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

Florian Seiffarth[1]✉, Tamás Horváth[1,2,3], and Stefan Wrobel[1,2,3]

[1]Dept. of Computer Science, University of Bonn, Bonn, Germany
[2]Fraunhofer IAIS, Schloss Birlinghoven, Sankt Augustin, Germany
[3]Fraunhofer Center for Machine Learning, Sankt Augustin, Germany
{seiffarth,horvath,wrobel}@cs.uni-bonn.de

Abstract. Motivated by various binary classification problems in structured data (e.g., graphs or other relational and algebraic structures), we investigate some algorithmic properties of closed set and half-space separation in abstract closure systems. Assuming that the underlying closure system is finite and given by the corresponding closure operator, we formulate some negative and positive complexity results for these two separation problems. In particular, we prove that deciding half-space separability in abstract closure systems is NP-complete in general. On the other hand, for the relaxed problem of maximal closed set separation we propose a simple greedy algorithm and show that it is efficient and has the best possible lower bound on the number of closure operator calls. As a second direction to overcome the negative result above, we consider Kakutani closure systems and show first that our greedy algorithm provides an algorithmic characterization of this kind of set systems. As one of the major potential application fields, we then focus on Kakutani closure systems over graphs and generalize a fundamental characterization result based on the Pasch axiom to graph structure partitioning of finite sets. Though the primary focus of this work is on the generality of the results obtained, we experimentally demonstrate the practical usefulness of our approach on vertex classification in different graph datasets.

Keywords: closure systems · half-space separation · binary classification

1 Introduction

The theory of binary separation in $\mathbb{R}^d$ by hyperplanes goes back to at least Rosenblatt's pioneer work on perceptron learning in the late fifties [12]. Since then several deep results have been published on this topic including, among others, Vapnik and his co-workers seminal paper on support vector machines [2]. The general problem of binary separation in $\mathbb{R}^d$ by hyperplanes can be regarded as follows: Given two finite sets $R, B \subseteq \mathbb{R}^d$, check whether their convex hulls are disjoint, or not. If not then return the answer "No" indicating that R and B are not separable by a hyperplane. Otherwise, there exists a hyperplane in $\mathbb{R}^d$ such that the convex hull of R lies completely in one of the two half-spaces defined by the hyperplane and that of B in the other one. The class of an unseen point in

$\mathbb{R}^d$ is then predicted by that of the training examples in the half-space it belongs to. The correctness of this generic method for $\mathbb{R}^d$ is justified by the result of Kakutani [9] that any two disjoint convex sets in $\mathbb{R}^d$ are always separable by a hyperplane.

While hyperplane separation in $\mathbb{R}^d$ is a well-founded field, the adaptation of the above idea to other types of data, such as graphs and other relational and algebraic structures has received less attention by the machine learning community. In contrast, the idea of abstract half-spaces over finite domains has intensively been studied among others in geometry and theoretical computer science (see, e.g., [4,5,10,15]). Using the fact that the set of all convex hulls in $\mathbb{R}^d$ forms a *closure system*, the underlying idea of generalizing hyperplane separation in $\mathbb{R}^d$ to arbitrary finite sets E is to consider some semantically meaningful closure system $\mathcal{C}$ over E (see, e.g., [16] for abstract closure structures). A subset H of E is then considered as an *abstract* half-space, if H and its complement both belong to $\mathcal{C}$. In this field of research there is a special focus on characterization results of special closure systems, called *Kakutani closure systems* (see, e.g., [4,16]). This kind of closure systems satisfy the following property: If the closures of two sets are disjoint then they are half-space separable in the closure system.

Utilizing the results of other research fields, in this work we deal with the *algorithmic* aspects of half-space separation in closure systems over *finite* domains (or ground sets) from the point of view of *binary classification*. In all results presented in this paper we assume that the abstract closure system is given implicitly via the corresponding *closure operator*. This assumption is justified by the fact that the cardinality of a closure system can be exponential in that of the domain. The closure operator is regarded as an *oracle* (or black box) which returns in *unit time* the closure of any subset of the domain. Using these assumptions, we first show that deciding whether two subsets of the ground set are half-space separable in the underlying abstract closure system is NP-complete.

In order to overcome this negative result, we then relax the problem setting of half-space separation to *maximal* closed set[1] separation. That is, to the problem of finding two closed sets in the closure system that are disjoint, contain the two input subsets, and have no supersets in the closure system w.r.t. these two properties. For this relaxed problem we give a simple 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 way to resolve the negative result mentioned above, we then focus on Kakutani closure systems. We first show that any deterministic algorithm deciding whether a closure system is Kakutani or not requires exponentially many closure operator calls in the worst-case. Despite this negative result, Kakutani closure systems remain highly interesting for our purpose because there are various closure systems which are known to be Kakutani. We also prove that the greedy algorithm mentioned above provides an algorithmic characterization of Kakutani closure systems. This implies that for these systems the output is always a partitioning of the domain into two half-spaces containing the closures of the input sets if and only if their closures are disjoint.

