, any half-space separation of A, B in $\mathcal{C}_\rho$ is a half-space separation of $\rho(A)$ and $\rho(B)$ in $\mathcal{C}_\rho$. Clearly, the HSS problem can be decided in linear time for Kakutani closure systems: For any $A, B \subseteq E$ just calculate $\rho(A)$ and $\rho(B)$ and check whether they are disjoint, or not.

Example 3. *The closure system used in Example 1 is not Kakutani. For an example of Kakutani closure systems, consider an arbitrary non-empty finite subset $E \subset \mathbb{R}^2$ of a circle and define the set system $\mathcal{C} \subseteq 2^E$ as follows: For all $E' \subseteq E$, $E' \in \mathcal{C}$ if and only if there exits a closed half-plane $H \subseteq \mathbb{R}^2$ satisfying $E' = H \cap E$. One can easily check that $(E, \mathcal{C})$ is a Kakutani closure system.* □

In the theorem below we show that Algorithm 1, besides solving the MCSS problem, also provides an *algorithmic characterization* of Kakutani closure systems.

Theorem 5. *Let $(E, \mathcal{C}_\rho)$ be a closure system with corresponding closure operator ρ. Then $(E, \mathcal{C}_\rho)$ is Kakutani if and only if for all non-empty $A, B \subseteq E$ with $\rho(A) \cap \rho(B) = \emptyset$, the output of Algorithm 1 is a partitioning of E.*

Proof. The sufficiency is immediate by Theorem 3 and the definition of Kakutani closure systems. For the necessity, let $(E, \mathcal{C}_\rho)$ be a Kakutani closure system. It suffices to show that for all $e \in F$ selected in Line 7 of Algorithm 1, e is always added to one of H_1 or H_2; the claim then follows by Theorem 3 for this direction. Suppose for contradiction that there exists an $e \in F$ selected in Line 7 that can be used to extend neither of the closed sets H_1, H_2. Since H_1 and H_2 are disjoint closed sets and $(E, \mathcal{C}_\rho)$ is Kakutani, there are $H_1', H_2' \in \mathcal{C}_\rho$ such that $H_1 \subseteq H_1'$, $H_2 \subseteq H_2'$, and $H_2' = (H_1')^c$. Hence, exactly one of H_1' and H_2' contains e, say H_1'. By the choice of e, $\rho(H_1 \cup \{e\}) \cap H_2 \neq \emptyset$. Since ρ is monotone, $\rho(H_1 \cup \{e\}) \subseteq H_1'$ and hence H_1' and H_2' are not disjoint; a contradiction. □

The characterization result formulated in Theorem 5 cannot, however, be used to decide in time polynomial in $|E|$, whether an *intensionally* given closure system $(E, \mathcal{C}_\rho)$ is Kakutani, or not. More precisely, in Theorem 6 below we have a negative result for the following decision problem:

Kakutani Problem: *Given* a closure system $(E, \mathcal{C}_\rho)$, where $\mathcal{C}_\rho$ is given by the corresponding closure operator ρ, *decide* whether $(E, \mathcal{C}_\rho)$ is Kakutani, or not.

5. A similar property was considered by the Japanese mathematician Shizou Kakutani for Euclidean spaces (cf. Kakutani, 1937)

Theorem 6. *Any algorithm solving the Kakutani problem above requires* $\Omega\left(2^{|E|/2}\right)$ *closure operator calls.*

Proof. We can assume w.l.o.g. that $\emptyset \in \mathcal{C}_\rho$, as otherwise there are no two separable subsets of E. For any even number[6] $n \in \mathbb{N}$, consider a set E with $|E| = n$ and the set system

$$\mathcal{C}_\rho = \{X \subseteq E : |X| \leq n/2\} \cup \{E\} \ .$$

We claim that $(E, \mathcal{C}_\rho)$ is a Kakutani closure system. Since $\emptyset, E \in \mathcal{C}_\rho$ and $|C_1 \cap C_2| \leq n/2$ for any $C_1, C_2 \in \mathcal{C}_\rho$, $(E, \mathcal{C}_\rho)$ is closed under intersection and hence, it is a closure system. To see that it is Kakutani, notice that all $X \in \mathcal{C}_\rho$ with $|X| = n/2$ are half-spaces; all other closed sets $Y \in \mathcal{C}_\rho$ with $0 < |Y| < n/2$ are not half-spaces. Thus, for any non-empty $A, B \subseteq E$ with $\rho(A) \cap \rho(B) = \emptyset$, $\rho(A)$ can be extended to a half-space $H_1 \in \mathcal{C}_\rho$ such that $H_1 \cap \rho(B) = \emptyset$. By construction, H_1 and its complement H_1^c form a half-space separation of A and B. Hence, $(E, \mathcal{C}_\rho)$ is Kakutani. Note also that for any $C \in \mathcal{C}_\rho$ with $|C| = n/2$, $(E, \mathcal{C}_\rho \setminus \{C\})$ remains a closure system, but becomes non-Kakutani.

We are ready to prove the lower bound claimed. Suppose for contradiction that there exists an algorithm $\mathfrak{A}$ that decides the Kakutani problem with strictly less than $\binom{n}{n/2} = \Omega\left(2^{n/2}\right)$ closure operator calls. Then, for $(E, \mathcal{C}_\rho)$ above, there exists a half-space $C \in \mathcal{C}_\rho$ with $|C| = n/2$ such that $\mathfrak{A}$ has not called ρ for C. But then $\mathfrak{A}$ returns the same answer for the Kakutani and non-Kakutani closure systems $(E, \mathcal{C}_\rho)$ and $(E, \mathcal{C}_\rho \setminus \{C\})$, contradicting its correctness. □

The exponential lower bound in Theorem 6 above holds for *arbitrary* (finite) closure systems. Fortunately, there is a broad class of closure systems that are known to be Kakutani. In particular, as a generic application field of Kakutani closure systems, in Section 5 we focus on Kakutani closure systems over *graphs* and in Section 6 on those over finite *lattices*.

5. Application I: γ-Closure Systems over Graphs

As a first application of Theorem 5, in this section we consider Kakutani closure systems over *graphs*. We start by *generalizing* a fundamental result characterizing this kind of closure systems by means of the *Pasch axiom* (Chepoi, 1994; Ellis, 1952) to *graph structured* set systems. Using the Pasch axiom, the Kakutani problem for closure systems over graphs can be decided with polynomially many calls of the closure operator, in contrast to the exponential lower bound in Theorem 6 for the general case. Since, however, the algorithm we are aware of is practically infeasible even for graphs with some hundred nodes, we turn our attention to establishing structural conditions in terms of *forbidden minors* that imply the Kakutani property (for a good reference on graph minors the reader is referred to, e.g., Diestel, 2012).

More precisely, as a second contribution of this section we show that closure systems over graphs are Kakutani whenever the underlying graph does not contain $K_{2,3}$ as a minor. This result may be of some independent interest as well. We also show that the converse of the claim does not hold, implying that $K_{2,3}$ does *not* provide a forbidden minor characterization of Kakutani closure systems over graphs. Together with the characterization result of

6. A similar proof applies to odd numbers. For simplicity, we omit the discussion of that case.

Chartrand and Harary (1967), our result immediately implies that closure systems over outerplanar graphs and hence, over trees are always Kakutani. Though the latter result is well-known, it is typically derived directly from the Pasch axiom. In contrast, we obtain it as an immediate consequence of our result mentioned above. Using the positive result on trees, in Section 7 we report experimental results obtained by Algorithm 1 on vertex classification for trees. Some of the results are especially remarkable by noting that we used the algorithm in its general form as described in Algorithm 1, i.e., without utilizing any domain specific knowledge (except for the particular closure operator, of course).

Throughout this section, we consider γ-closure systems over graphs. The following fundamental result provides a characterization of this kind of closure systems to be Kakutani.

