Identifying Clusters from Positive Data*

John Case[1], Sanjay Jain[2], Eric Martin[3], Arun Sharma[4] and Frank Stephan[4]

[1] Computer and Information Sciences Department, 101A Smith Hall, University of Delaware, Newark, DE 19716-2586, United States of America, case@cis.udel.edu
[2] School of Computing, National University of Singapore, Singapore 119260, sanjay@comp.nus.edu.sg
[3] School of Computer Science and Engeniering, UNSW Sydney NSW 2052, Australia, emartin@cse.unsw.edu.au
[4] National ICT Australia, Sydney Research Laboratory at Kensington, UNSW Sydney NSW 2052, Australia, arun.sharma@nicta.com.au and frank.stephan@nicta.com.au

Abstract. The present work studies clustering from an abstract point of view and investigates its properties in the framework of inductive inference. Any class S considered is given by a numbering $A_0, A_1, \ldots$ of nonempty subsets of $\mathbb{N}$ or $\mathbb{Q}^k$ which is used as a hypothesis space. A clustering task is a finite and nonempty set of indices of pairwise disjoint sets. The class S is said to be clusterable if there is an algorithm which, for every clustering task I, converges in the limit on any text for $\cup_{i \in I} A_i$ to a finite set J of indices of pairwise disjoint clusters such that $\cup_{j \in J} A_j = \cup_{i \in I} A_i$. A class is called semiclusterable if there is such an algorithm which finds a J with the last condition relaxed to $\cup_{j \in J} A_j \supseteq \cup_{i \in I} A_i$.

The relationship between natural topological properties and clusterability is investigated. Topological properties can provide sufficient or necessary conditions for clusterability but they cannot characterize it. On one hand, many interesting conditions make use of both the topological structure of the class and a well-chosen numbering. On the other hand, the clusterability of a class does not depend on the decision which numbering of the class is used as a hypothesis space for the clusterer.

These ideas are demonstrated in the context of geometrically defined classes. Clustering of many of these classes requires besides the text for the clustering task some additional information: the class of convex hulls of finitely many points in a rational vector space can be clustered with the number of clusters as additional information. Interestingly, the class of polygons (together with their interiors) is clusterable if the number of clusters and the overall number of vertices of these clusters is given to the clusterer as additional information. Intriguingly this additional information is not sufficient for classes including fig-

* J. Case is supported in part by NSF grant number CCR-0208616 and USDA IFAFS grant number 01-04145. S. Jain is supported in part by NUS grant number R252-000-127-112. A. Sharma and F. Stephan work at National ICT Australia which is funded by the Australian Government's Department of Communications, Information Technology and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Centre of Excellence Program.

ures with holes.

While some classes are unclusterable due to their topological structure, others are only computationally intractable. Oracles can be used to distinguish between both cases: the former cannot be clustered using any oracle while the latter can be clustered using some oracle. It is shown that there are maximal oracles that allow clustering of all problems that can be clustered with the help of some oracle. In particular, E is maximal iff $E \geq_T K \wedge E' \geq_T K''$. Furthermore, no 1-generic oracle below K and no 2-generic oracle permits to cluster any class which is not clusterable without an oracle.

Keywords. Inductive inference, clustering, numbering, Turing degree, topological and geometric properties of clusterable classes.

1 Introduction

The aim of the paper is to study the role of computation and topology in the clustering process. To this aim, the following topics are investigated in an abstract model of clustering:

1. necessary or sufficient topological conditions for clustering;
2. various relationships between clustering, learning and hypothesis spaces;
3. clusterability of many natural classes of geometrically defined objects;
4. oracles as a method to distinguish between topological and computational aspects of clustering.

Clustering has been widely studied in several forms in the fields of machine learning and statistics [2, 5, 17, 19]. However abstract treatments of the topic are rare. Kleinberg [14] provides an axiomatic approach to clustering, but, in his settings, computability per se is not an issue. In contrast, the present work investigates clustering from the perspective of Gold style learning theory where limitations stem from uncomputable phenomena.

The basic setting is that a class of potential clusters is given. This class is recursively enumerable. A finite set I of (indices of) pairwise disjoint clusters from the given class is called a *clustering task*. Given such a task, the clusterer – which might be any algorithmic device – receives a text containing all the data occurring in these clusters and is supposed to find in the limit a set J of (indices of) pairwise disjoint clusters which cover all the data to be seen. There are two variants with respect to a third condition: if one requires that the union of the clusters given by I is the same as the union of the clusters given by J, then one refers to this problem as *clustering*; if this condition is omitted, then one refers to this problem as *semiclustering*.

Clustering might be in many cases more adequate than semiclustering: for example the clustering tasks from the class $S_{\mathrm{conv},k}$ defined in Definition 8.1 are collections of convex sets having a positive distance from each other. The solution to such a clustering task is unique since each of these sets corresponds to a cluster. A clusterer has to identify these sets while a semiclusterer can just converge to the convex hull of all data to be seen. Such a solution is legitimate for semiclustering since it is again a member of the class $S_{\mathrm{conv},k}$. But it fails to meet the intuition behind clustering at all since it does not distinguish the data from the various clearly different clusters.

Note that in the process of clustering, it is sufficient to find the set J of indices mentioned above. From this J one can find for every data-item x in the set $\cup_{j\in J} A_j$ of all permitted data the unique cluster where x belongs to. One just enumerates the sets with the indices in J until the data-item appears in one of them and then uses the index of this set as a description for the cluster to which this data-item belongs. So, from a recursion-theoretic point of view, finding the set J is the relevant part of a given clustering problem.

For every indexed class of recursively enumerable sets there is a canonical translation from these indices to type-0 grammars in the Chomsky hierarchy which generate the corresponding sets. This links the current setting of clustering to grammatical inference although there is no need herein to exploit the detailed structure of the grammars obtained by such a translation.

1. A class has the Finite Containment Property iff any finite union of its members contains only finitely many other members. In Section 5 it is shown that classes satisfying this natural property separate the basic notions of clusterability, semi-clusterability and learnability. There is no purely topological characterization of clusterable classes: if a class contains an infinite set C and all singleton sets disjoint from C then the class is clusterable iff C is recursive. Proposition 6.1 gives a further characterization which depends on the numbering: a class of disjoint sets is clusterable iff it has a numbering where every set occurs only finitely often. Section 6 provides some further sufficient criteria for clusterability which take into account topological aspects as well as properties of the given hypothesis space. These criteria are refinements of the Finite Containment Property.

2. Clusterable classes are learnable but not vice versa. Although clusterable classes are by definition uniformly recursively enumerable, the set of clustering tasks might fail to be. Proposition 3.2 shows that a clusterable class can be clustered using any class preserving hypothesis space, that is, any hypothesis space which enumerates the members of the class only. But by Example 3.3 a clusterable class might not be clusterable with respect to some class comprising hypothesis space, that is, some hypothesis space which enumerates the members of a superclass.

3. In Sections 7 and 8 it is demonstrated how one can map down concrete examples into this general framework. These concrete examples are geometrically defined subsets of $\mathbb{Q}^k$: affine sets, classes of sets with distinct accumulation points and convex hulls of finite sets. This third example is not clusterable but it turns out to be clusterable if some additional information about the task given to the clusterer is revealed. While there are several natural candidates for the additional information in the case of convex hulls of finite sets, this approach becomes much more difficult when dealing with clusters of other shapes. In the case of polygons

in the 2-dimensional space, the additional information provided might consist of the number of clusters plus the overall number of vertices in the polygons considered. Still this additional information is insufficient for clustering classes of geometric object some of which have holes. But the k-dimensional area is a sufficient additional information as long as one rules out that the symmetric difference of two clusters has the k-dimensional area 0.

4. Oracles are a way to distinguish between topological and computational difficulty of a clustering problem. In Section 4 the relationship between an oracle E and the classes clusterable relative to E is investigated. For example, every 1-generic oracle E which is Turing reducible to the halting problem is trivial: every class which is clusterable relative to E is already clusterable without any oracle. On the other hand, some classes are even not clusterable relative to any oracle. Proposition 4.3 characterizes the maximal oracles which permit to cluster any class which is clusterable relative to *some* oracle; in particular it is shown that such oracles exist.

2 The Basic Model

Most of the notation follows [11, 18]. The next paragraph summarizes the most important notions used in the present work.