[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.

Regarding potential applications of maximal closed set and half-space separations, we then turn our attention to graphs.[2] Using the notion of convexity for graphs induced by shortest paths [6], we generalize a fundamental characterization result of Kakutani closure systems based on the Pasch axiom [4] to graph structured partitioning of finite sets. Potential practical applications of this generalization result include e.g. graph clustering and partitioning or mining logical formulas over graphs.

Besides the positive and negative theoretical results, we also present extensive experimental results for binary vertex classification in graphs, by stressing that our generic approach is not restricted to graphs. In the experiments we first consider trees and then arbitrary graphs. Regarding *trees*, the closure systems considered are always Kakutani. Our results clearly demonstrate that a remarkable predictive accuracy can be obtained even for such cases where the two sets of vertices corresponding to the two classes do not form half-spaces in the closure systems. Since the closure systems considered over *arbitrary* graphs are not necessarily Kakutani, the case of vertex classification in arbitrary graphs is reduced to that in trees as follows: Consider a set of random spanning trees of the graph at hand and predict the vertex labels by the majority vote of the predictions in the spanning trees. Our experimental results show that this heuristic results in considerable predictive performance on sparse graphs. We emphasize that we deliberately have *not* exploited any domain specific properties in the experiments, as our primary goal was to study the predictive performance of our *general* purpose algorithm. We therefore also have not compared our results with 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. Section 3 is concerned with the negative result on the complexity of the half-space separation problem and with the relaxed problem of maximal closed set separation. Section 4 is devoted to Kakutani and Section 5 to non-Kakutani closure systems. Finally, in Section 6 we conclude and formulate some open problems. Due to space limitations we omit the proofs from this short version.

2 Preliminaries

In this section we collect the necessary notions and notation for set and closure systems (see, e.g., [4,16] for references on closure systems and separation axioms).

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: $\emptyset, E \in \mathcal{C}$ and $X \cap Y \in \mathcal{C}$ holds for all $X, Y \in \mathcal{C}$. The reason of requiring $\emptyset \in \mathcal{C}$ is discussed below. 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 that any closure system can be defined by a *closure operator*,

[2] An entirely different application to binary classification in *distributive lattices* with applications to inductive logic programming and formal concept analysis is discussed in the long version of this paper.

i.e., a function $\rho : 2^E \to 2^E$ satisfying 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 operator ρ over E with $\rho(\emptyset) = \emptyset$ the corresponding closure system, denoted $(E, \mathcal{C}_\rho)$, is defined by its fixed points, i.e., $\mathcal{C}_\rho = \{X \subseteq E : \rho(X) = X\}$. Conversely, for a closure system $(E, \mathcal{C}_\rho)$, the corresponding closure operator ρ is defined by $\rho(X) = \bigcap\{C : X \subseteq C \wedge C \in \mathcal{C}\}$ for all $X \subseteq E$. Depending on the context we sometimes omit the underlying closure operator from the notation and denote the closure system at hand 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.

As an example, for any finite set $E \subset \mathbb{R}^d$, the set system $(E, \mathcal{C})$ with $\mathcal{C} = \{\text{conv}(X) \cap E : X \subseteq E\}$ forms a closure system, where $\text{conv}(X)$ denotes the convex hull of X in $\mathbb{R}^d$. Note that in contrast to convexity in $\mathbb{R}^d$, $\mathcal{C}_\rho$ is not atomic in general, i.e., singletons are not necessarily closed.

We now turn to the generalization of binary separation in $\mathbb{R}^d$ by hyperplanes to that in abstract closure systems by half-spaces (cf. [16] for a detailed introduction into this topic). In the context of machine learning, one of the most relevant questions concerning a closure system $(E, \mathcal{C})$ 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 [4]. 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. Since we are interested in half-space separations, in the definition of closure systems above we require $\emptyset \in \mathcal{C}$, as otherwise there are no half-spaces in $\mathcal{C}$. The following property will be used many times in what follows:

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

Notice that the above generalization does not preserve all natural properties of half-space separability 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[3] are disjoint. In contrast, this property does not hold for finite closure systems in general. To see this, consider the closure system $(\{1, 2, 3\}, \mathcal{C})$ with $\mathcal{C} = \{\emptyset, \{1\}, \{2\}, \{1, 2\}, \{1, 2, 3\}\}$. Note that $\mathcal{C}$ is non-atomic, as $\{3\} \notin \mathcal{C}$. Although $\{1\}$ and $\{2\}$ are both closed and disjoint, they cannot be separated by a half-space in $\mathcal{C}$ because the only half-space containing $\{1\}$ contains also $\{2\}$.