Theorem 7. *(Chepoi, 1994; Ellis, 1952) For any finite graph $G = (V, E)$, the corresponding γ-closure system $(V, \mathcal{C}_\gamma)$ is Kakutani if and only if γ fulfills the Pasch axiom, i.e.,*

$$x \in \gamma(\{u, v\}) \land y \in \gamma(\{u, w\}) \implies \gamma(\{x, w\}) \cap \gamma(\{y, v\}) \neq \emptyset$$

for all $u, v, w, x, y \in V$.

In Proposition 3 below we generalize Theorem 7 to *graph structured* set systems. More precisely, a *graph structured partitioning* (GSP) is a triple $\mathfrak{G} = (S, G, \mathcal{P})$, where S is a finite set, $G = (V, E)$ is a graph, and $\mathcal{P} = \{\text{bag}(v) \subseteq S : v \in V\}$ is a partitioning of S into $|V|$ non-empty subsets (i.e., $\text{bag}(v) \neq \emptyset$, $\bigcup_{v \in V} \text{bag}(v) = S$, and $\text{bag}(u) \cap \text{bag}(v) = \emptyset$ for all $u, v \in V$ with $u \neq v$). The set $\text{bag}(v)$ associated with $v \in V$ is referred to as the *bag* of v. For a GSP $\mathfrak{G} = (S, G, \mathcal{P})$ with $G = (V, E)$, let $\sigma : 2^S \to 2^S$ be defined by

$$\sigma : S' \mapsto \bigcup_{v \in V'} \text{bag}(v) \tag{3}$$

with

$$V' = \gamma(\{v \in V : \text{bag}(v) \cap S' \neq \emptyset\})$$

for all $S' \subseteq S$, where γ is the closure operator defined in (2).

GSPs arise for example in *graph clustering* (see, e.g., Schaeffer, 2007) and *graph partitioning* (see, e.g., Buluc et al., 2016), which play an important role in many practical applications, such as, for example, community network mining.

Proposition 3. *Let $\mathfrak{G} = (S, G, \mathcal{P})$ be a GSP with $G = (V, E)$. Then σ defined in (3) is a closure operator on S. Furthermore, the corresponding closure system $(S, \mathcal{C}_\sigma)$ is Kakutani if and only if γ corresponding to the closure system $(V, \mathcal{C}_\gamma)$ fulfills the Pasch axiom on G.*

Proof. Since $\mathcal{P}$ is a partitioning of S and γ is a closure operator on V, the extensivity and monotonicity of σ are immediate from those of γ. Furthermore, for all $S' \subseteq S$ we have

$$\gamma(\{v \in V : \text{bag}(v) \cap \sigma(S') \neq \emptyset\}) = \gamma(\{v \in V : \text{bag}(v) \cap S' \neq \emptyset\})$$

by the idempotency of γ. Hence, σ is also idempotent, completing the proof of the first claim.

Regarding the second part, note that the function $\varphi : \mathcal{C}_\gamma \to 2^S$ defined by

$$\varphi : V' \mapsto \bigcup_{v \in V'} \text{bag}(v)$$

for all $V' \in \mathcal{C}_\gamma$ is a bijection between $\mathcal{C}_\gamma$ and $\mathcal{C}_\sigma$, satisfying

$$V_1 \cap V_2 = \emptyset \iff \varphi(V_1) \cap \varphi(V_2) = \emptyset$$

for all $V_1, V_2 \in \mathcal{C}_\gamma$. This immediately implies the second claim. □

Notice that any graph $G = (V, E)$ can be regarded as the (trivial) GSP $\mathfrak{G} = (V, G, \mathcal{P})$, where all blocks in $\mathcal{P}$ are singletons with $\text{bag}(v) = \{v\}$ for all $v \in V$. Hence, Theorem 7 holds only for a special case of the proposition above.

Regarding Theorem 7, note that the Pasch axiom can be turned into a *polynomial* algorithm for deciding the Kakutani problem for closure systems over graphs. Indeed, checking the condition for all $\binom{n}{5}$ quintuples of vertices requires $O\left(n^5\right)$ calls of the closure operator γ, where n is the size of the graph. Since γ can be calculated in time $O\left(n^2\right)$ (Dourado, Gimbel, Kratochvíl, Protti, & Szwarcfiter, 2009), the Kakutani problem can be solved in $O\left(n^7\right)$ time. Though, in contrast to the general case formulated in Theorem 6, this bound on the time complexity is polynomial, the (naive) algorithm sketched above is infeasible even for graphs with some hundreds vertices. Since we are not aware of any *fast* algorithm for deciding the Kakutani problem for closure systems over graphs, below we focus on formulating structural conditions implying the Kakutani property.

Using the characterization result above, in the theorem below we give a sufficient condition for the Kakutani property for closure systems over graphs. To state the theorem, we recall that $K_{2,3}$ denotes the complete bipartite graph (V_1, V_2, E) with $|V_1| = 2$ and $|V_2| = 3$. Furthermore, a graph H is a *minor* of a graph G if H can be obtained from G by a sequence of vertex and edge deletions and edge contractions (see, e.g., Diestel, 2012).

Theorem 8. *For any finite graph $G = (V, E)$, the corresponding γ-closure system $(V, \mathcal{C}_\gamma)$ is Kakutani if G does not contain $K_{2,3}$ as a minor.*

Proof. We prove the claim by contraposition. Let $G = (V, E)$ be a graph such that $(V, \mathcal{C}_\gamma)$ is *not* Kakutani. Then, by Theorem 7, γ does not fulfill the Pasch axiom, i.e.

$$\exists u, v, w \in V \text{ and } x \in \gamma(\{u, v\}), y \in \gamma(\{u, w\}) \text{ s.t. } \gamma(\{v, y\}) \cap \gamma(\{x, w\}) = \emptyset. \quad (4)$$

We claim that u, v, w, x, y are pairwise different. We show this property only for x and w; the proofs for the other vertex pairs are similar. Suppose for contradiction that $x = w$. Then, by (4), x (resp. y) lies on a shortest path between u and v (resp. u and x). But then there is a shortest path between v and y containing x (i.e., $x \in \gamma(\{v, y\})$), contradicting the disjointedness condition in (4).

We are ready to show that (4) implies that G contains $K_{2,3} = (V_1, V_2, E)$ as a minor with $V_1 = \{x, y\}$ and $V_2 = \{u, v, w\}$. By (4), there are shortest paths P_{uv} between u, v with $x \in P_{uv}$ and P_{uw} between u, w with $y \in P_{uw}$. Let u' be the common vertex of P_{ux} and P_{uy} that has the maximum distance to u. It must be the case that $u' \neq x$, $u' \neq y$, and x (resp. y) lies on the subpath $P_{u'v}$ of P_{uv} (resp. $P_{u'w}$ of P_{uw}), as otherwise the disjointness condition in (4) does not hold. Moreover, as $\gamma(\{v, y\}) \cap \gamma(\{x, w\}) = \emptyset$ by (4), the subpaths P_{xv} of $P_{u'v}$ and P_{yw} of $P_{u'w}$ must be vertex disjoint. Hence, G contains the subgraph depicted in Fig. 2a (it suffices to consider only one shortest path between u' and u). Regarding the shortest paths between v and y, note that none of them can contain u', as