Basic Notation 2.1. A class S is assumed to consist of recursively enumerable subsets of a countable underlying set $\mathbb{U}$ where in Sections 3–6, $\mathbb{U}$ is the set of natural numbers $\mathbb{N}$ and in Sections 7 and 8, $\mathbb{U}$ is a rational vector space of finite positive dimension. Mostly, S is even required to be *uniformly recursively enumerable* which means that there is a sequence $A_0, A_1, \ldots$ of subsets of $\mathbb{U}$ such that first $S = \{A_0, A_1, \ldots\}$ and second $\{(i, x) \in \mathbb{N} \times \mathbb{U} : x \in A_i\}$ is recursively enumerable. Such a sequence $A_0, A_1, \ldots$ is called a *numbering* for S.

The letters I, J, H always range over finite subsets of $\mathbb{N}$ and furthermore $\text{norm}(I) = \sum_{i \in I} 2^i$ is called the norm of I. Define A_I as $\cup_{i \in I} A_i$. Let $A_{i,s}$ denote the set of elements enumerated into A_i within s steps and let $A_{I,s} = \cup_{i \in I} A_{i,s}$. Without loss of generality $A_{i,s} \subseteq \{0, 1, \ldots, s\}$ for all s. The sets $A_{i,s}$ are uniformly recursive, that is, $\{(i, s, x) : x \in A_{i,s}\}$ is recursive.

Let $\text{disj}(S)$ contain all finite sets I such that $A_i \cap A_j = \emptyset$ for all different $i, j \in I$. The sets in $\text{disj}(S)$ are called *clustering tasks*. There is an approximation $\text{disj}_s(S)$ to $\text{disj}(S)$ such that $I \in \text{disj}_s(S)$ if $A_{i,s} \cap A_{j,s} = \emptyset$ for all different $i, j \in I$.

For any set A let $|A|$ be the cardinality of A. Let A^* be the set of all finite sequences of members of A and $|\sigma|$ be the length of a string $\sigma \in A^*$.

A text for a nonempty set $A \subseteq \mathbb{U}$ is any infinite sequence containing all elements but no nonelements of A. Clusterers and semiclusterers are recursive functions from $\mathbb{U}^*$ to finite subsets of $\mathbb{N}$, learners are recursive functions from $\mathbb{U}^*$ to $\mathbb{N}$. Also mappings M^E represented by a machine M having access to an oracle E are considered. An element $\sigma \in A^*$ is called a *stabilizing sequence* (*for*

A and M) if $M(\sigma\tau) = M(\sigma)$ for all $\tau \in A^*$.

The sequence $W_0, W_1, \ldots$ denotes an acceptable numbering of all recursively enumerable sets and W_e can be interpreted as the domain of the e-th partial-recursive function φ_e. The set $K = \{e : e \in W_e\}$ is called the halting problem and this notion can be generalized to computation relative to oracles: A' is the halting problem relative to A; in particular K' is the halting problem relative to K and K'' the one relative to K'. For more information on iterated halting problems see [18, page 450].

Definition 2.2. A class $S = \{A_0, A_1, \ldots\}$ of clusters is called *clusterable* iff there is a clusterer M which, for every $I \in \operatorname{disj}(S)$, converges on every text for A_I to a $J \in \operatorname{disj}(S)$ with $A_J = A_I$. Such an M is called a *clusterer for S*.

S is called *semiclusterable* if one replaces $A_J = A_I$ by the weaker condition that $A_J \supseteq A_I$.

S is called *learnable in the limit from positive data with respect to the hypothesis space* $A_0, A_1, \ldots$ iff there is a learner M which for every $i \in \mathbb{N}$ converges on every text for A_i to a $j \in \mathbb{N}$ with $A_j = A_i$. In the following "*learnable*" stands for "learnable in the limit from positive data with respect to the hypothesis space $A_0, A_1, \ldots$".

Note that every learner, clusterer or semiclusterer M which succeeds on A has a stabilizing sequence $\sigma \in A^*$. Furthermore, $M(\sigma)$ is then also a correct hypothesis for A.

Remark 2.3. A clusterer M for $S = \{A_0, A_1, \ldots\}$ might also use a different hypothesis space instead of the default one. Here a numbering $B_0, B_1, \ldots$ is called the *hypothesis space of M* iff for every clustering task I and any text for A_I, M converges on this text to a finite set J such that $B_J = A_I$ and $B_i \cap B_j = \emptyset$ for all different $i, j \in J$. The hypothesis space is *class preserving* if $S = \{B_0, B_1, \ldots\}$ and *class comprising* if $S \subseteq \{B_0, B_1, \ldots\}$. Nevertheless, in the light of Proposition 3.2, it is assumed that a clusterer uses the default numbering $A_0, A_1, \ldots$ as its hypothesis space unless explicitly stated otherwise.

Remark 2.4. Many learning criteria have analogous definitions for clustering. For example, a machine M is confident iff it converges on every input to some hypothesis. So one could consider the notion of confidently clusterable classes. This notion is more restrictive, that is, there are classes which are clusterable but not confidently clusterable. In many respects, the theory developed on the basis of these notions is very similar to the corresponding one for learning due to the following reason.

Many separations of different criteria C_1 and C_2 in learning from positive data can be carried over to separations of the corresponding criteria $\tilde{C}_1$ and $\tilde{C}_2$ in clustering. Given a class S separating the learning criterion C_1 from C_2, one can consider the class

$$\tilde{S} = \{\tilde{A} : A \in S\} \text{ where } \tilde{A} = \{0\} \cup \{x + 1 : x \in A\}$$

to separate $\tilde{C}_1$ from $\tilde{C}_2$. The main idea is to use the 0 in order to avoid that any two members of $\tilde{S}$ are disjoint. Then every clustering task and every reasonable hypothesis is a singleton set. Learners for S and clusterers for $\tilde{S}$ can be translated into each other.

For example, there is a class S which is learnable but not confidently learnable. Then the class $\tilde{S}$ is clusterable but not confidently clusterable. That is, $\tilde{S}$ witnesses that the notion of confident clustering is more restrictive than the notion of clustering.

This explains some of the many similarities between learning and clustering. Thus the present work is not focussing on the introduction and study of clusterability notions parallel to the many variants of learning in the limit. Instead the emphasis is more given on the relations between clusterability on one side and topological, recursion-theoretic and geometrical properties of classes under consideration on the other side.

3 Numberings and Clustering

The main topic of this section is to investigate the role of numberings in clustering. A natural question is whether clustering is independent of the numbering chosen as the hypothesis space. Another important issue is the relationship between numberings of the class of clusters and numberings of the class of finite disjoint unions of clusters. The latter, which represents the clustering tasks, might not have a numbering despite of the fact that the former does, as shown in the next example. The class of sets representing the clustering tasks in this example cannot be made recursively enumerable by changing the numbering of the class of clusters.

Example 3.1. *Let $A_0 = \{0\}$ and let, for every $i \in \mathbb{N}$ and $j \in \{1, 2\}$,*

$$A_{2i+j} = \begin{cases} \{2i+j\} & \text{if } i \notin K; \\ \{0, 2i+j\} & \text{if } i \in K. \end{cases}$$

The class $S = \{A_0, A_1, \ldots\}$ is uniformly recursively enumerable but the class $\{A_I : I \in disj(S)\}$ is not since

$$i \notin K \Leftrightarrow (\exists I \in disj(S))\, [\{2i+1, 2i+2\} \subseteq A_I].$$

This connection holds for all numberings of S but fails for any numbering of the superclass of all finite sets.

Thus there are clusterable classes where the corresponding class of all clustering tasks does not have a numbering. Nevertheless, the result of de Jongh and Kanazawa [4] carries over to clustering: whenever a class is clusterable with respect to a class comprising hypothesis space then the class is also clusterable with respect to every class preserving hypothesis space. Actually the following result is even a bit stronger since it does not require that the hypothesis space

$B_0, B_1, \ldots$ is class preserving but only that it satisfies the following more technical but also more general condition:

$$S \subseteq \{B_J : (\forall i, j \in J)\,[i = j \vee B_i \cap B_j = \emptyset]\}.$$

An application of the next result is that every uniformly recursively enumerable class consisting only of finite sets is clusterable.

Proposition 3.2. *Let $A_0, A_1, \ldots$ be a numbering of a class S and $B_0, B_1, \ldots$ be another numbering such that for every $I \in disj(S)$ there is a J with $A_I = B_J$. If there is a clusterer for S using the hypothesis space $B_0, B_1, \ldots$ then there is another clusterer that uses the original numbering $A_0, A_1, \ldots$ as its hypothesis space.*

Proof. Assume that M is a clusterer for S using the numbering $B_0, B_1, \ldots$ as its hypothesis space. Note that M is required to be correct only on tasks from S, whereas the superclass $\{B_0, B_1, \ldots\}$ is not required to be clusterable.