3 Half-Space and Maximal Closed Set Separation

Our goal in this work is to investigate the algorithmic aspects of half-space and closed set separations in abstract closure systems. That is, given two subsets A, B

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

of the ground set, we require the algorithm to return a half-space separation of A and B in $\mathcal{C}$, if such a half-separation exists; o/w the answer "No". As mentioned above, two finite subsets in $\mathbb{R}^d$ can always be separated by a hyperplane if and only if their convex hulls are disjoint. Thus, to decide if two finite subsets of $\mathbb{R}^d$ are separable by a hyperplane, it suffices to check whether their convex hulls are disjoint, or not. As shown above, the situation is different for abstract closure systems because the disjointness of the closures of A and B does not imply their half-space separability in $\mathcal{C}$. This difference makes, among others, our more general problem setting computationally difficult, as shown in Theorem 3 below. Similarly to the infinite closure system over $\mathbb{R}^d$ defined by the family of all convex hulls in $\mathbb{R}^d$, we also assume that the (abstract) closure system is given implicitly via the closure operator. This is a natural assumption, as the cardinality of the closure system is typically exponential in that of the ground set.

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* (i) a closure system $(E, \mathcal{C}_\rho)$ with $|E| < \infty$, where $\mathcal{C}_\rho$ is given by the closure operator ρ which returns in unit time for any $X \subseteq E$ the closure $\rho(X) \in \mathcal{C}_\rho$ and (ii) subsets $A, B \subseteq E$, *decide* whether A and B are half-space separable in $\mathcal{C}_\rho$, or not.

Clearly, the answer is always "No" whenever $\rho(A) \cap \rho(B) \neq \emptyset$, as $\rho(A)$ (resp. $\rho(B)$) are the smallest closed sets in $\mathcal{C}$ containing A (resp. B). 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 result, we adopt the definition of *convex* vertex sets of a graph defined by shortest paths [6]. More precisely, for an undirected graph $G = (V, E)$ we consider the set system $(V, \mathcal{C}_\gamma)$ with

$$V' \in \mathcal{C}_\gamma \iff \forall u, v \in V', \forall P \in \mathcal{S}_{u,v} : V(P) \subseteq V' \quad (1)$$

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. Using the above definition of graph convexity, we consider the following problem definition [1]:

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

Notice that 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 in [1]:

Theorem 2. *The* CONVEX 2-PARTITIONING *problem is NP-complete.*

Using the above concepts and result, we are ready to state the main negative result for this section, by noting that its proof is based on a reduction from the CONVEX 2-PARTITIONING problem.

Theorem 3. *The* HSS *problem is* NP*-complete.*

Furthermore, we can ask for the input $(E, \mathcal{C}_\rho), A, B$ of the HSS problem if there exist disjoint closed sets $H_1, H_2 \in \mathcal{C}_\rho$ with $A \subseteq H_1$ and $B \subseteq H_2$ of *maximum* combined cardinality (i.e., there are no disjoint closed sets $H_1', H_2' \in \mathcal{C}_\rho$ with $A \subseteq H_1'$ and $B \subseteq H_2'$ such that $|H_1| + |H_2| < |H_1'| + |H_2'|$). More precisely, we are interested in the following problem:

MAXIMUM CLOSED SET SEPARATION PROBLEM: *Given* (i) a closure system $(E, \mathcal{C}_\rho)$ as in the HSS problem definition, (ii) subsets $A, B \subseteq E$, and (iii) an integer $k > 0$, *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 $|H_1| + |H_2| \geq k$.

Corollary 4 below is an immediate implication of Theorem 3.

Corollary 4. *The* MAXIMUM CLOSED SET SEPARATION *problem is* NP*-complete.*

The negative results above motivate us to relax below the HSS and the MAXIMUM CLOSED SET SEPARATION problems.

3.2 Maximal Closed Set Separation

One way to overcome the negative results formulated in Theorem 3 and Corollary 4 is to relax 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* (i) a closure system $(E, \mathcal{C}_\rho)$ as in the HSS problem definition, (ii) 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 $H_1 \subsetneq H_1'$ and $H_2 \subsetneq H_2'$, or return "NO" if such sets do not exist.

In this section we present Alg. 1, that solves the MCSS problem and is optimal w.r.t. the worst-case number of closure operator calls. Alg. 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 subsets A, B of 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 (unextended) other 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

Algorithm 1: MAXIMAL CLOSED SET SEPARATION (MCSS)

Input: finite closure system $(E, \mathcal{C}_\rho)$ given by a closure operator ρ and $A, B \subseteq E$
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

extended, then F will be correspondingly 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 Alg. 1:

Theorem 5. *Alg. 1 is correct and solves the MCSS problem by calling the closure operator at most* $2|E| - 2$ *times.*

To state the optimality of Alg. 1 w.r.t. the number of closure operator calls in Corollary 7 below, we first state the following result.

Theorem 6. *There exists no deterministic algorithm solving the MCSS problem calling the closure operator less than* $2|E| - 2$ *times in the worst-case.*

The following corollary is immediate from Theorems 5 and 6.