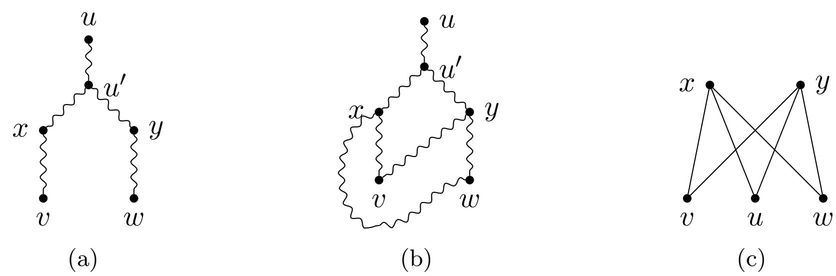


Figure 2: Graph minor of Non-Kakutani graphs

otherwise $x \in \gamma(\{v, y\})$ and thus, $\gamma(\{v, y\}) \cap \gamma(\{x, w\})$ would be non-empty. Furthermore, by (4), they cannot contain x and w. In a similar way, none of the shortest paths between x and w contains u', y, v. Combining these properties with the one implied by (4) that all shortest paths between v and y are pairwise vertex disjoint with all shortest path between w and x, we have that G contains the subgraph given in Fig. 2b. The minor $K_{2,3}$ claimed in the theorem is then obtained from this subgraph by edge contraction (cf. Fig. 2c). □

Remark 1. *The converse of Theorem 8 does not hold, implying that $K_{2,3}$ as a forbidden minor does not characterize the Kakutani property for closure systems over graphs. Indeed, for all complete graphs $K_n = (V, E)$, the corresponding closure system $(V, 2^V)$ is Kakutani. The claim then follows by noting that $K_{2,3}$ is a minor of K_n for all $n \geq 5$.*

In Corollary 2 below we formulate an immediate implication of Theorem 8. We recall that a graph is *outerplanar* if it can be embedded in the plane such that there are no two edges crossing in an interior point and all vertices lie on the outer face.

Corollary 2. *For any outerplanar graph $G = (V, E)$, the corresponding γ-closure system $(V, \mathcal{C}_\gamma)$ is Kakutani.*

Proof. It follows directly from Theorem 8 together with the characterization result of outerplanar graphs by Chartrand and Harary (1967). □

The corollary above applies also to trees, as they are (special) outerplanar graphs. This allows, among others, for a direct application of Theorem 8 to vertex classification in trees; we report some experimental results on this problem in Section 7.

6. Application II: λ-Closure Systems over Lattices

Our second application field is concerned with closure systems over *lattices*. The focus lies, as before, on the HSS and MCSS problems for the special case that the underlying ground set is some finite lattice and the closure operator for a subset S of the ground set is defined by the set of all elements lying between the infimum and supremum of S. For this kind of closure systems we give another greedy algorithm obtained by adapting Algorithm 1 to lattices. Assuming that the closures of the input sets A and B to be separated are disjoint, the greedy algorithm extends them into a disjoint *maximal ideal* I and a *maximal filter* F such that $A \subseteq I$ and $B \subseteq F$ or vice versa. This specialized version has some important advantages over the generic Algorithm 1. In particular, for certain problem classes it reduces the number of closure operator calls *logarithmically*. This is the situation e.g. in frequent closed itemset mining (Pasquier, Bastide, Taouil, & Lakhal, 1999) or formal concept analysis (Ganter et al., 2005). Furthermore, the disjointness of the closures of any two sets can be decided by comparing their suprema and infima. A further important property of the greedy algorithm specialized to lattices is that it regards the input sets A and B above *symmetrically*. This is a crucial difference e.g. to *inductive logic programming* (Muggleton, 1991; Nienhuys-Cheng & Wolf, 1997; Plotkin, 1970), where one is typically interested in finding the smallest ideal of the *subsumption lattices* that contains the set of positive examples. If this smallest ideal, with supremum defined by the least general generalization of the set of positive examples, is *not* disjoint with the set of negative examples, then the separation problem has no solution. This case, however, does not exclude the situation that there is a (maximal) filter containing the set of positive examples that is disjoint with a (maximal) ideal containing the set of negative examples. In addition to these properties, we also show that our modified greedy algorithm comprises an *algorithmic* characterization of Kakutani closure systems over lattices. This provides an alternative point of view to the algebraic characterization in terms of *distributivity* known so far for this kind of closure systems (see, e.g., Kubiś, 2002).

6.1 Notions and Notation

We start by recalling some basic notions from *lattice theory* (see, e.g., Davey & Priestley, 2002; Grätzer, 2011). Let $(S; \leq)$ be a partially ordered set (or poset) and $X \subseteq S$. An element of S is the *supremum* of X, denoted $\sup X$, if $\sup X \geq x$ for all $x \in X$ and $y \geq \sup X$ for all $y \in S$ with $y \geq x$ for all $x \in X$. Similarly, an element of S is the *infimum* of X, denoted $\inf X$, whenever $\inf X \leq x$ for all $x \in X$ and $y \leq \inf X$ for all $y \in S$ with $y \leq x$ for all $x \in X$. A poset $(L; \leq)$ is a *lattice* if $\sup\{a, b\}$ and $\inf\{a, b\}$ exist for all $a, b \in L$. It follows from the definitions that if $(L; \leq)$ is a lattice then $\sup L'$ and $\inf L'$ exist for all finite subsets $L' \subseteq L$. Thus, if L is finite then it is *bounded*, i.e., has a bottom and top element $\bot_L = \inf L$ and $\top_L = \sup L$, respectively. Unless otherwise stated, throughout this section by lattices we always mean *finite* lattices.

Let $(L; \leq)$ be a lattice. For an element $a \in L$, $x \in L$ is an *upper* (resp. *lower*) *cover* of a if $a \leq x$ (resp. $x \leq a$) and for all $y \in L$ with $a \leq y$ (resp. $a \geq y$) it holds that $a \leq x \leq y$ (resp. $y \leq x \leq a$). The set of upper (resp. lower) covers of a is denoted by $\mathrm{C}_\uparrow(a)$ (resp. $\mathrm{C}_\downarrow(a)$). A lattice L is called *distributive* if $\inf\{a, \sup\{b, c\}\} = \sup\{\inf\{a, b\}, \inf\{a, c\}\}$ holds for all $a, b, c \in L$.

A *sublattice* of a lattice L is a non-empty subset of L which is a lattice. An *ideal* I of L is a non-empty subset of L satisfying $\sup\{a, b\} \in I$ for all $a, b \in I$ and $a \in I$ whenever $a \in L$, $b \in I$, and $a \leq b$. A proper ideal I (i.e., which satisfies $I \subsetneq L$) is *prime* if $a \in I$ or $b \in I$ whenever $\inf\{a, b\} \in I$. The dual notions of ideals and prime ideals are called *filters* and *prime filters*, respectively. One can easily check that all ideals and filters of L are sublattices of L. Furthermore, as $|L| < \infty$ by assumption, an ideal I (resp. filter F) can be represented by $\sup I$ (resp. $\inf F$). The ideal (resp. filter) of L with top (resp. bottom) element a is denoted by $(a]$ (resp. $[a)$). It follows from the definitions that the complement of a prime ideal of L is a prime filter of L and vice versa.

Closure Systems over Lattices For finite lattices $(L;\leq)$, we will consider the usual *closure* operator (see, e.g., van de Vel, 1984), i.e., the function $\lambda : 2^L \to 2^L$ defined by

$$\lambda : L' \mapsto \{x \in L \mid \inf L' \leq x \leq \sup L'\} \tag{5}$$

for all $L' \subseteq L$. The set $\lambda(L')$ forms a *closed* sublattice of L, where a sublattice S of L is *closed* if for all $a, b \in S$ and for all $c \in L$, $c \in S$ whenever $a \leq c \leq b$. Thus, $(L, \mathcal{C}_\lambda)$ is a *closure system* formed by the family of closed sublattices of L together with the empty set. This type of closure systems will be referred to as λ-*closure systems.* In Lemmas 2 and 3 below we formulate some basic properties of finite lattices and λ-closure systems. Though most of the claims follow from basic properties of lattices, we provide all proofs for the reader's convenience.