The clusterer M has on every A_I with $I \in \mathrm{disj}(S)$ a stabilizing sequence σ_I which can be found in the limit: $\sigma_I = \lim_s \sigma_{I,s}$ with $\sigma_I, \sigma_{I,t} \in A_I^*$ and $M(\sigma_I \tau) = M(\sigma_I)$ for all I, t and all $\tau \in A_I^*$.

Then the following clusterer N uses $A_0, A_1, \ldots$ as its hypothesis space.

Algorithm N. On input of length s, N computes the output J of M fed with the same input and searches for the set $H \subseteq \{0, 1, \ldots, s\}$ of least norm satisfying the following conditions:

- $H \in \mathrm{disj}_s(S)$;
- $\sigma_{H,s} \in B_{J,s}^*$;
- $M(\sigma_{H,s}\tau) = M(\sigma_{H,s})$ for all $\tau \in B_{J,s}^*$ of length up to s.

If H is found then output H else output $\emptyset$.

Verification. Given a task and a text for this task, let J be the hypothesis to which M converges. Let I be the set of least norm such that $A_I = B_J$ and $I \in \mathrm{disj}(S)$. Note that for all H with $\mathrm{norm}(H) < \mathrm{norm}(I)$, either $H \notin \mathrm{disj}(S)$ or $\mathrm{range}(\sigma_H) \not\subseteq B_J$ or there is a $\tau \in B_J^*$ such that $M(\sigma_H \tau) \neq M(\sigma_H)$. Thus, if the length s of the part of the text fed into N is sufficiently large, then the following properties hold:

- $I \subseteq \{0, 1, \ldots, s\}$;
- for all H with $\mathrm{norm}(H) \leq \mathrm{norm}(I)$, $H \in \mathrm{disj}_s(S) \Leftrightarrow H \in \mathrm{disj}(S)$;
- for all $H \in \mathrm{disj}(S)$ with $\mathrm{norm}(H) \leq \mathrm{norm}(I)$, $\sigma_{H,s} = \sigma_H$;
- M outputs J and $\sigma_I \in B_{J,s}^*$;
- for all $H \in \mathrm{disj}(S)$ with $\mathrm{norm}(H) < \mathrm{norm}(I)$, either $\sigma_H \notin B_J^*$ or there is a $\tau \in B_{J,s}^*$ of length up to s with $M(\sigma_H \tau) \neq M(\sigma_H)$.

Hence I satisfies the search conditions of N but no H with $\mathrm{norm}(H) < \mathrm{norm}(I)$ does. Thus N converges on a text for B_J to the set I and N witnesses that S is clusterable using the hypothesis space $A_0, A_1, \ldots$ for S. ∎

Example 3.3. *The converse of Proposition 3.2 does not hold: there is a clusterable class S and a numbering of a superclass such that no clusterer for S can use this numbering as a hypothesis space.*

Proof. For every i let $A_i = \{\langle i, x\rangle : x \leq |W_i|\}$ and let $S = \{A_0, A_1, \ldots\}$. It is easy to see that S is clusterable using the numbering $A_0, A_1, \ldots$ as the hypothesis space: on input σ the clusterer just outputs $\{i : \langle i, 0\rangle \in \mathrm{range}(\sigma)\}$.

For better readability, the second numbering has two indices. One defines that $B_{i,j} = \{\langle i, x\rangle : \min(\{j, x\}) \leq |W_i|\}$. Note that $B_{i,j} = \{\langle i, 0\rangle, \langle i, 1\rangle, \ldots\}$ iff either W_i is infinite or $j \leq |W_i|$.

Assume by way of contradiction that M is a clusterer for S using the second numbering as its hypothesis space. Given any i, M converges on every text for A_i to a singleton $\{(i, j)\}$ with $A_i = B_{i,j}$. If W_i is finite then $j > |W_i|$. Thus one can compute relative to K whether W_i is finite:

1. taking a default enumeration of A_i, one can use K to determine the j such that M – using this enumeration as a text for A_i – converges to $\{(i, j)\}$;
2. one can use K to determine whether $|W_i| > j$;
3. if so, W_i is infinite, if not, W_i is finite.

This K-recursive algorithm contradicts to the fact that the set $\{i : W_i \text{ is finite}\}$ has the same Turing degree as K' and gives the desired contradiction. ∎

Although there are classes $S = \{A_0, A_1, \ldots\}$ such that $\{A_I : I \in \mathrm{disj}(S)\}$ is not uniformly recursively enumerable, the superclass $\{A_I : I \subseteq \mathbb{N} \wedge |I| \text{ is finite}\}$ is uniformly recursively enumerable. A clusterer is at the same time a learner for S using the hypothesis space given by the numbering $B_0, B_1, \ldots$ which satisfies $B_{\mathrm{norm}(I)-1} = A_I$ for all nonempty sets I. But learnability of uniformly recursively enumerable classes does not depend on the hypothesis space; following a result of de Jongh and Kanazawa [4] there is also a learner for S which uses $A_0, A_1, \ldots$ as its hypothesis space. So every clusterable class is learnable although the converse direction does not hold.

Property 3.4. *Every clusterable class is learnable.*

Examples 3.5. 1. *The class S_{gold} consisting of $\mathbb{N}$ and all its finite subsets is neither learnable nor clusterable. But S_{gold} is semiclusterable.*

2. *The class S_{sing} consisting of all singletons $\{x\}$ and the set $\mathbb{N}$ is learnable and semiclusterable but not clusterable.*

3. *Let C be infinite and recursively enumerable. The class S_C consisting of C and all singletons $\{x\}$ disjoint from C is learnable. Furthermore, S_C is clusterable iff S_C is semiclusterable iff C is recursive.*

Proof. 1. Gold [9] observed that S_{gold} is not learnable. By Property 3.4, the class S_{gold} is also not clusterable. But S_{gold} is semiclusterable by the trivial algorithm which always conjectures an index for $\mathbb{N}$.

2. The class S_{sing} is learnable by the algorithm which conjectures an index for $\text{range}(\sigma)$ if $|\text{range}(\sigma)| = 1$ and an index for $\mathbb{N}$ if $|\text{range}(\sigma)| \neq 1$. Since every finite set belongs to a clustering task from S_{sing}, the structure of the clustering tasks of S_{sing} is equal to the one of S_{gold}. So S_{sing} is semiclusterable but not clusterable.

3. Note that the class S_C has a numbering by taking $A_i = C$ if $i \in C$ and $A_i = \{i\}$ otherwise. One first enumerates i into A_i and whenever i shows up in C, then one enumerates also the other elements of C into A_i.

The class S_C can be learned by conjecturing the cluster A_i for the first number i occurring in the text; once selected, the output is kept forever.

If C is recursive, then S_C is clusterable: on input σ, one outputs $\text{range}(\sigma)$ if $\text{range}(\sigma) \cap C = \emptyset$ and $\{\min(C)\} \cup (\text{range}(\sigma) - C)$ otherwise. What this algorithm does is that it outputs the set containing the minimal indeces of the clusters which intersect the set of data items seen so far. Note that every clusterer is also a semiclusterer. So S_C is semiclusterable as well.

It remains to show that C is recursive whenever there is a semiclusterer M for S_C. The set C has a stabilizing sequence σ with respect to M. Now let $J = M(\sigma)$. There is a finite and possibly empty set D disjoint from C such that $A_J = C \cup D$. So one has that

$$x \notin C \Leftrightarrow x \in D \vee (\exists \tau \in C^*)\,[M(\sigma x \tau) \neq M(\sigma)].$$

These formulas witness that $\overline{C}$ is recursively enumerable. Since C itself is also recursively enumerable, the set C is actually recursive. ∎

The classes S_{sing} and S_C where C is nonrecursive are learnable but not clusterable. Both have the property that they are not closed under disjoint union. The next result shows that this property is essential for getting examples which are learnable but not clusterable.

Property 3.6. *Let the class S be closed under disjoint union, that is, $A \cup B \in S$ for all disjoint $A, B \in S$. Then S is clusterable iff S is learnable.*

A learner M for a class S is called *prudent* if it only outputs indices of sets it learns. One can enumerate all possible hypotheses $e_0, e_1, \ldots$ of M and so obtain a numbering $B_0, B_1, \ldots$ with $B_i = W_{e_i}$ of a learnable superclass of S. Fulk [8] showed that every learnable class has a prudent learner. Therefore, it is sufficient to consider only uniformly recursively enumerable classes for learning. So Fulk's result can be stated as follows.