Corollary 7. *Alg. 1 is optimal w.r.t. the worst-case number of closure operator calls.*

In Section 4 we consider Kakutani closure systems, a special kind of closure systems, for which Alg. 1 solves the HSS problem correctly and efficiently.

4 Kakutani Closure Systems

A natural way to overcome the negative result stated in Theorem 3 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* [4] if it fulfills the S_4 *separation axiom* [5] defined as follows: For all $A, B \subseteq E$, A and B are half-space separable in $(E, \mathcal{C}_\rho)$ if and only if $\rho(A) \cap \rho(B) = \emptyset$. 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 $\mathcal{C}_\rho$. We recall that all closure systems $(E, \mathcal{C})$ considered in this work are finite (i.e., $|E| < \infty$). 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.

The following theorem, one of our main results in this paper, claims that Alg. 1 solving the MCSS problem provides also an *algorithmic characterization* of Kakutani closure systems.

Theorem 8. *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 $A, B \subseteq E$ with $\rho(A) \cap \rho(B) = \emptyset$, the output of Algorithm 1 is a partitioning of E.*

The characterization result formulated in Theorem 8 cannot, however, be used to decide in time polynomial in $|E|$, whether a closure system $(E, \mathcal{C}_\rho)$ is Kakutani, or not if it is given by ρ. More precisely, in Theorem 9 below we have a negative result for the following problem:

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

Theorem 9. *Any* deterministic *algorithm solving the Kakutani problem above requires $\Omega\left(2^{|E|/2}\right)$ closure operator calls.*

While the exponential lower bound in Theorem 9 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 4.1 we focus on graphs. We first present a generalization of a fundamental result [4,5] characterizing Kakutani closure systems over *graphs* by means of the Pasch axiom and mention some potential applications of this generalization result.

4.1 Kakutani Closure Systems over Graphs

As a generic application field of Kakutani closure systems, in this section we focus our attention on *graphs*. For a graph $G = (V, E)$, we consider the closure system $(V, \mathcal{C}_\gamma)$ defined in (1). The following fundamental result provides a characterization of Kakutani closure systems over graphs.

Theorem 10. *[4,5] Let $G = (V, E)$ be a graph. Then $(V, \mathcal{C}_\gamma)$ defined in (1) is Kakutani if and only if γ fulfills the Pasch axiom, i.e.,*

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

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

[4] A similar property was considered by the Japanese mathematician Shizou Kakutani for Euclidean spaces (cf. [9])

[5] For a good reference on convexity structures satisfying the S_4 separation property, the reader is referred e.g. to [4].

The theorem below is an application of Theorem 10 to trees[6]:

Theorem 11. *Let $G = (V, E)$ be a tree. Then $(V, \mathcal{C}_\gamma)$ defined in (1) is Kakutani.*

Besides the direct application of Theorem 11 to vertex classification in trees, it provides also a natural heuristic for vertex classification in *arbitrary* graphs; we discuss this heuristic together with an empirical evaluation in Section 5.

Remark 12. We note that the converse of Theorem 11 does not hold. Indeed, let $G = (V, E)$ be a graph consisting of a single cycle. One can easily check that the corresponding closure system $(V, \mathcal{C}_\gamma)$ defined in (1) is Kakutani, though G is not a tree.

Motivated by potential theoretical and practical applications, in Theorem 13 below we generalize Theorem 10 to a certain type of *structured* set systems. More precisely, a *graph structure partitioning* (GSP) is a triple $\mathfrak{G} = (S, G, \mathcal{P})$, where S is a finite set, $G = (V, E)$ is a graph, and $\mathcal{P} = \{\mathrm{bag}(v) \subseteq S : v \in V\}$ is a partitioning of S into $|V|$ *non-empty* subsets (i.e., $\bigcup_{v \in V} \mathrm{bag}(v) = S$ and $\mathrm{bag}(u) \cap \mathrm{bag}(v) = \emptyset$ for all $u, v \in V$ with $u \neq v$). The set $\mathrm{bag}(v)$ 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

$$\sigma : S' \mapsto \bigcup_{v \in V'} \mathrm{bag}(v) \tag{2}$$

with

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

for all $S' \subseteq S$, where γ is the closure operator corresponding to $(V, \mathcal{C}_\gamma)$ defined in (1). That is, take first the closure $V' \subseteq V$ of the set of vertices of G that are associated with a bag having a non-empty intersection with S' and then the union of the bags for the nodes in V'. We have the following result for σ.

Theorem 13. *Let $\mathfrak{G} = (S, G, \mathcal{P})$ be a GSP with $G = (V, E)$. Then σ defined in (2) is a closure operator on S. Furthermore, the corresponding closure system $(S, \mathcal{C}_\sigma)$ is Kakutani whenever γ corresponding to $(V, \mathcal{C}_\gamma)$ fulfills the Pasch axiom on G.*

Clearly, Theorem 13 generalizes the result formulated in Theorem 10, as 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 $\mathrm{bag}(v) = \{v\}$ for all $v \in V$. Theorem 13 has several potential applications to graphs with vertices associated with the blocks of a partitioning of some set in a bijective manner. This kind of graphs can be obtained for example from graph clustering (see, e.g., [13]) or graph partitioning (see, e.g., [3]) that play an important role e.g. in community network mining.