Lemma 2. *Let $(L;\leq)$ be a finite lattice and $A, B \subseteq L$. Then the following statements are equivalent:*

(i) $\inf B \not\leq \sup A$*,*

(ii) $[\inf B) \cap (\sup A] = \emptyset$*,*

(iii) there exist an ideal $I \subseteq L$ and a filter $F \subseteq L$ with $I \cap F = \emptyset$ such that $A \subseteq I \wedge B \subseteq F$.

Proof. For (i) $\Longrightarrow$ (ii), suppose for contradiction that $[\inf B) \cap (\sup A] \neq \emptyset$. Then there is an $x \in L$ with $\inf B \leq x$ and $x \leq \sup A$, contradicting (i). The proof of (ii) $\Longrightarrow$ (iii) follows directly from the fact that $[\inf B)$ is a filter and $(\sup A]$ an ideal. Regarding (iii) $\Longrightarrow$ (i), it must be the case that $\inf B \not\leq \sup A$, as otherwise $\inf F \leq \inf B \leq \sup A \leq \sup I$, contradicting the disjointness of I and F. □

Lemma 3. *Let $(L, \mathcal{C}_\lambda)$ be the λ-closure system over a finite lattice $(L;\leq)$ and $A, B \subseteq L$. Then $\lambda(A) \cap \lambda(B) = \emptyset$ if and only if there exist an ideal $I \subseteq L$ and a filter $F \subseteq L$ with $I \cap F = \emptyset$ such that $(A \subseteq I \wedge B \subseteq F)$ or $(B \subseteq I \wedge A \subseteq F)$.*

Proof. The proof of the "*if*" direction is immediate by $I, F \in \mathcal{C}_\lambda$. Regarding the other direction, we first claim that $\lambda(A) \cap \lambda(B) = \emptyset$ implies $\inf B \not\leq \sup A$ or $\inf A \not\leq \sup B$. Suppose for contradiction that $\inf B \leq \sup A$ and $\inf A \leq \sup B$. Then $\inf B \leq \sup\{\inf A, \inf B\} \leq \sup B$ and $\inf A \leq \sup\{\inf A, \inf B\} \leq \sup A$, implying $\sup\{\inf A, \inf B\} \in \lambda(A) \cap \lambda(B)$, which contradicts $\lambda(A) \cap \lambda(B) = \emptyset$. The claim then follows from Lemma 2 by the symmetry of A and B. □

6.2 Maximal Closed Set Separation in Lattices

Applying Theorem 4 to λ-closure systems over a lattice $(L;\leq)$, we have that Algorithm 1 requires $O(|L|)$ closure operator calls. If the cardinality of L is exponential in some parameter n, then the bound above becomes exponential in n. As an example, in case of *formal concept analysis* (Ganter et al., 2005), the cardinality of the concept lattice can be exponential in that of the underlying sets of objects and attributes. As another example, the lattice formed by the family of *closed (item)sets* of a transaction database over n items can also be exponential in n (Boros, Gurvich, Khachiyan, & Makino, 2003). These and other examples motivate us to adapt Algorithm 1 to lattices in a natural way, allowing for

Algorithm 2: MAXIMAL CLOSED SET SEPARATION IN LATTICES

Input: lattice $(L; \leq)$ with $|L| < \infty$ given by an upward and a downward refinement operator returning $\mathrm{C}_{\uparrow}(a)$ and $\mathrm{C}_{\downarrow}(a)$ for any $a \in L$, and $A, B \subseteq L$

Output: supremum of a maximal ideal $I \in \mathcal{C}_\lambda$ and infimum of a maximal filter $F \in \mathcal{C}_\lambda$ separating A and B in $(L, \mathcal{C}_\lambda)$ with λ defined in (5) if $\lambda(A) \cap \lambda(B) = \emptyset$; "NO" o/w

1 **if** $(\sup A \not\geq \inf B)$ **then** $\top_I \leftarrow \sup A,\ \bot_F \leftarrow \inf B$
2 **else if** $(\sup B \not\geq \inf A)$ **then** $\top_I \leftarrow \sup B,\ \bot_F \leftarrow \inf A$
3 **else return** *"NO"*
4 **while** $\exists u \in \mathrm{C}_{\uparrow}(\top_I)$ *with* $u \not\geq \bot_F$ **do** $\top_I \leftarrow u$
5 **while** $\exists l \in \mathrm{C}_{\downarrow}(\bot_F)$ *with* $l \not\leq \top_I$ **do** $\bot_F \leftarrow l$
6 **return** $\top_I, \bot_F$

an upper bound on the number of closure operator calls in terms of the cardinalities of the *upper* and *lower covers* of a lattice and the maximum chain length in L. As we show below, in case of concept lattices or (frequent) closed itemset lattices, the exponential bound above reduces to $O\left(n^2\right)$.

The algorithm solving the MCSS-problem for finite lattices is given in Algorithm 2. It assumes that the input lattice $(L; \leq)$ is given by an upward $\mathrm{C}_{\uparrow}$ and a downward $\mathrm{C}_{\downarrow}$ refinement operator returning the sets of upper resp. lower covers for the elements of L. For any $A, B \subseteq L$, the algorithm first checks whether their closures are disjoint or not; this is decided by comparing the suprema and infima of A and B (cf. Lines 1–3). If the two closed sets are disjoint then, by Lemma 3, L has a smallest ideal I and a smallest filter F such that I and F are disjoint and either $\lambda(A) \subseteq I$ and $\lambda(B) \subseteq F$ or vice versa. The algorithm then iteratively tries to extend either I into a larger ideal or F into a larger filter in such a way that the extension does not violate the disjointness condition. In the first case, the supremum of I is replaced by one of its upper covers; in the second one the infimum of F by one of its lower covers. Finally, the algorithm stops when the current ideal and hence, the current filter becomes prime or when any further extension makes them non-disjoint.

Algorithm 2 has some important advantageous properties over Algorithm 1. In particular, note that while Algorithm 1 considers *all* uncovered elements for the extension of the current closed sets, Algorithm 2 restricts the choice of the next generator element to $\mathrm{C}_{\uparrow}(\sup I) \cup \mathrm{C}_{\downarrow}(\inf F)$, i.e., to a *subset* of the set of elements uncovered so far. Although in the worst case this change does not improve the number of closure operator calls stated in Theorem 4 for Algorithm 1, below we show that a logarithmic bound holds for certain closure systems over lattices. Another advantageous property of Algorithm 2 is that it utilizes that the disjointness of two closed sets can be decided by comparing two elements only, i.e., the supremum of the current ideal with the infimum of the current filter. Furthermore, the closure operator can be calculated in an easy way by taking advantage of the fact that any closed sublattice of L can be represented by its top and bottom elements. We have the following result for Algorithm 2:

Theorem 9. *For any λ-closure system over a finite lattice $(L; \leq)$, Algorithm 2 solves the MCSS problem correctly.*

Proof. Let $(L, \mathcal{C}_\lambda)$ be the λ-closure system over a lattice $(L; \leq)$ and $A, B \subseteq L$. The correctness for the case that $\lambda(A) \cap \lambda(B) \neq \emptyset$ (or equivalently, A and B are not separable in $\mathcal{C}_\lambda$) is immediate from Lemmas 2 and 3. Applying Lemma 3 to the case that $\lambda(A) \cap \lambda(B) = \emptyset$, there exist disjoint ideal I and filter F in $\mathcal{C}_\lambda$ such that $A \subseteq I$ and $B \subseteq F$ or vice versa. For the symmetry of A, B we can assume without loss of generality that $A \subseteq I$ and $B \subseteq F$. Then, by Lemma 2, the condition in Line 1 holds and thus, the algorithm terminates in Line 6 for this case. Consider the sequences $u_1, \ldots, u_p \in L$ and $l_1, \ldots, l_q \in L$ selected in this order in Lines 4 and 5, respectively. By construction, $A \subseteq (u_0] \subsetneq (u_1] \subsetneq \ldots \subsetneq (u_p]$ and $B \subseteq [l_0) \subsetneq [l_1) \subsetneq \ldots \subsetneq [l_q)$, where $u_0 = \sup A$ and $l_0 = \inf B$. Furthermore, as $u_p \ngeq l_q$ (cf. Line 5), the ideal $(u_p]$ and filter $[l_q)$ corresponding to the output $\top_I = u_p$ and $\bot_F = l_q$ are disjoint by Lemma 2. Thus, they form a closed set separation of A and B in $\mathcal{C}_\lambda$.