Property 3.7 [11, Proposition 5.20]. *Every learnable class has a prudent learner. In particular, every learnable class is contained in some learnable and uniformly recursively enumerable class.*

So every learnable class can be extended to one which is learnable and uniformly recursively enumerable. But in contrast to learning in the limit, this requirement turns out to be restrictive for clustering. Indeed, Proposition 3.9 below gives for every $\{0,1\}$-valued function $F \not\leq_T K''$ a clusterable class which is not contained in any uniformly recursively enumerable clusterable class. Furthermore, the union of any two such classes, given by different functions F, F', is no longer clusterable. So one cannot cover these classes by countably many clusterable superclasses. Most interesting results are based on Definition 2.2 with the consequence that only countably many classes are clusterable. The more general notion below expands the collection of clusterable classes to an uncountable one. Although the latter collection contains many irregular classes of limited interest, it still gives some fundamental insights. In this case one uses the acceptable numbering $W_0, W_1, \ldots$ of all recursively enumerable sets as the hypothesis space for the clusterer.

Definition 3.8. A class S of recursively enumerable sets is *clusterable in the general sense* iff there is a machine M which converges on every text for the union of finitely many disjoint sets $L_0, L_1, \ldots, L_n \in S$ to a finite set J of indices of pairwise disjoint members of S such that $L_0 \cup L_1 \cup \ldots \cup L_n = \cup_{e \in J} W_e$.

Proposition 3.9. *Let F be a $\{0,1\}$-valued function which is not computable relative to the oracle K''. For all $x, y \in \mathbb{N}$ and $z \in \{0,1\}$ let $A_{x,z}, B_{x,y}$ be defined as*

$$A_{x,z} = \{\langle x,0,z\rangle, \langle x,1,z\rangle, \langle x,2,z\rangle, \ldots\};$$
$$B_{x,y} = \{\langle x,y,0\rangle, \langle x,y,1\rangle\}.$$

Then the class S containing all sets $A_{x,z}$ and $B_{x,y}$ with $x, y \in \mathbb{N}$ and $z = F(x)$ is clusterable but not contained in any clusterable class which is uniformly recursively enumerable.

Proof. A clustering algorithm outputs on input σ a set J which contains indices of the following sets:

- $A_{x,z}$ whenever $\langle x,0,z\rangle \in \text{range}(\sigma)$ but $\langle x,0,1-z\rangle \notin \text{range}(\sigma)$;
- $B_{x,y}$ whenever $B_{x,y} \subseteq \text{range}(\sigma)$.

The verification of the correctness of this algorithm can be carried out by taking into account that for every x the following holds: S contains exactly one of the sets $A_{x,0}, A_{x,1}$; every clustering task never contains both $A_{x,z}$ and $B_{x,y}$.

Assume by way of contradiction that $C_0, C_1, \ldots$ is a numbering of a clusterable superclass of S. This numbering contains exactly only one of the sets $A_{x,0}, A_{x,1}$ since $A_{x,F(x)} \in S$ and every class containing both $A_{x,0}, A_{x,1}$ together with the sets $B_{x,y}$ for all $y \in \mathbb{N}$ is not clusterable. The class of all $A_{x,0}, A_{x,1}$ and $B_{x,y}$ has a basic principle with the class S_{sing} from Examples 3.5 in common: the set $A_{x,0} \cup A_{x,1}$ is the disjoint union of the subsets $B_{x,0}, B_{x,1}, \ldots$ and therefore no clusterable superclass of S contains both sets $A_{x,0}$ and $A_{x,1}$. Thus one can get F from the numbering $C_0, C_1, \ldots$ as follows:

$$F(x) = z \Leftrightarrow (\exists i)\,[C_i = A_{x,z}].$$

Since the equality of two recursively enumerable sets can be tested relative to the oracle K', the function F would be computable relative to K'' in contradiction to the choice of F. ∎

4 Clustering and Oracles

Oracles are a method to measure the complexity of a problem. Some classes are clusterable with a suitable oracle while others cannot be clustered with any oracle. So the use of oracles permits to distinguish between problems caused by the computational difficulty of the class involved from those which are unclusterable for topological reasons. This is illustrated in the following remark.

Remark 4.1. Recall the classes S_C and S_{gold} from Examples 3.5. The class S_C is clusterable iff the set C in its definition is recursive. It is easy to see that supplying C as an oracle to the clusterer resolves all computational problems in the case that C is not recursive. But the class S_{gold} is unclusterable because of its topological structure and remains unclusterable relative to every oracle.

Oracles have been extensively studied in the context of inductive inference [1, 6, 13, 16]. These studies considered arbitrary classes and not uniformly recursively enumerable ones. The results for arbitrary classes carry over directly from learning to clustering in the general sense.

Remark 4.2. Fortnow and coworkers [6] investigated the question which oracles are *maximal for learning* in the sense that they enable to solve all principally solvable learning problems. Jain and Sharma [13] showed that there is no maximal oracle. The same holds for clustering: for every oracle E the class S^E_{jump} consisting of all sets $\{2x, 2x+1\}$ with $x \in E'$ and $\{2x\}, \{2x+1\}$ with $x \notin E'$ is clusterable in the general sense relative to oracle F iff $E' \leq_T F'$. The reason is that a clusterer M^F on a text for $\{2x, 2x+1\}$ can figure out in the limit how many clusters of S^E_{jump} are needed to cover $\{2x, 2x+1\}$:

$$x \in E' \Leftrightarrow M^F \text{ converges on } 2x\,(2x+1)\,(2x+1)\ldots \text{ to } I \text{ with } |I| = 1;$$
$$x \notin E' \Leftrightarrow M^F \text{ converges on } 2x\,(2x+1)\,(2x+1)\ldots \text{ to } I \text{ with } |I| = 2.$$

Thus there is no oracle E which is *maximal for clustering in the general sense*, meaning that every class which is clusterable in the general sense relative to some oracle is also clusterable in the general sense relative to E.

An oracle is called *trivial for clustering in the general sense* iff every class which is clusterable in the general sense relative to this oracle is also clusterable in the general sense without it. Now it is shown that a nonrecursive oracle E is trivial for clustering in the general sense iff it has 1-generic degree and is Turing reducible to the halting problem, that is, Case 1 below is satisfied.

Case 1: $E \leq_T G$ for a 1-generic set $G \leq_T K$. Let S be any class which is clusterable in the general sense relative to E. By [6, Lemma 4.19] there is a clusterer M^E which asks on every text belonging to any clustering task from S only finitely many queries to E. The answers to these queries can be successfully figured out in the limit and thus there is a recursive clusterer for S which converges on every text of any finite disjoint union of sets in S to exactly the same output as M^E. In particular, E is trivial for clustering in the general sense.

Case 2: $E \not\leq_T G$ for any 1-generic set $G \leq_T K$. Kummer and Stephan [16, Theorem 10.5] showed that there is a class S^E_{func} which is learnable relative to E but not without any oracle. This class S^E_{func} consists of graphs of recursive functions and following Remark 2.4, one can assume without loss of generality that $f(0) = 0$ for every function whose graph is in S^E_{func}. The class S^E_{func} is on one hand clusterable in the general sense relative to E and on the other hand not clusterable in the general sense without any oracle. In particular, S^E_{func} witnesses that E is not trivial for clustering in the general sense.

The previous remark completes the investigation of clustering in the general sense within the present work. From now on, S denotes again a uniformly recursively enumerable family of clusters. That is, $S = \{A_0, A_1, \ldots\}$ and the set $\{(i,x) \in \mathbb{N}^2 : x \in A_i\}$ is recursively enumerable.

The usefulness of oracles with respect to clustering differs much from the case of clustering in the general sense. Dealing only with uniformly recursively enumerable classes reduces the ability to separate oracles by suitable classes. The definitions for maximal and trivial oracles for clustering are the following ones.

Call an oracle E *maximal for clustering* if every uniformly recursively enumerable class which is clusterable relative to some oracle is already clusterable relative to E. Call an oracle E *trivial for clustering* if every uniformly recursively enumerable class which is clusterable relative to E is already clusterable without any oracle.

Here the word "maximal" instead of "omniscient" is used since by Remark 4.1 some classes are not clusterable with any oracle. In contrast, omniscient oracles for learning functions permit to learn the class of all recursive functions [1] and do not leave any function learning problem unsolved.

The next result shows that in contrast to the case of clustering in the general sense there are maximal oracles for clustering. The characterization for trivial oracles for function learning [6] can be carried over to clustering with respect to oracles below K.

Proposition 4.3. *For every oracle E the following statements are equivalent:*

(a) *$E \geq_T K$ and $E' \geq_T K''$;*
(b) *the oracle E is maximal for learning from positive data – every uniformly recursively enumerable class is either not learnable with any oracle or learnable with oracle E;*

(c) *the oracle E is maximal for clustering – every uniformly recursively enumerable class is either not clusterable with any oracle or clusterable with oracle E.*

Proof. Assume that E satisfies $E \geq_T K$ and $E' \geq_T K''$ and assume that $S = \{A_0, A_1, \ldots\}$ is clusterable relative to some oracle. Then S satisfies Angluin's telltale condition below and one can actually give an algorithm which succeeds with the oracle E.