Another application of Theorem 13 may arise from *quotient* graphs; a graph $G = (V, E)$ is a quotient graph of a graph $G' = (V', E')$ if V is formed by

[6] The claim holds for outerplanar graphs as well. For the sake of simplicity we formulate it in this short version for trees only, as it suffices for our purpose.

the equivalence classes of V' with respect to some equivalence relation ρ (i.e., $V = V'/\rho$) and for all $x, y \in V$, $\{x, y\} \in E$ if and only if $x = [u]_\rho, y = [v]_\rho$ for some $u, v \in V'$ with $\{u, v\} \in E'$, where $[u]_\rho$ (resp. $[v]_\rho$) denotes the equivalence class of u (resp. v). Such a quotient graph can be regarded as a GSP $\mathfrak{G} = (V', G, \mathcal{P})$, where $\mathcal{P}$ is the partitioning of V' corresponding to the equivalence relation ρ and for all $v \in V$, $\text{bag}(v) = [v']_\rho$ if $v = [v']_\rho$ for some $v' \in V'$. Quotient graphs play an important role in logic based graph mining[7] (see, e.g., [14]), which, in turn, can be regarded as a subfield of inductive logic programming (ILP). More precisely, regarding a graph $G' = (V', E')$ as a first-order goal clause $C_{G'}$ (see, e.g., [14]), in ILP one may be interested in finding a subgraph G of G', such that $C_{G'}$ logically implies the first-order goal clause C_G representing G and G is of minimum size with respect to this property. In ILP, C_G is referred to as a *reduced* clause (see [11] for further details on clause reduction); in graph theory G is called the *core* of G'. By the characterization result of subsumptions between clauses in [7], logical implication is equivalent to graph *homomorphism* for the case considered. Thus, G can be considered as the quotient graph of G' induced by φ, where all vertices $v \in V$ are associated with the equivalence class $[v] = \{u \in V' : \varphi(u) = v\}$; the vertices of G' in $[v]$ are regarded structurally equivalent with respect to homomorphism. Note that G is a tree structure of G' whenever G' is a tree, allowing for the same heuristic discussed in Section 5 for arbitrary GSPs.

4.2 Experimental Results

In this section we empirically demonstrate the potential of Alg. 1 on predictive problems over Kakutani closure systems. For this purpose we consider the binary vertex classification problem over free trees. We stress that our main goal with these experiments is to demonstrate that a remarkable predictive performance can be obtained already with the very general version of our algorithm as described in Alg. 1 and with its modification for the case that the closures of the input two sets are not disjoint. The latter case can occur when the sets of vertices belonging to the same class are not half-spaces. Since we do not utilize any domain specific features in our experiments (e.g., some strategy for selecting non-redundant training examples[8]), we do not compare our generic approach to the state-of-the-art algorithms specific to the vertex classification problem.

We evaluate our algorithm on synthetic tree datasets with binary labeled vertices (see below for the details). Formally, for a closure system $(E, \mathcal{C}_\rho)$ let L_r and L_b form a partitioning of E, where the elements of L_r (resp. L_b) will be referred to as *red* (resp. *blue*) vertices. We consider the following supervised

[7] While in ordinary graph mining the pattern matching is typically defined by subgraph isomorphism, it is the graph homomorphism in logic based graph mining, as subsumption between first-order clauses reduces to homomorphism between graphs (see [8] for a discussion).

[8] In case of trees, such a non-redundant set could be obtained by considering only leaves as training examples.

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 of E and an element $e \in E$, *predict* whether $e \in L_r$ or $e \in L_b$. Depending on whether or not L_r (and hence, L_b) forms a half-space in $(E, \mathcal{C}_\rho)$, we consider the following two cases in our experiments:

(i) If L_r (and hence, L_b) is a half-space, then $\rho(R)$ and $\rho(B)$ are always disjoint and hence the algorithm returns some half-spaces $H_r, H_b \in (E, \mathcal{C}_\rho)$ with $R \subseteq H_r$ and $B \subseteq H_b$ because $(E, \mathcal{C}_\rho)$ is Kakutani. The class of e is then predicted by *blue* if $e \in H_b$; o/w by *red*. Note that H_r and H_b can be different from L_r and L_b, respectively.