We now show that $(u_p]$ and $[l_q)$ form a *maximal* closed set separation of A and B in $\mathcal{C}_\lambda$. Suppose for contradiction that there exist $I', F' \in \mathcal{C}_\lambda$ with $I' \cap F' = \emptyset$, $(u_p] \subseteq I'$, and $[l_q) \subseteq F'$ such that at least one of the two containments is proper. We present the proof for $(u_p] \subsetneq I'$ only; the case of $[l_q) \subsetneq F'$ is similar. Since $(u_p] \subsetneq I'$, there exists an $u \in \mathrm{C}_\uparrow(u_p)$ with $u_p \lneq u \leq \sup I'$. But then, by Line 4 we have $u \geq l_q$, which contradicts $I' \cap F' = \emptyset$ by Lemma 3, as $\sup I' \geq u \geq l_q \geq \inf F'$. □

One can easily check that the number of evaluations of the relation "$\leq$" in Lines 4 and 5 is $O(H_L C_L)$, where H_L is the maximum chain length in L and $C_L = \max_{x \in L} \max\{|\mathrm{C}_\uparrow(x)|, |\mathrm{C}_\downarrow(x)|\}$. Proposition 4 below utilizes this property for the special case that the underlying lattice is a family of subsets of some finite ground set.

Proposition 4. *Let $(L, \mathcal{C}_\lambda)$ be a λ-closure system over a lattice $(L; \subseteq)$ with $L \subseteq 2^E$ for some ground set E of cardinality n. Then Algorithm 2 solves the MCSS problem for the λ-closure system over $(L; \subseteq)$ with $O(n^2)$ evaluations of the subset relation.*

Proof. It follows directly from the remark above by $H_L = O(n)$ and $H_C = O(n)$. □

Since $L \subseteq 2^E$ for some ground set E with $|E| = n$, $|L|$ can be exponential in n. As an application of Proposition 4 to concept lattices and closed (frequent) itemsets, we have that maximal closed separations can be found in time polynomial in the size of the underlying ground sets for these types of closure systems.

We now present two illustrative examples of the application of Algorithm 2 to lattices.

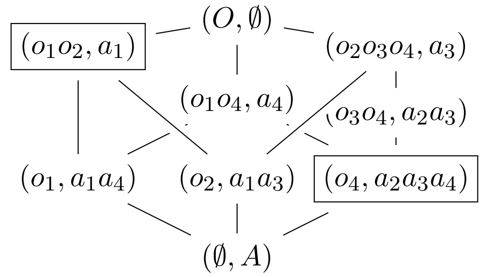
Example 4. *Our first example is concerned with concept lattices (Ganter et al., 2005). Given a binary matrix M over a set O of rows (objects) and a set A of columns (attributes), a formal concept $C = (O', A')$ is a maximal submatrix of M such that all entries in C are equal to 1. It is a well-known fact (see, e.g., Ganter et al., 2005) that the set of all concepts of M together with $(\emptyset, A)$ and $(O, \emptyset)$ form a lattice $(L; \leq)$ with the partial order "$\leq$" defined by*

$$(O_1, A_1) \leq (O_2, A_2) \iff O_1 \subseteq O_2 \ .$$

Consider the concept lattice in Fig. 3b defined by the $O \times A$ binary matrix M in Fig. 3a, where $O = \{o_1, \ldots, o_4\}$ is the set of objects representing o_1 = 'equilateral triangle', o_2 = 'right triangle', o_3 = 'rectangle', o_4 = 'square', and $A = \{a_1, \ldots, a_4\}$ is the set of attributes

	a_1	a_2	a_3	a_4
o_1	1	0	0	1
o_2	1	0	1	0
o_3	0	1	1	0
o_4	0	1	1	1

(a) Binary matrix M over $O \times A$.



(b) Concept lattice defined by M.

Figure 3: Example of a formal context (LHS) and its corresponding concept lattice (RHS).

corresponding to $a_1 =$ *'has 3 vertices',* $a_2 =$ *'has 4 vertices',* $a_3 =$ *'has a right angle', and* $a_4 =$ *'is equiliteral'.*[7]

Suppose we want to separate the singleton concept sets $A = \{(o_4, a_2a_3a_4)\}$ *and* $B = \{(o_1o_2, a_1)\}$*, i.e., the set consisting of the concept 'square' from that containing the concept of 'triangle'. As* $\sup A = (o_4, a_2a_3a_4) \not\geq (o_1o_2, a_1) = \inf B$*, Algorithm 2 first extends the smallest ideal* $I = \{(\emptyset, A), (o_4, a_2a_3a_4)\}$ *containing* A *into a maximal ideal* I' *such that* $\sup I' \not\geq (o_1o_2, a_1)$*. Since both elements of* $\mathrm{C}_{\uparrow}((o_4, a_2a_3a_4)) = \{(o_1o_4, a_4), (o_3o_4, a_2a_3)\}$ *satisfies this condition, in Line 4 one of them is selected arbitrarily, say* (o_1o_4, a_4)*. For the new ideal* I *with* $\sup I = (o_1o_4, a_4)$ *we have that it cannot be extended, as for the only element* $(O, \emptyset)$ *in the upper covers of* (o_1o_4, a_4) *we have* $(O, \emptyset) > (o_1o_4, a_4)$*. The algorithm therefore continuous in Line 5 by extending the smallest filter* $F = \{(o_1o_2, a_1), (O, \emptyset)\}$ *containing* B *into a maximal filter* F' *such that* $\inf F' \not\leq \sup I = (o_1o_4, a_4)$*. The only element it can select from* $\mathrm{C}_{\downarrow}((o_1o_2, a_1)) = \{(o_1, a_1a_4), (o_2, a_1a_3)\}$ *is* (o_2, a_1a_3)*, as* $(o_1, a_1a_4) < (o_1o_4, a_4)$*. For the new filter* F *with* $\inf F = (o_2, a_1a_3)$ *we have that it cannot be extended, as for the only element* $(\emptyset, A)$ *in the lower covers of* (o_2, a_1a_3) *we have* $(\emptyset, A) < (o_1o_4, a_4)$*. The algorithm therefore terminates with the supremum* (o_1o_4, a_4) *and infimum* (o_2, a_1a_3) *of the maximal separating closed sets*

$$I = \{(\emptyset, A), (o_1, a_1a_4), (o_4, a_2a_3a_4), (o_1o_4, a_4)\}$$

and

$$F = \{(o_2, a_1a_3), (o_1o_2, a_1), (o_2o_3o_4, a_3), (O, \emptyset)\} \ ,$$

respectively. That is, 'square' is separated from 'triangle' by the set of concepts specializing 'equilateral objects' and that generalizing 'right triangles'. Note that I *and* F *do not form a half-space separation because* $(o_3o_4, a_2a_3) \notin I \cup F$*.*

Our second example is concerned with finding consistent hypotheses in *inductive logic programming*. For simplicity, the example below is restricted to a very simple first-order

7. See (Ignatov, 2014) for more details on this example.

vocabulary by noting that the same idea holds for any finite sublattice of the *subsumption lattice* (cf. Nienhuys-Cheng & Wolf, 1997) for the definition and some formal properties of the subsumption lattice). More precisely, in the example below we assume that the vocabulary consists of a single predicate symbol P of arity n for some $n \in \mathbb{N}$ and a set V of variables. An atom $P(t_1, \ldots, t_n)$ with $t_1, \ldots, t_n \in V$ *generalizes* $P(t'_1, \ldots, t'_n)$, denoted $P(t_1, \ldots, t_n) \geq P(t'_1, \ldots, t'_n)$, if there exists a function $\sigma : V \to V$ such that $P(\sigma(t_1), \ldots, \sigma(t_n)) = P(t'_1, \ldots, t'_n)$. Two atoms A_1, A_2 are *equivalent* if $A_1 \leq A_2$ and $A_2 \leq A_1$. Let L be a maximal set of P-atoms, each of the above form, that contains no two equivalent atoms. Clearly, each element of L can be represented by any atoms from its equivalence class. It holds that $(L; \leq)$ is a finite lattice. Furthermore, the bottom (resp. top) element of L is a P-atom such that all variables in it are pairwise different (resp. are the same). Using the above notions and notation, we are ready to formulate our other example.