Angluin's Condition [3]. The class S is clusterable with the help of some oracle iff for every $I \in \mathrm{disj}(S)$, there is a finite set D, called a *telltale set for I*, such that $D \subseteq A_I$ and no $J \in \mathrm{disj}(S)$ satisfies $D \subseteq A_J \subset A_I$.

Note that one can test with oracle K'' whether the telltale condition holds for given I, D: $F(D, I) = 1 \Leftrightarrow (\forall J \in \mathrm{disj}(S))\, [D \not\subseteq A_J \vee A_J \not\subset A_I]$. This condition has an E-recursive approximation $F_s(D, I)$ which converges for $s \to \infty$ to 1 if $F(D, I) = 1$ holds and to 0 otherwise.

Algorithm M. Given an E-recursive enumeration of $\{(D, J) : D$ is a finite subset of $\mathbb{N}$ and $J \in \mathrm{disj}(S)\}$, $M(\sigma)$ outputs J from the first pair (D, J) satisfying the following conditions:

- $D \subseteq \mathrm{range}(\sigma) \subseteq A_J$;
- $F_{|\sigma|}(D, J) = 1$.

In order to guarantee that M is total, the search is limited to the first $|\sigma|$ pairs and $M(\sigma)$ outputs $\emptyset$ if none of the first $|\sigma|$ pairs qualifies.

Verification. Since every clustering task $I \in \mathrm{disj}(S)$ has a telltale set D' such that $D' \subseteq A_I$ and $F(D', I) = 1$, the algorithm converges to some pair (D, J) with $F(D, J) = 1$. One has that $D \subseteq A_I \subseteq A_J$ and it then follows from Angluin's condition that $A_I = A_J$.

Complete Class. It remains to show that Condition (a) on E is necessary. The class S_{comp} considered here consists of the sets $A_{i,D}$ defined below where $i \in \mathbb{N}$ and D is a finite subset of $\mathbb{N}$. Note that below the entry for $\emptyset$ is given explicitly and therefore $D \neq \emptyset$ in the second entry; in particular $\max(D)$ is defined there.

$$\begin{aligned} A_{i,\emptyset} &= \{\langle i, x\rangle : x \in W_i \cup \{0\}\}; \\ A_{i,D} &= \{\langle i, x\rangle : x \in D \cup \{0\} \\ &\qquad \vee\ (x > \max(D) \wedge \{z : \max(D) \leq z < x\} \subseteq W_i) \\ &\qquad \vee\ (x \leq \max(D) \wedge \{z : x \leq z \leq \max(D)\} \subseteq W_i)\}. \end{aligned}$$

Clusterer N with oracle K$''$. Given input σ, $N^{K''}$ determines the sets

$$B_i = \mathrm{range}(\sigma) \cap \{\langle i, 0\rangle, \langle i, 1\rangle, \ldots\}.$$

Then J consists of the pairs (i, D) where $B_i \neq \emptyset$ and D is a finite set of least norm satisfying one of the following conditions:

1. $A_{i,D} = B_i$;
2. $D = \emptyset$, $B_i \subseteq A_{i,\emptyset}$ and W_i coinfinite;
3. $A_{i,D} = B_i \cup \{\langle i, x\rangle : x \geq \max(D)\}$.

Then $N^{K''}$ outputs J.

Verification. Given a task I, the clusterer obviously finds all i where $(i, C) \in I$ for some C. Furthermore, there is at most one C with $(i, C) \in I$ since $A_{i,C}$ always contains $\langle i, 0\rangle$. It is easy to see that all three cases are disjoint and that $N^{K''}$ converges syntactically whenever $N^{K''}$ converges semantically.

- If $A_{i,C}$ is finite then eventually all elements show up and $N^{K''}$ outputs an index for this set.
- If $A_{i,C}$ is infinite and W_i coinfinite then $C = \emptyset$ and no finite subset of $A_{i,\emptyset}$ is in S. Thus the first case does not apply and $N^{K''}$ puts $(i, \emptyset)$ into J according to the second case.
- If $A_{i,C}$ is infinite and W_i cofinite, then let a_i be the first number such that all $x \geq a_i$ are in W_i. In particular, the set $D = \{x \leq a_i : \langle i, x\rangle \in A_{i,C}\}$ satisfies $A_{i,D} = A_{i,C}$ and therefore (i, D) or some equivalent index goes into J whenever B_i contains all $\langle i, x\rangle \in A_{i,C}$ with $x \leq a_i$.

This completes the verification of the clusterer $N^{K''}$. It is easy to see that S is also learnable relative to K''. Furthermore, S is clusterable and learnable relative to any oracle which is maximal for clustering.

Hardness of S. It is sufficient to show that learning is hard since no member of S is the disjoint finite union of more than two other members of S and every clusterer therefore has to find for every $L \in S$ a singleton $\{(i, D)\}$ such that $A_{i,D} = L$. In the following assume that an oracle E and a learner O^E using the oracle E are given. Note that every set $A_{i,\emptyset}$ has a stabilizing sequence. Let σ_i be the first stabilizing sequence found by a search applying the oracle E'. Note that $O^E(\sigma_i)$ has to be an index for $A_{i,\emptyset}$ since $O^E(\sigma_i\tau) = O^E(\sigma_i)$ for all $\tau \in (A_{i,\emptyset})^*$ by the definition of a stabilizing sequence. Let b_i be the maximum of all y with $\langle i, y\rangle \in \mathrm{range}(\sigma_i)$.

There is an index i such that $W_i = \mathbb{N} \oplus K$. Then $A_{i,\emptyset} = \{\langle i, y\rangle : y \in W_i\}$. Now consider any x with $2x > b_i$ and $A_{i,D}$ where D consists of $2x$ and all y with $\langle i, y\rangle \in \mathrm{range}(\sigma_i)$. If $x \in K$ then $\mathrm{range}(\sigma_i) \cup \{\langle i, 2x+1\rangle\} \subseteq A_{i,\emptyset}$. If $x \notin K$ then $A_{i,D} - A_{i,\emptyset} = \{\langle i, 2x+1\rangle\}$. Therefore,

$$x \in K \Leftrightarrow 2x + 1 \in W_i;$$
$$x \notin K \Leftrightarrow (\exists \tau \in (A_{i,\emptyset})^*)\,[O^E(\sigma_i\langle i, 2x+1\rangle\tau) \neq O^E(\sigma_i)].$$

A finite modification of the above formula takes care of the x with $2x \leq b_i$ and shows that K is computable relative to E.

Assume that the set W_i is cofinite and a_i is as above the minimum of all

x with $\{x, x+1, \ldots\} \subseteq W_i$. Now consider any $y < a_i$ with $y \in W_i$. Then $A_{i,\emptyset} - \{\langle i, y \rangle\}$ is in S_{comp}. Since σ_i is a stabilizing sequence for $A_{i,\emptyset}$, $\langle i, y \rangle$ occurs in σ_i. Thus there are no elements of W_i strictly between b_i and a_i. In particular, W_i is cofinite iff every $x > b_i$ with $x \in W_i$ actually satisfies $\{x, x+1, \ldots\} \subseteq W_i$. As it is already known that $K \leq_T E$, one has that $K' \leq E'$ and the following algorithm decides relative to E' whether W_i is cofinite.

Given i, compute relative to E' the sequence σ_i and the number b_i. Check whether there is an $x > b_i$ with $x \in W_i$. If not, then W_i is finite and thus coinfinite. If so, one can find such an x with oracle E. Then W_i is cofinite iff $\{x, x+1, \ldots\} \subseteq W_i$, which can again be checked with oracle E'.

So exploiting that $E \geq_T K$ and $E' \geq_T K'$, one can derive that $E' \geq_T K''$ since K'' and the index-set $\{i : W_i \text{ is cofinite}\}$ have the same Turing degree. This completes the proof. ∎

An oracle G is k-generic if for every Σ^0_k-set T of strings there is a prefix $\eta \preceq G$ such that either $\eta \in T$ or $\eta' \notin T$ for all $\eta' \succeq \eta$. There are 1-generic sets but no 2-generic sets below K. Nevertheless, k-generic sets exist for all $k \in \{1, 2, \ldots\}$.

Proposition 4.4. *Let E be a nonrecursive oracle with $E \leq_T K$.*

(a) *If E has 1-generic degree then E is trivial and permits only to cluster classes which can already be clustered without any oracle.*
(b) *If E does not have 1-generic degree then there is a uniformly recursively enumerable class which can be clustered using the oracle E but not without any oracle.*

The same characterizations hold for learning in place of clustering.