(ii) If L_r (and hence, L_b) is *not* a half-space in $(E, \mathcal{C}_\rho)$ then $\rho(R) \cap \rho(B)$ can be non-empty. In case of $\rho(R) \cap \rho(B) = \emptyset$, we run Alg. 1 in its original form; by the Kakutani property it always returns two half-spaces H_r and H_b with $R \subseteq H_r$ and $B \subseteq H_b$. The class of e is then predicted in the same way as described in (i). Otherwise (i.e., $\rho(R) \cap \rho(B) \neq \emptyset$), we greedily select a maximal subtree T' such that its vertices have not been considered so far and the closures of the red and blue training examples in T' are disjoint in the closure system corresponding to T'; note that this is also Kakutani.[9] We then run Alg. 1 on this closure system and predict the class of the unlabeled vertices of T' by the output half-spaces as above. We apply this algorithm iteratively until all vertices have been processed.

For the empirical evaluation of the predictive performance of Alg. 1 and its variant described in (ii) above, we used the following synthetic datasets D1 and D2:

D1 For case (i) we considered random trees of size $100, 200, \ldots, 1000, 2000, \ldots, 5000$ (see the x-axes of Fig. 1). For each tree size we then generated 50 random trees and partitioned the vertex set (i.e., E) of each tree into L_r and L_b such that L_r and L_b are half-spaces in $(E, \mathcal{C}_\rho)$ and satisfy $\frac{1}{3} \leq \frac{|L_r|}{|L_b|} \leq 3$.

D2 For case (ii) we proceeded similarly except for the requirement that L_r, L_b are half-spaces. Instead, the labels partition the tree into around 10 maximal subtrees, each of homogeneous labels.

For all trees in D1 and D2 we generated 20 random training sets of different cardinalities (see the y-axes in Fig. 1 and 2). 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. 1 for D1 and in Fig. 2 for D2. For each tree size (x-axes) and training set cardinality (y-axes) we plot the average accuracy obtained for the 1000 learning settings considered. The accuracy is calculated in the standard way, i.e., for a partitioning H_r, H_b of E returned by the algorithm it is defined by

$$\frac{|\{e \in E \setminus D : \ e \text{ is correctly classified}\}|}{|E \setminus D|} ,$$

[9] We formulate this heuristic for trees for simplicity. In the long version we show that this idea can be generalized to any graph satisfying the Pasch axiom.

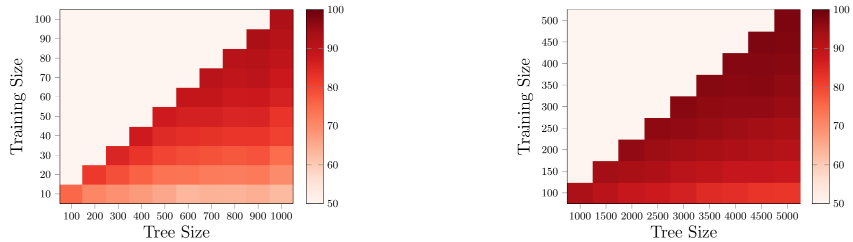


Fig. 1: Accuracy of vertex classifications where labels are half-spaces (cf. dataset D1).

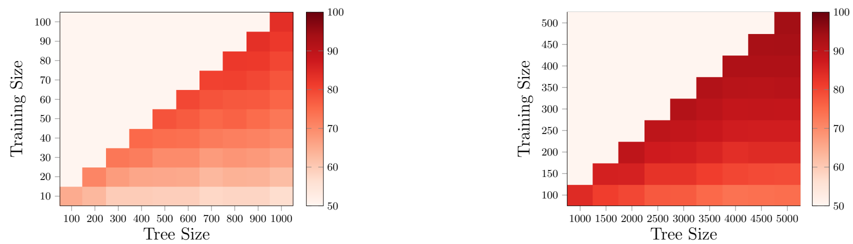


Fig. 2: Accuracy of vertex classification where labels are not half-spaces and partition the tree into around 10 subtrees, each of homogeneous labels (cf. dataset D2).

where D denotes the training set.

Regarding D1 (Fig. 1) 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. 1). 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. 1). The explanation of these surprisingly considerable results raise some interesting theoretical questions for probabilistic combinatorics, as the output half-spaces can be inconsistent with the partitioning formed by L_r, L_b.

Regarding D2 (Fig. 2), we need about 10% training examples to achieve an accuracy of at least 80%, and for trees having at least 600 vertices (see the LHS of Fig. 2). With increasing tree size, the relative amount of training examples decreases to obtain a similar accuracy. In particular, for trees of size 5000, already 150 training examples (i.e., 3%) suffice to achieve 80% accuracy (see the RHS of Fig. 2), indicating that the simple heuristic described in (ii) performs quite well on larger trees. Our further experimental results not presented in this short version suggest that the relative size of the training data depends sublinearly on the number of label homogeneous subtrees.