Example 5. *Consider the lattice $(L; \leq)$ defined above for P with $n = 5$ and $V = \{v, w, x, y, z\}$. We can assume without loss of generality that $\bot_L = P(z, z, z, z)$ and $\top_L = P(v, w, x, y, z)$. Let $A = \{P(w, x, y, y, z), P(w, x, y, z, z)\}$ and $B = \{P(y, y, y, y, z), P(y, y, y, z, y)\}$ denote the sets of positive and negative examples, respectively. In the most common problem setting in inductive logic programming (Nienhuys-Cheng & Wolf, 1997), one is interested in finding a P-atom $g \in L$ such that g generalizes all elements of A and none of the elements in B, if such a g exists. Clearly, such a g exists if and only if $\sup A = P(v, w, x, y, z)$ does not generalize any of the P-atoms in B. Since this is not the case for our example, the consistent hypothesis finding problem has no solution.*

If, however, we only require A and B to be separable in $(L, \mathcal{C}_\lambda)$, then Algorithm 2 returns a solution. Indeed, while $\sup A = P(v, w, x, y, z) \geq P(z, z, z, z, z) = \inf B$, for $\sup B$ and $\inf A$ we have $\sup B = P(x, x, y, x, z) \not\geq P(x, y, z, z, z) = \inf A$ (cf. Lines 1 and 2 of Algorithm 2). Thus, Algorithm 2 returns $\top_I = P(w, w, x, y, z)$ and $\bot_F = P(y, z, z, z, z)$. For the corresponding ideal $(P(w, w, x, y, z)]$ and filter $[P(y, z, z, z, z))$ we have that they are disjoint, contain B and A, respectively, and are maximal in $(L, \mathcal{C}_\lambda)$ with respect to these properties. In other words, the output of algorithm separates A and B by the sets of P-atoms that are generalizations of $P(y, z, z, z, z)$ and are generalized by $P(w, w, x, y, z)$, respectively. This example also shows that our approach is able to produce an output for such cases where the traditional approaches based on Plotkin's least general generalization (Plotkin, 1970) have no solution. The reason is that Algorithm 2 treats the input two sets symmetrically, in contrast to all such approaches.

6.3 Kakutani Closure Systems over Lattices

In this section we consider *Kakutani* λ-closure systems over finite lattices. This kind of closure systems have a well-known *algebraic* characterization in terms of distributivity (see, e.g., Kubiś, 2002; van de Vel, 1984). As an orthogonal result, in Theorem 10 below we show that Algorithm 2 provides an *algorithmic* characterization of Kakutani λ-closure systems over lattices.

Theorem 10. *Let $(L, \mathcal{C}_\lambda)$ be the λ-closure system over a finite lattice $(L; \leq)$. Then $(L, \mathcal{C}_\lambda)$ is Kakutani if and only if for all non-empty $A, B \subseteq L$ with $\lambda(A) \cap \lambda(B) = \emptyset$, the ideal $(\top_I]$ and filter $[\bot_F)$ defined by the output $\top_I$ and $\bot_F$ of Algorithm 2 form a partitioning of L.*

Proof. The sufficiency is immediate by Theorem 9. For the necessity, let $(L, \mathcal{C}_\lambda)$ be a Kakutani closure system and $A, B \subseteq L$ with $\lambda(A) \cap \lambda(B) = \emptyset$. Let $u_1, \ldots, u_p$ and $l_1, \ldots, l_q$ be the maximal sequences considered in the proof of Theorem 9 for the case of $\lambda(A) \cap \lambda(B) = \emptyset$. For their last elements we have that $(u_p], [l_q) \in \mathcal{C}_\lambda$ and $(u_p] \cap [l_q) = \emptyset$. Since $\mathcal{C}_\lambda$ is Kakutani, there is a proper partitioning $H, H^c \in \mathcal{C}_\lambda$ of L such that $(u_p] \subseteq H$ and $[l_q) \subseteq H^c$. Thus, $\bot_L \in H$ and $\top_L \in H^c$, implying that H is a prime ideal and H^c is its complement prime filter. Suppose for contradiction that one of the two containments above, say the first one, is proper (i.e., $(u_p] \subsetneq H$). But then, using similar arguments as in the proof of Theorem 9, there exists an element $u \in \mathrm{C}_\uparrow(u_p)$ such that u will be selected after u_q in Line 4. This contradicts that u_q is the last element selected in Line 4. □

Corollary 3. *Let $(L, \mathcal{C}_\lambda)$ be the λ-closure system over a finite lattice $(L; \leq)$. Then $(L, \mathcal{C}_\lambda)$ is distributive if and only if for all non-empty $A, B \subseteq L$ with $\lambda(A) \cap \lambda(B) = \emptyset$, the ideal $(\top_I]$ and filter $[\bot_F)$ defined by the output $\top_I$ and $\bot_F$ of Algorithm 2 form a partitioning of L.*

Proof. Immediate from Theorem 10 and the characterization of λ-closure systems over finite lattices in terms of distributivity (Kubiś, 2002; van de Vel, 1984). □

7. Some Illustrative Experimental Results

In this section we empirically demonstrate the potential of Algorithm 1 on binary classification problems over Kakutani and non-Kakutani closure systems. For both types of closure systems we consider the case that the input sets are half-space separable in the underlying closure system $(E, \mathcal{C})$. That is, the ground set E is partitioned into two blocks E_1 and E_2 corresponding to the two labels. To measure the predictive performance, for the closed sets $H_1, H_2 \subseteq E$ returned by Algorithm 1 for the input $A \subseteq E_1$ and $B \subseteq E_2$, respectively, we consider the accuracy and coverage defined by

$$\text{Accuracy} = \frac{|E_1 \cap H_1'| + |E_2 \cap H_2'|}{|H_1' \cup H_2'|} \tag{6}$$

and

$$\text{Coverage} = \frac{|H_1' \cup H_2'|}{|E'|} \quad , \tag{7}$$

where $E' = E \setminus (A \cup B)$, $H_1' = E' \cap H_1$, and $H_2' = E' \cap H_2$. Note that for Kakutani closure systems, Algorithm 1 always returns a half-space separation of A and B by Theorem 5. That is, the coverage of the output is always 1. Accordingly, for this case we measure the accuracy only.

We stress that our main goal with these experiments is to demonstrate that a remarkable predictive performance can be achieved already with the very general greedy strategy used by Algorithm 1. Since we do not utilize any domain specific knowledge in these experiments (e.g., for the selection of non-redundant training examples[8]), we do not compare our generic approach to any of the state-of-the-art algorithms designed for some specific problem.

8. In case of γ-closure systems over trees, such a non-redundant set could be obtained by considering only leaves as training examples.

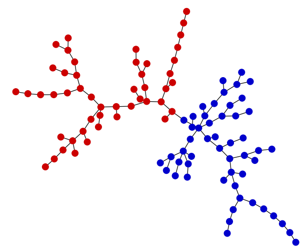


Figure 4: A synthetic tree of size 100 together a random partitioning of its vertex set.