Proof (a). Clustering S and learning $\tilde{S} = \{A_I : I \in \text{disj}(S)\}$ have the same difficulty if one does not require the use of the hypothesis space $\{A_0, A_1, \ldots\}$. Therefore, if one can cluster S with the help of oracle E, then one can also learn $\tilde{S}$ with the help of the same oracle. Kummer and Stephan [16, Theorem 10.5] showed that $\tilde{S}$ can be learned without any oracle. This learner can be interpreted as a clusterer which outputs only singleton classes and uses an acceptable numbering of all recursively enumerable sets as its hypothesis space. By Proposition 3.2 one can translate this learner into a clusterer using $A_0, A_1, \ldots$ as its hypothesis space.

(b). By [16, Theorem 10.5] there is a class S^E_{func} of graphs of recursive functions which can be learned relative to oracle E but not without any oracle. The main task it to build a uniformly recursively enumerable superclass which still can be learned with oracle E. Without loss of generality all functions f with a graph in S^E_{func} satisfy that $f(0) = 0$. Therefore $\langle 0, 0 \rangle$ is in all members of S^E_{func} and the superclass S to be constructed and S is clusterable iff S is learnable.

Since $E \leq_T K$ the oracle E has a recursive approximation $E_0, E_1, \ldots$ and besides M^E also the approximations M^{E_s} working with some E_s instead of E are defined and uniformly recursive.

The Class S. For every given $i, j, k \in \mathbb{N}$, let $A_{i,j,k}$ contain all pairs $\langle x, y\rangle$ which satisfy one of the conditions (1), (2), (3) below. The class S consists of all $A_{i,j,k}$ with $i, j, k \in \mathbb{N}$.

Condition 1. The pair $\langle x, y\rangle$ is just $\langle 0, 0\rangle$.

Condition 2. There is a number $s \geq \max(\{i, j, k, x\})$ such that the following statements hold:

- $\varphi_i(z)$ is defined for all $z \leq \max(\{j, x\})$, $\varphi_i(0) = 0$ and $\varphi_i(x) = y$;
- for all t with $k \leq t \leq s$, $M^{E_t}(\langle 0, \varphi_i(0)\rangle\langle 1, \varphi_i(1)\rangle \ldots \langle j, \varphi_i(j)\rangle) = \{i\}$;
- for $z = j, j+1, \ldots, \max(\{j, x\})$, $M^{E_s}(\langle 0, \varphi_i(0)\rangle\langle 1, \varphi_i(1)\rangle \ldots \langle z, \varphi_i(z)\rangle) = \{i\}$;
- either $j = 0$ or $M^{E_s}(\langle 0, \varphi_i(0)\rangle\langle 1, \varphi_i(1)\rangle \ldots \langle j-1, \varphi_i(j-1)\rangle) \neq \{i\}$.

Condition 3. This condition does not depend on $\langle x, y\rangle$ since it covers the case where the parameters do not permit to construct a desired set but might already have caused the enumeration of pairs different from $\langle 0, 0\rangle$:

- φ_i is defined on $0, 1, \ldots, j$;
- $M^{E_s}(\eta) \neq M^{E_k}(\eta)$ for some $s > k$ and $\eta \preceq \langle 0, \varphi_i(0)\rangle\langle 1, \varphi_i(1)\rangle \ldots \langle j, \varphi_i(j)\rangle$.

It is easy to see that this class is uniformly recursively enumerable. The intuition behind the conditions is the following. Condition 1 makes the set $A_{i,j,k}$ nonempty and enforces that S is clusterable relative to E iff S is learnable relative to E. Condition 2 tries to put information on the graph of φ_i into $A_{i,j,k}$ where j, k serve as additional information. Condition 3 takes care of the class when the choice of the parameters j, k is inadequate. Note that whenever M^E converges for a total function f to $\{i\}$ such that $\varphi_i = \mathrm{graph}(f)$, then there is a position j from which on M^E has converged to $\{i\}$. In particular, $A_{i,j,k} = \mathrm{graph}(f)$ where k is the least number such that $M^{E_s}(\eta) = M^E(\eta)$ for all $s \geq k$ and $\eta \preceq \langle 0, f(0)\rangle\langle 1, f(1)\rangle \ldots \langle j, f(j)\rangle$. Let (i_0, j_0, k_0) be an index of $\{\langle 0, 0\rangle\}$ and (i_1, j_1, k_1) of $\{\langle x, y\rangle : x, y \in \mathbb{N}\}$.

Algorithm N with Oracle E. On input σ, N^E does the following steps:

1. let $f(m)$ be the least number y such that $\langle m, y\rangle$ is in $\mathrm{range}(\sigma)$;
2. if $f(0)$ or $f(1)$ cannot be recovered from $\mathrm{range}(\sigma)$ then output (i_0, j_0, k_0) and halt;
3. if there is $\langle m, z\rangle \in \mathrm{range}(\sigma)$ with $z > f(m)$ then output (i_1, j_1, k_1) and halt;
4. find largest n such that $f(0), f(1), \ldots, f(n)$ can be recovered from $\mathrm{range}(\sigma)$;
5. compute for $m = 0, 1, \ldots, n$ the indices i_m such that

$$M^E(\langle 0, f(0)\rangle\langle 1, f(1)\rangle \ldots \langle m, f(m)\rangle) = \{i_m\}$$

and let $k_{m,|\sigma|}$ be the least number o such that $\langle 0, 1\rangle$ is not enumerated into $A_{i_m,m,o}$ within $|\sigma| - o$ steps;

6. determine all numbers $m \in \{1, 2, \ldots, n\}$ such that either range(σ) consists of the elements enumerated into $A_{i_m, m, k_{m,|\sigma|}}$ within $|\sigma|$ steps or m is the least number with $i_m = i_{m+1} = \ldots = i_n$;
7. output $\{(i_m, m, k_{m,|\sigma|}\})$ for the least m that was selected in Step 6 and halt.

Verification. Let L be any set in S. It is easy to see that N^E identifies the sets A_{i_0,j_0,k_0} and A_{i_1,j_1,k_1}. Thus one can consider any set $L \in S$ which is of the form $\{\langle x, f(x)\rangle : x < b\}$ where $b \in \{2, 3, \ldots, \infty\}$ and f is a recursive function. It is obvious that any $\langle x, y\rangle \in$ range(σ) satisfies $f(x) = y$, thus f is correctly recovered by N^E and n is the largest integer such that all pairs $\langle x, f(x)\rangle$ with $x \leq n$ occur in σ.

In the following let j be the least number such that $j < b$ and there is a k with $A_{i_j,j,k} = L$. Now fix this k to be the minimal one with $A_{i_j,j,k} = L$. Then $A_{i_j,j,o} \neq L$ for all $o < k$; indeed $A_{i_j,j,o} = A_{i_1,j_1,k_1}$ for these o. Note that $k_{m,t}$ converges for $t \to \infty$ to k from below and that k is the first integer such that the $m, f(0), f(1), \ldots, f(m)$ chosen by the algorithm satisfy

$$\begin{aligned}(\forall m' \leq m)\,(\forall s \geq k)\ \ [\ \ & M^{E_s}(\langle 0, f(0)\rangle\langle 1, f(1)\rangle \ldots \langle m', f(m')\rangle) \\ & = M^E(\langle 0, f(0)\rangle\langle 1, f(1)\rangle \ldots \langle m', f(m')\rangle)\,].\end{aligned}$$

Given any text for L, assume that σ is so long that the following statements hold:

1. all pairs $\langle m, f(m)\rangle$ with $m \leq j$ have occurred in σ and thus N^E knows $f(0), f(1), \ldots, f(j)$;
2. if L is finite then $L =$ range(σ) and all elements of L are enumerated into $A_{i_j,j,k}$ within $|\sigma|$ steps;
3. for all $m \leq j$ and $t > |\sigma|$, $k_{m,t} = k_{m,|\sigma|}$;
4. an element of $A_{i_m,m,k_{m,|\sigma|}} - L$ is enumerated into $A_{i_m,m,k_{m,|\sigma|}}$ within $|\sigma|$ steps whenever this difference is not empty and $m \leq j$;
5. an element of $L - A_{i_m,m,k_{m,|\sigma|}}$ has occurred in σ whenever this difference is not empty and $m \leq j$.

The first statement implies that N^E can recover the relevant part of f. The second statement implies that whenever L is finite then its elements and the finitely many elements of $A_{i_j,j,k}$ are explicitly known to the learner. The third statement enforces that $k_{j,|\sigma|} = k$ and thus k is known to the learner. The fourth and fifth statement guarantee for all $m \leq j$ that N^E does not output $\{(i_m, m, k_{m,|\sigma|})\}$ whenever $A_{i_m,m,k_{m,|\sigma|}} \neq L$. By the choice of j, this holds for all $m \leq j$ and N^E outputs $\{(i_j, j, k)\}$ on input σ. Thus N^E identifies L. ∎

The following example shows that there is a difference between the trivial oracles for clustering in the general sense and clustering of uniformly recursively enumerable classes.