5 Non-Kakutani Closure Systems

After the discussion of Kakutani closure systems including the negative result on the KAKUTANI problem, in this section we consider non-Kakutani closure systems and show how to extend some of the results of the previous section to this kind of set systems. In particular, we first consider *arbitrary* graphs, which are non-Kakutani, as they do not fulfill the Pasch axiom in general (cf. Thm. 10). As a second type of non-Kakutani closure systems, we then consider finite point configurations in $\mathbb{R}^d$. Although none of these two types of closure systems are Kakutani in general, the experimental results presented in this section show that Alg. 1, combined with a natural heuristic in case of graphs, can effectively be applied to both cases.

The natural heuristic mentioned above reduces the vertex classification problem in non-Kakutani closure systems over arbitrary graphs to Kakutani closure systems by considering random spanning trees of the underlying graph. More precisely, given a graph $G = (V, E)$ and training sets $R \subseteq L_r$ and $B \subseteq L_b$, where L_r and L_b form an unknown partitioning of V, we proceed as follows:

1. we pick a set of spanning trees, each uniformly at random,
2. apply (ii) from Sect. 4.2 to each spanning tree generated with input R and B, and
3. predict the class of an unlabeled vertex by the majority vote of the vertex classification obtained for the spanning trees.

5.1 Experimental Results

Similarly to Section 4.2 on Kakutani closure systems, in this section we empirically demonstrate the potential of Alg. 1 on predictive problems over non-Kakutani closure systems. We first consider the binary vertex classification problem over arbitrary graphs and then over finite point sets in $\mathbb{R}^d$. Similarly to the case of Kakutani closure systems, we do not utilize any domain specific features, as our focus is on measuring the predictive performance of a general-purpose algorithm. In particular, in case of point configurations in $\mathbb{R}^d$ we use only convex hulls (the underlying closure operator), and no other information (e.g. distances). For the empirical evaluations on graphs and on finite point sets in $\mathbb{R}^d$ we used the following synthetic datasets D3 and D4, respectively:

- D3 We generated random connected graphs of size $500, 1000, 1500, 2000$ and edge density (i.e., #edges/#vertices) $1, 1.2, \ldots, 3$. In particular, for each graph size and for each edge density value, 50 random graphs have been picked. We partitioned the vertex set of each graph via that of a random spanning tree into random half-spaces L_r and L_b w.r.t. to the tree's Kakutani closure system. For all labelled graphs generated, the ratio of the vertex labels satisfies $\frac{1}{3} \leq \frac{|L_r|}{|L_b|} \leq 3$.
- D4 We considered randomly generated finite point sets in $\mathbb{R}^d$ for $d = 2, 3, 4$ with labels distributing around two centers. For every $d = 2, 3, 4$, we generated 100 different point sets in $\mathbb{R}^d$, each of cardinality 1000.

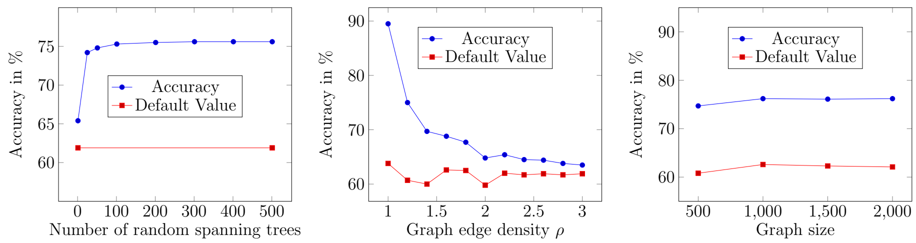
For all graphs in D3 we generated 20 random training sets with 10% of the size of the graphs. The results are presented in Fig. 3. For each number of random spanning tree generated, edge density, and graph size (x-axes) we plot the average accuracy obtained for the 1000 learning settings considered (i.e., 50 graphs $\times$ 20 training datasets). The accuracy is calculated in the same way as above (cf. Sec. 4.2).

In Fig. 3a we first investigate the predictive accuracy depends on the *number of random spanning trees*. One can see that classification via majority vote of around 100 random spanning trees remarkably increases the accuracy over less random spanning trees from 65% to 75%, while considering up to 500 spanning trees has almost no further effect on it. As a trade-off between accuracy and runtime we have therefore fixed the number of spanning trees to (the odd number) 101 for the other experiments.

The results concerning *edge densities* are presented in Fig. 3b. As expected, the edge density has an important effect on the accuracy ranging from 90% for edge density 1 (i.e., trees) to 65% for edge density 3. Notice that for edge density 3, the results are very close to the default value, indicating that our general approach has its remarkable performance on very sparse graphs only. (We recall that except for the closure operator, our algorithm is entirely uninformed regarding the structure.)

Finally, the *graph size* appears to have no significant effect on the predictive performance, as shown in Fig. 3c. For the edge density of 1.2, the accuracy is consistently around 75% for graphs with 500 nodes up to 2000. This is another important positive feature of our algorithm.