Vertex Classification in Trees Our first experiments are concerned with binary vertex classification in synthetic trees using γ-closure systems (cf. Section 3). More precisely, for every synthetic tree $G = (V, E)$ and corresponding γ-closure system $(V, \mathcal{C}_\gamma)$, we generated a random partitioning $L_r, L_b \in \mathcal{C}_\gamma$ of V (see, Fig. 4 for an example). The half-spaces L_r and L_b will be referred to as *red* and *blue* vertices, respectively. We then generated two random subsets $R \subseteq L_r, B \subseteq L_b$, and considered the following supervised learning task: *Given* a training set $D = R \cup B$ with $R \subseteq L_r, B \subseteq L_b$ for some *unknown* partitioning $L_r, L_b \in \mathcal{C}_\gamma$ of V and a vertex $v \in V$, *predict* whether $v \in L_r$ or $v \in L_b$. Since L_r and L_b are half-spaces, Algorithm 1 always returns some half-spaces $H_r, H_b \in \mathcal{C}_\gamma$ with $R \subseteq H_r$ and $B \subseteq H_b$ by Theorem 5 because $(V, \mathcal{C}_\gamma)$ is Kakutani by Corollary 2. The class of v is then predicted by *blue* if $v \in H_b$; o/w by *red*. Note that H_r and H_b can be different from L_r and L_b, respectively.

For the empirical evaluation of the predictive performance of Algorithm 1 above we used the following synthetic dataset:

D1 For all $s = 100, 200, \ldots, 1000, 2000, \ldots, 5000$, we generated 50 random trees of size s (see the x-axes in Fig. 5) and partitioned the vertex set of all trees $G = (V, E)$ into two random half-spaces L_r, L_b in $(V, \mathcal{C}_\gamma)$ such that $\frac{1}{3} \leq \frac{|L_r|}{|L_b|} \leq 3$.

For all trees in D1 we generated 20 random training sets, each of different cardinalities (see the y-axes in Fig. 5). In this way we obtained 1000 learning tasks (50 trees $\times$ 20 random training sets) for each tree size (x-axes) and training set cardinality (y-axes). The results are presented in Fig. 5. For each tree size (x-axes) and training set cardinality (y-axes) we plot the average of the accuracies (see Eq. (6)) obtained for the 1000 learning settings considered. Since the underlying closure systems are always Kakutani, the coverage is not reported.

Regarding D1 (Fig. 5), one can observe that a remarkable average accuracy over 80% can be obtained already for 40 training examples even for trees of size 1000. This corresponds to a relative size of 2.5% (see the LHS of Fig. 5). With increasing tree size, the relative size of the training set reduces to 2%, as we obtain a similar average accuracy already for 100 training examples for trees of size 5000 (see the RHS of Fig. 5). The explanation of these surprisingly considerable results raises some interesting theoretical questions for probabilistic combinatorics, as the output half-spaces can be *inconsistent* with the partitioning formed by L_r, L_b.

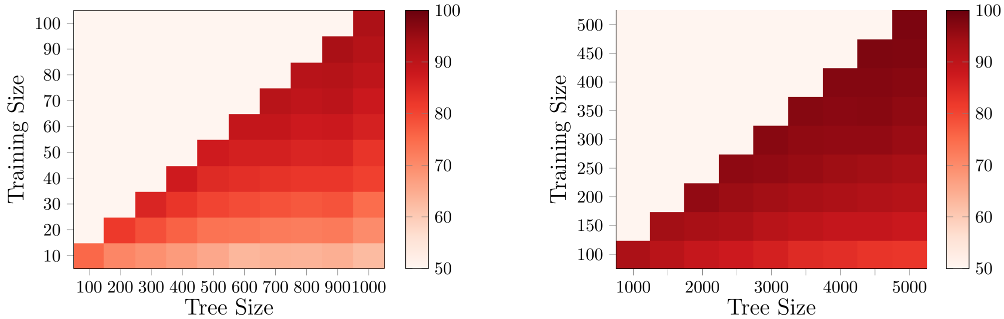


Figure 5: Accuracy of vertex classifications where labels are half-spaces (cf. dataset D1).

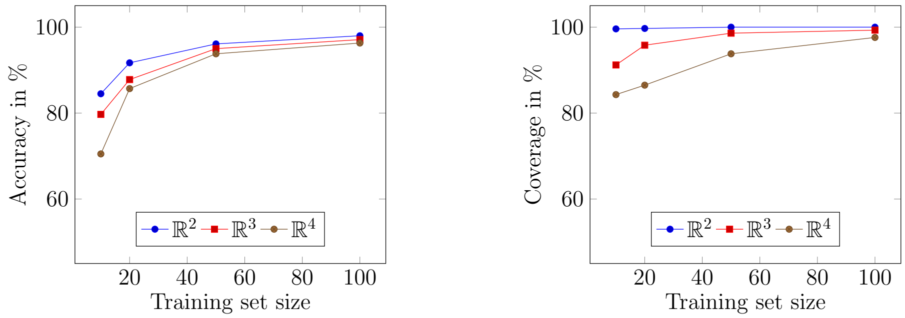
Point Classification in $\mathbb{R}^d$ In our second experiment we measure the predictive performance of Algorithm 1 for classification tasks in α-closure systems over finite subsets of $\mathbb{R}^d$ (cf. Section 2). For these experiments we considered several artificial datasets generated as follows:

D2 For every $d = 2, 3, 4$, we generated 100 sets by taking the union of two random sets $P, N \subset \mathbb{R}^d$ satisfying $\text{conv}(P) \cap \text{conv}(N) = \emptyset$ and $|P| = |N| = 500$ (i.e., each set consists of $1,000$ points).

For each set $E = P \cup N$ in D2 we considered four learning tasks obtained by selecting some random sets $A \subseteq P$ and $B \subseteq N$ for $|P \cup N| = 10, 20, 50, 100$, respectively. Algorithm 1 was run on these input sets A and B, together with the closure operator α defined in (1). The prediction was made by the two maximal disjoint closed sets returned by Algorithm 1; the points not selected as training examples (i.e., $E \setminus (A \cup B)$) were used for the test. To evaluate our approach, we calculated the accuracy (Eq. (6)) and coverage (Eq. (7)) for each of the 4×100 problem settings, for every $d = 2, 3, 4$.

Notice that for E, P, N above, $\text{conv}(P) \cap \text{conv}(N) = \emptyset$ implies $P, N \in \mathcal{C}_\alpha$, i.e., A and B are half-space separable in $(E, \mathcal{C}_\alpha)$. Still, the output of Algorithm 1 is not necessarily a half-space separation of A and B, as $(E, \mathcal{C}_\alpha)$ is not Kakutani in general. To see this, consider the example in Fig. 1 with initial sets $A = \{x\}$ and $B = \{u, v\}$. Clearly, $P = \{x, y, z\}$ and $N = \{u, v, w\}$ form a half-space separation of A and B. However, for the case that Algorithm 1 considers y, w, z in this order in Line 7, we have $H_1 = \{x, y, w\}$ and $H_2 = \{u, v\}$ after y and w have been processed. But then z can be added neither to H_1 nor to H_2 without violating the disjointness condition. The above considerations imply that not only the accuracy, but also the coverage can be lower than 100% in these experiments.

The average accuracy and coverage results are reported in Fig. 6. In particular, Fig. 6a shows that the cardinality of the training set (x-axis) has a significant effect on the *accuracy* (y-axis), ranging from 70% to 98% from 10 (i.e., 1%) to 100 (i.e., 10%) training examples, respectively. Note that for small training sets, the accuracy is very sensitive to the dimension. In particular, the difference is more than 10% for 10 training examples. However, it



(a) accuracy (b) coverage

Figure 6: Average accuracy and coverage results on the D2 datasets for different number of training examples.

vanishes with increasing training set size.[9] Regarding the *coverage* (see Fig. 6b), it was at least 90% in most of the cases, by noting that it shows a similar sensitivity to the size of the training data as the accuracy.