Example 4.5. *Every* 2*-generic oracle is trivial for clustering.*

Proof. Assume that G is 2-generic and $S = \{A_0, A_1, \ldots\}$ is clusterable relative to G via an oracle machine M^G. Without loss of generality, M^E is total for every oracle E. Now consider the following sets of strings.

$$\begin{aligned} T = \{\eta : (\exists I, J)\,(\exists x, t)\,(\exists \sigma \in A_I^*)\,(\forall \tau \in A_I^*)\,(\forall E \succeq \eta)\,(\forall s \geq t) \\ [(J \notin \mathrm{disj}_s(S) \vee A_{I,s}(x) \neq A_{J,s}(x)) \wedge I \in \mathrm{disj}_s(S) \wedge M^E(\sigma\tau) = J]\}; \\ U_{I,\sigma} = \{\vartheta : (\exists \tau \in A_I^*)\,(\forall E \succeq \vartheta)\,[M^E(\sigma\tau) \neq M^E(\sigma)]\}. \end{aligned}$$

The oracles quantified in the definitions above are only evaluated up to a certain point. Thus one can make the definitions of the sets to be Σ_2^0.

Given any $I \in \mathrm{disj}(S)$, M^G has a stabilizing sequence σ for A_I. If σ is not also a stabilizing sequence for M^E then there is a τ and a prefix $\vartheta \preceq E$ with $M^E(\sigma\tau) \neq M^E(\sigma)$ where all queries to E while computing these two values only target the domain of ϑ. Thus $\vartheta \in U_{I,\sigma}$. Since G is 2-generic and σ is a stabilizing sequence for M^G, there is a prefix $\eta \preceq G$ such that no extension $\vartheta \succeq \eta$ is in $U_{I,\sigma}$. In particular, σ is a stabilizing sequence for A_I and M^E whenever the oracle E satisfies $E \succeq \eta$. So the stabilizing sequence σ is uniform for all M^E with $E \succeq \eta$.

The set T contains now all η such that there is for some $I \in \mathrm{disj}(S)$ and a uniform stabilizing sequence for A_I with respect to η such that $M^E(\sigma)$ outputs for all $E \succeq \eta$ a J such that either $J \notin \mathrm{disj}(S)$ or $A_J \neq A_I$. It follows again that $\eta \not\preceq G$ for all $\eta \in T$. Thus there is a prefix $\theta \preceq G$ such that no extension of θ is in T.

Algorithm N. The clusterer N is a variant of the locking sequence hunting construction and searches simultaneously for an $\eta \succeq \theta$ and σ built from the data and a J such that $M^E(\sigma\tau) = J$ for all $E \succeq \eta$ and τ obtained from data seen so far. That is, if at stage s the set D is the range of all data seen so far, N searches the first pair (σ, η) in an enumeration of $\mathbb{N}^* \times \{0,1\}^*$ such that

- $\sigma \in D^*$ and $\eta \in \theta \cdot \{0,1\}^*$;
- for all $\tau \in D^*$ with $|\tau| \leq s - |\sigma|$ and all $E, F \succeq \eta$, $M^E(\sigma\tau) = M^F(\sigma)$.

Let $J = M^F(\sigma)$ for the set $F = \{x : \eta(x) \downarrow = 1\}$ N outputs J.

Verification. First one has to note that the search always terminates. The reason is that if σ is longer than s then the second search condition becomes void. Furthermore, there is a pair (σ, η) which is a uniform stabilizing sequence for A_I satisfying $\eta \succeq \theta$ and N finds such a sequence in the limit. Since G strongly avoids T in the sense that no extension of the prefix θ is in T, any pair (σ, η) considered by N infinitely often is a correct uniform stabilizing sequence and thus N converges to an index $J \in \mathrm{disj}(S)$ of A_I. ∎

5 The Finite Containment Property

The main topic of this section is to investigate the relationship between the topological structure of the class S and the question whether S is clusterable.

Recall that the classes S_{gold} and S_{sing} are not clusterable for topological reasons: they contain a cluster which is the disjoint infinite union of further clusters. So one might impose the following natural condition in order to overcome this problem.

Definition 5.1. A class $S = \{A_0, A_1, \ldots\}$ has the *Finite Containment Property* if every finite union of clusters contains only finitely many clusters. That is, for all i there are only finitely many sets $B \in S$ with $B \subseteq A_{\{0,1,\ldots,i\}}$.

Note that the Finite Containment Property is not necessary for clusterability. The class $\{\{i, i+1, \ldots\} : i \in \mathbb{N}\}$ is learnable and clusterable but does not satisfy the Finite Containment Property.

It is easy to see that the Finite Containment Property implies Angluin's condition: for every set A_I there are only finitely many sets A_J with $A_J \subset A_I$. If one takes D to be a set which contains for each A_J with $A_J \subset A_I$ the minimum of $A_I - A_J$, then D is finite and there is no A_J left with $D \subseteq A_J \subset A_I$.

Property 5.2. *If $S = \{A_0, A_1, \ldots\}$ has the Finite Containment Property then S is clusterable relative to every oracle E with $E \geq_T K$ and $E' \geq_T K''$.*

Although the Finite Containment Property guarantees clusterability from the topological point of view, it fails to guarantee clusterability from the recursion-theoretic point of view. Indeed the class S_{comp} given in the proof of Proposition 4.3 satisfies the Finite Containment Property. If W_i is cofinite then there are only finitely many subsets of $\{\langle i, 0\rangle, \langle i, 1\rangle, \ldots\}$ in S_{comp}. If W_i is coinfinite then $A_{i,\emptyset}$ is the only infinite subset of $\{\langle i, 0\rangle, \langle i, 1\rangle, \ldots\}$ in S_{comp} and all further subsets of $\{\langle i, 0\rangle, \langle i, 1\rangle, \ldots\}$ are finite sets which are not contained in $A_{i,\emptyset}$.

Note that the class S_C from Examples 3.5 is, for the case that C is nonrecursive, a witness for a class satsifying the Finite Containment Property which is learnable but not semiclusterable.

Property 5.3. *There is a class satisfying the Finite Containment Property which is learnable but neither clusterable nor semiclusterable.*

Since the topological structure of S_C is the same whenever C is infinite, clusterability of the class S_C is not determined by its topological structure.

Property 5.4. *Clusterability cannot be characterized in topological terms only.*

If one takes C to be the halting problem K then S_C witnesses that the oracle K is necessary for semiclustering, even in the case where classes have to satisfy the Finite Containment Property. Proposition 5.5 below shows that semiclustering is much easier than clustering: every uniformly recursively enumerable class is semiclusterable using the halting problem as an oracle. In particular, no topological condition can make semiclustering impossible, only computational conditions can.

Furthermore, every uniformly recursive class is semiclusterable. But this condition is not necessary, neither for semiclusterable nor for clusterable classes. For example the class $\{ \{x : \varphi_x(x) \downarrow = i\} : i \in \mathbb{N}\}$ is clusterable but consists of pairwise disjoint and recursively inseparable sets.

Proposition 5.5. *Every class has a semiclusterer using the halting problem as an oracle. Furthermore, a class $S = \{A_1, A_2, \ldots\}$ is semiclusterable without any oracle if the representation of the class is a uniformly recursive family, that is, if $\{(i, x) \in \mathbb{N}^2 : x \in A_i\}$ is recursive and not only recursively enumerable.*

Proof. It is sufficient to assume that M can check whether some x is in A_i. This can either be done by using the halting problem as an oracle or by assuming that the sequence $A_0, A_1, \ldots$ is uniformly recursive.

Now M on input σ determines all $J \subseteq \{0, 1, \ldots, |\sigma|\}$ such that $J \in \mathrm{disj}_{|\sigma|}(S)$ and $\mathrm{range}(\sigma) \subseteq A_J$. If there are several such sets, M outputs the one with the least norm. If there are none, M outputs $\emptyset$.

Note that all J which either do not represent disjoint sets or do not contain all data showing up in the limit are eventually disqualified. On the other hand, the set I representing the clustering task is among the determined sets whenever $|\sigma| \geq \max(I)$. So M converges in the limit to some J such that $\mathrm{norm}(J) \leq \mathrm{norm}(I)$, $J \in \mathrm{disj}(S)$ and $A_I \subseteq A_J$. Therefore M witnesses that S is semiclusterable. ∎

By Property 5.3 one can separate learnability from clusterability and semiclusterability by a class satisfying the Finite Containment Property. The next results show that there are no implications between the notions of learnability, clusterability and semiclusterability except the following two: "clusterable ⇒ semiclusterable" and "clusterable ⇒ learnable". All nonimplications are witnessed by classes satisfying the Finite Containment Property.