For each classification task for finite point sets in $\mathbb{R}^d$ we considered random training sets of different cardinalities for D4 and applied Alg. 1 with the convex hull operator in $\mathbb{R}^d$ to these training data. The prediction has been made by the algorithm's output consisting of two maximal disjoint closed sets. (Note that they are not necessarily half-spaces because the closure system is not Kakutani in general). Accordingly, some of the points have not been classified. To evaluate our approach, we calculated the precision and recall for each problem setting. The results are reported in Fig. 4. Fig. 4a shows that the cardinality of the training set has a significant effect on the accuracy, ranging from 70% to 98% for 10 (i.e., 1%) to 100 (i.e., 10%) training examples, respectively. Note that for small training sets, the precision is very sensitive to the dimension. In particular, the difference is more than 10% for 10 training examples. However, the difference vanishes with increasing training set size. We have carried out experiments with larger datasets as well; the results not presented here for space limitations clearly indicate that the precision 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. Regarding the recall (cf. Fig. 4b), 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 precision.

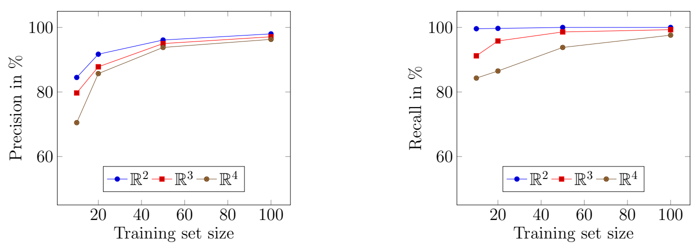


(a) 500 nodes and 600 edges ($\rho = 1.2$) with training set size of 50.

(b) 500 nodes and different edge numbers $e = \rho \cdot n$ with training set size of 50.

(c) Edge density $\rho = 1.2$ and $\frac{n}{10}$ training examples.

Fig. 3: Vertex classification in graphs



(a) Point set size 1000.

(b) Point set size 1000.

Fig. 4: Classification in finite point sets

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

6 Concluding Remarks

The results of this paper show that despite several theoretical difficulties, impressive predictive accuracy can be obtained by a simple 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.

Our approach raises a number of interesting theoretical, algorithmic, and practical questions. In particular, in this paper we deliberately have not utilized any domain specific knowledge (and accordingly, not compared our results to any state-of-the-art algorithm specific to some structure). It would be interesting to specialize Alg. 1 to some particular problem by enriching it with additional information and compare only then its predictive performance to some specific method.

For the theoretical and algorithmic issues, we note that it would be interesting 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. Another interesting 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., [8]). It is an interesting 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.

Acknowledgements

Part of this work has been funded by the Ministry of Education and Research of Germany (BMBF) under project ML2R (grant number 01/S18038C).

References

1. Artigas, D., Dantas, S., Dourado, M., Szwarcfiter, J.: Partitioning a graph into convex sets. Discrete Mathematics **311**(17), 1968 – 1977 (2011)
2. Boser, B., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: Proc. of the 5th Annual ACM Workshop on Computational Learning Theory. pp. 144–152. ACM Press (1992)
3. Buluç, A., Meyerhenke, H., Safro, I., Sanders, P., Schulz, C.: Recent advances in graph partitioning. In: Algorithm Engineering - Selected Results and Surveys, LNCS, vol. 9220, pp. 117–158. Springer (2016)
4. Chepoi, V.: Separation of two convex sets in convexity structures. J. Geom. **50**(1), 30–51 (1994)
5. Ellis, J.W.: A general set-separation theorem. Duke Math. J. **19**(3), 417–421 (1952)
6. Farber, M., Jamison, R.: Convexity in graphs and hypergraphs. SIAM Journal on Algebraic Discrete Methods **7**(3), 433–444 (1986)
7. Gottlob, G.: Subsumption and implication. Inform. Process. Lett. **24**(2), 109–111 (1987)
8. Horváth, T., Turán, G.: Learning logic programs with structured background knowledge. Artificial Intelligence **128**(1-2), 31–97 (2001)
9. Kakutani, S.: Ein Beweis des Satzes von Edelheit über konvexe Mengen. Proc. Imp. Acad. Tokyo **13**, 93–94 (1937)
10. Kubiś, W.: Separation properties of convexity spaces. J. Geom. **74**(1), 110–119 (2002)
11. Plotkin, G.: A note on inductive generalization. Mach. Intell. **5**, 153–163 (1970)
12. Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review **65**, 386–408 (1958)
13. Schaeffer, S.E.: Survey: Graph clustering. Comput. Sci. Rev. **1**(1), 27–64 (2007)
14. Schulz, T.H., Horváth, T., Welke, P., Wrobel, S.: Mining tree patterns with partially injective homomorphisms. In: Machine Learning and Knowledge Discovery in Databases. LNCS, vol. 11052, pp. 585–601. Springer (2019)
15. van de Vel, M.: Binary convexities and distributive lattices. Proc. London Math. Soc. **48**(1), 1–33 (1984)
16. van de Vel, M.: Theory of Convex Structures, Volume 50 (North-Holland Mathematical Library). North Holland (1993)