In summary, our experimental results reported in this section clearly demonstrate that surprisingly considerable predictive performance can be obtained with Algorithm 1, even for non-Kakutani closure systems.

8. Concluding Remarks

Throughout this paper we considered *binary* separation problems only. Clearly, they can naturally be extended to *multi-class* separation problems, i.e., in which we are interested in generalizing the half-space separation problem to finding a k-partitioning of the ground set and the maximal closed set separation problem to finding k maximal closed sets that are pairwise disjoint, for some $k \geq 2$ integer. While the generalization of our results concerning maximal closed set separation is straightforward, it is less obvious for the k-partitioning problem. We note that for the special case of graphs, this problem has already been studied by Artigas et al. (2011).

The *experimental* results presented in the previous section show that despite several theoretical difficulties, impressive predictive accuracy can be obtained already by a simple generic greedy algorithm for binary classification problems over abstract closure systems. This is somewhat surprising because the only information about the "nature" of the data has been encoded in the underlying closure operator. That is, in our experiments we *deliberately* have not utilized any domain specific knowledge (and accordingly, not compared our results to any domain specific state-of-the-art algorithm). The specialization of Algorithm 1 to lattices in Section 6 provides an example that a remarkable improvement can be obtained by using some additional background knowledge specific to the particular problem at hand.

9. We have carried out experiments with larger datasets as well; the results on those datasets clearly indicate that the accuracy remains quite stable w.r.t. the size of the point set. For example, for a training set size of 40, it was consistently around 94% for different cardinalities.

It would be interesting to look at further specializations of Algorithm 1 by enriching the data with additional information, such as, for example, domain specific or abstract distances between elements and closed sets. Having such a distance measure, it is one of the most natural questions whether the idea of half-space separation with *maximum margin* (Vapnik, 1998) can be generalized to half-space and maximal closed set separation in *abstract* closure systems.

In the general problem settings considered in this work we assumed that the closure operator is an oracle, which returns the closure of a set *extensionally*. In case of closure systems over lattices, closed sets (e.g., ideals and filters) can, however, be represented *intensionally* (i.e., by their suprema and infima). As another example, for closure systems over trees we have that any half-space has a succinct intensional representation e.g. by a single node together with the edge connecting it to the complement half-space. These and other examples motivate the study of structural properties of closure systems allowing for some compact *intensional* representation of abstract half-spaces and closed sets. A further problem is to study algorithms solving the HSS and MCSS problems for closure systems, for which an upper bound on the VC-dimension is known in advance. The relevance of the VC-dimension in this context is that for any closed set $C \in \mathcal{C}_\rho$ of a closure system $(E, \mathcal{C}_\rho)$, there exists a set $G \subseteq E$ with $|G| \leq d$ such that $\rho(G) = C$, where d is the VC-dimension of $\mathcal{C}_\rho$ (see, e.g., Horváth & Turán, 2001). It is an open question whether the lower bound on the number of closure operator calls can be characterized in terms of the VC-dimension of the underlying closure system. Finally, regarding Kakutani closure systems, it is an interesting research direction to study the relaxed notion of *almost* Kakutani closure systems, i.e., in which the combined size of the output closed sets are close to the cardinality of the ground set.

Acknowledgements

This work was partly supported by the Ministry of Education and Research of Germany (BMBF) under project ML2R (grant number 01/S18038C) and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2070 - 390732324.

References

Artigas, D., Dantas, S., Dourado, M., & Szwarcfiter, J. (2011). Partitioning a graph into convex sets. *Discrete Mathematics*, *311*(17), 1968 – 1977.

Boros, E., Gurvich, V., Khachiyan, L., & Makino, K. (2003). On Maximal Frequent and Minimal Infrequent Sets in Binary Matrices. *Ann. Math. Artif. Intell.*, *39*(3), 211–221.

Boser, B., Guyon, I., & Vapnik, V. (1992). A Training Algorithm for Optimal Margin Classifiers. In *Proc. of the 5th Annual ACM Workshop on Computational Learning Theory*, pp. 144–152. ACM Press.

Buluc, A., Meyerhenke, H., Safro, I., Sanders, P., & Schulz, C. (2016). Recent Advances in Graph Partitioning. In *Algorithm Engineering - Selected Results and Surveys*, Vol. 9220 of *LNCS*, pp. 117–158. Springer.

Chartrand, G., & Harary, F. (1967). Planar Permutation Graphs. *Annales de l'I.H.P. Probabilités et statistiques*, *3*(4), 433–438.

Chepoi, V. (1994). Separation of two convex sets in convexity structures. *J. of Geometry*, *50*(1), 30–51.

Davey, B. A., & Priestley, H. A. (2002). *Introduction to Lattices and Order, Second Edition*. Cambridge University Press.

Diestel, R. (2012). *Graph Theory, 4th Edition*, Vol. 173 of *Graduate texts in mathematics*. Springer.

Dourado, M. C., Gimbel, J. G., Kratochvíl, J., Protti, F., & Szwarcfiter, J. L. (2009). On the computation of the hull number of a graph. *Discrete Mathematics*, *309*(18), 5668–5674.

Ellis, J. W. (1952). A general set-separation theorem. *Duke Math. J.*, *19*(3), 417–421.

Farber, M., & Jamison, R. (1986). Convexity in Graphs and Hypergraphs. *SIAM Journal on Algebraic Discrete Methods*, *7*(3), 433–444.

Ganter, B., Stumme, G., & Wille, R. (2005). *Formal Concept Analysis: Foundations and Applications*. Springer-Verlag, Berlin, Heidelberg.

Grätzer, G. (2011). *Lattice Theory: Foundation*. Springer Basel.

Horváth, T., & Turán, G. (2001). Learning logic programs with structured background knowledge. *Artificial Intelligence*, *128*(1-2), 31–97.

Ignatov, D. I. (2014). Introduction to Formal Concept Analysis and Its Applications in Information Retrieval and Related Fields. In *Information Retrieval - 8th Russian Summer School, RuSSIR 2014, Nizhniy, Novgorod, Russia, August 18-22, 2014, Revised Selected Papers*, pp. 42–141.

Kakutani, S. (1937). Ein Beweis des Satzes von Edelheit über konvexe Mengen. *Proc. Imp. Acad. Tokyo*, *13*, 93–94.

Kubiś, W. (2002). Separation properties of convexity spaces. *J. of Geometry*, *74*(1), 110–119.

Lloyd, J. W. (1987). *Foundations of Logic Programming, Second Edition*. Springer-Verlag.

Muggleton, S. (1991). Inductive logic programming. *New Generation Computing*, *8*(4), 295–318.

Nienhuys-Cheng, S.-H., & Wolf, R. (1997). *Foundations of Inductive Logic Programming*, Vol. 1228 of *LNAI*. Springer Verlag.

Pasquier, N., Bastide, Y., Taouil, R., & Lakhal, L. (1999). Efficient Mining of Association Rules Using Closed Itemset Lattices. *Inf. Syst.*, *24*(1), 25–46.

Plotkin, G. (1970). A Note on Inductive Generalization. *Machine Intelligence*, *5*, 153–163.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*, 386–408.

Schaeffer, S. E. (2007). Survey: Graph Clustering. *Comput. Sci. Rev.*, *1*(1), 27–64.

van de Vel, M. (1984). Binary Convexities and Distributive Lattices. *Proceedings of the London Mathematical Society*, *48*(1), 1–33.

van de Vel, M. (1993). *Theory of Convex Structures, Volume 50 (North-Holland Mathematical Library)*. North Holland.

Vapnik, V. (1998). *Statistical Learning Theory*. John_Wiley.