Proposition 5.6. *There is a class with the Finite Containment Property which is semiclusterable and learnable but not clusterable.*

Proof. Let S consist of the clusters

$$\begin{aligned} A_{3i} &= \{\langle i, x\rangle : x \in \mathbb{N}\}; \\ A_{3i+1} &= \{\langle i, x\rangle : x \text{ is even and } x < 2 + |W_i|\}; \\ A_{3i+2} &= \{\langle i, x\rangle : x \text{ is odd and } x < 2 + |W_i|\}. \end{aligned}$$

If W_i is infinite then A_{3i+1} consists of the $\langle i, x\rangle$ where x is even and W_{3i+2} consists of those $\langle i, x\rangle$ where x is odd. Since the union $A_0 \cup A_1 \cup \ldots \cup A_{3i+2}$ only contains the clusters $A_0, A_1, \ldots, A_{3i+2}$, the class S has the Finite Containment Property. Furthermore, S is semiclusterable by assigning to every input σ the set

$$\{3i : (\exists x \in \mathbb{N})\,[\langle i, x\rangle \in \mathrm{range}(\sigma)]\}.$$

Now it is shown that S is not clusterable. So assume by way of contradiction that a recursive M witnesses S to be clusterable.

For each A_{3i} one finds using the oracle K a stabilizing sequence $\sigma_i \in (A_{3i})^*$. One can reduce the question whether W_i is infinite to the question whether $\text{range}(\sigma_i) \subseteq A_{\{3i+1,3i+2\}}$, which is decidable relative to K: if W_i is infinite then $\text{range}(\sigma_i) \subseteq A_{\{3i+1,3i+2\}}$; if W_i is finite then $\text{range}(\sigma_i) \not\subseteq A_{\{3i+1,3i+2\}}$. The latter holds since otherwise σ_i would also be a stabilizing sequence for $A_{\{3i+1,3i+2\}}$ and M cannot have the same stabilizing sequence for two different sets with one being a subset of the other. The reduction of $\{i : W_i \text{ is infinite}\}$ to the oracle K contradicts the fact that $\{i : W_i \text{ is infinite}\}$ is Turing equivalent to K'.

It remains to show that the class S is learnable. This can be done by considering the following learner N.

Algorithm N. On input σ, let i be the least number such that a pair of the form $\langle i, x\rangle$ occurs in $\text{range}(\sigma)$. Then

$$N(\sigma) = \begin{cases} 3i & \text{if there are even and odd } y \text{ with } \langle i,y\rangle \in \text{range}(\sigma); \\ 3i+1 & \text{if there are only even } y \text{ with } \langle i,y\rangle \in \text{range}(\sigma); \\ 3i+2 & \text{if there are only odd } y \text{ with } \langle i,y\rangle \in \text{range}(\sigma). \end{cases}$$

Verification. N converges in the limit to $3i+j$ for the least i such that a pair $\langle i, x\rangle$ appears in the text and $j \in \{0,1,2\}$. It can easily be verified that the parameters i, j are chosen correctly whenever the text is for one of the sets A_{3i}, A_{3i+1} or A_{3i+2}. ∎

Example 5.7. *The class containing all sets $A_{3i}, A_{3i+1}, A_{3i+2}, A_{\{3i+1,3i+2\}}$ from the numbering $A_0, A_1, \ldots$ in the proof of Proposition 5.6 is neither learnable nor clusterable. But it satisfies the Finite Containment Property and is semiclusterable.*

A natural variant of the Finite Containment Property is the *Finite Meet Property* which says that each member of the class meets only finitely many other members. The class S_C witnesses that for $C = K$ one might need the oracle K to cluster a class satisfying the Finite Meet Property. Since the class given in the proof of Proposition 4.3 satisfies the Finite Containment Property and can be clustered only relative to maximal oracles, the next result shows that classes satisfying the Finite Meet Property are easier and require only the oracle K.

Proposition 5.8. *If a class satisfies the Finite Meet Property then it is clusterable with the halting-problem oracle K.*

Proof. Let $S = \{A_0, A_1, \ldots\}$ satisfy the Finite Meet Property. Let $\{b_0, b_1, \ldots\}$ be a text for A_I with $I \in \text{disj}(S)$; without loss of generality I consists of minimal indices, that is, for all $i \in I$ and for all j, if $A_i = A_j$ then $i \leq j$. Relative to K and the text, one can enumerate the set

$$H = \{h : A_h \cap \{b_0, b_1, \ldots\} \neq \emptyset \wedge (\forall j < h)\,[A_j \neq A_h]\,\}.$$

Now one considers all subsets $J \subseteq H$ with $J \in \text{disj}(S)$. Note that $I \subseteq H$ and $I \in \text{disj}(S)$, thus I is among the considered sets. Due to the Finite Meet Property, H is finite and only finitely many J are considered. Since these sets are uniformly recursive relative to K, one can find in the limit a considered set J which satisfies $A_J = \{b_0, b_1, \ldots\}$, that is, $A_J = A_I$. Thus S is clusterable using the oracle K. ∎

6 Numbering-Based Properties

Every uniformly recursively enumerable class of pairwise disjoint sets is learnable: the learner just waits until it finds that some $x \in \text{range}(\sigma)$ is in $A_{i,|\sigma|}$ for some $i \leq |\sigma|$ and from then on outputs the index i. But for nonrecursive sets C, the class S_C witnesses that such a class is not clusterable. So one has to consider not only properties of the class but also properties of some of its numberings. A class $\{A_0, A_1, \ldots\}$ has the *Numbering-Based Finite Containment Property* if for every I there are only finitely many j with $A_j \subseteq A_I$.

Proposition 6.1. *A class of pairwise disjoint sets has the Numbering-Based Finite Containment Property iff it is clusterable.*

Proof. Let $S = \{A_0, A_1, \ldots\}$ be a class of pairwise disjoint sets. Due to the Numbering-Based Finite Containment Property there are, for every i, only finitely many j with $A_j = A_i$. Now consider the following clusterer.

Algorithm M. On input σ, find the J of the largest norm which satisfies the following three conditions.

1. $J \subseteq \{0, 1, \ldots, |\sigma|\}$;
2. $J \in \text{disj}_{|\sigma|}(S)$;
3. $A_j \cap \text{range}(\sigma) \neq \emptyset$ for all $j \in J$.

Then output this J.

Verification. Note that the algorithm always terminates since $\emptyset$ satisfies the search conditions. Fix a clustering task I. The set $H = \{h : A_h \cap A_I \neq \emptyset\}$ is finite. Since M outputs always subsets of H, it follows that M converges to some $J \subseteq H$. This J is among the subsets of H the one with the largest norm which in addition also belongs to $\text{disj}(S)$. Since the members of S are pairwise disjoint, it holds for every $j \in J$ that A_j not only meets A_I but moreover is contained in A_I. Furthermore, if $i \in I$ then $A_i \cap A_J$ is not empty since otherwise $J \cup \{i\}$ is also a subset of H, is in $\text{disj}(S)$ and has a norm larger than the one of J. Thus $A_i \subseteq A_J$. Since this holds for all $i \in I$, $A_J = A_I$ and M is a clusterer for S.

Converse direction. Assume that N is a clusterer for S and consider the set

$$E = \{i : (\forall j < i)\,(\forall \sigma \in A_{j,i}^*, |\sigma| \leq i)\,(\exists \tau \in A_{\{i,j\}}^*)\,[N(\sigma\tau) \neq N(\sigma)]\}.$$

The set E is recursively enumerable since the universal quantifiers are bounded and the second one runs over strings of the finite set $A_{j,i}$ of all elements enumerated into A_j during i steps. Given any set in S, let i be its minimal index. Let $j < i$. Since $A_j \neq A_i$, A_j is disjoint from A_i, $\{j,i\} \in \mathrm{disj}(S)$ and $A_{\{i,j\}}$ is a proper superset of A_i. The clusterer N must converge on texts for A_j and $A_{\{i,j\}}$ to different outputs. Thus there is no $\sigma \in A_j^*$ with $N(\sigma) = N(\sigma\tau)$ for all $\tau \in A_{\{i,j\}}^*$. The index i is eventually enumerated into E. The set A_i has a stabilizing sequence σ. For all sufficiently large j with $A_i = A_j$, the length of σ is shorter than j and its range enumerated into A_i within j steps. It follows that σ prevents j from being enumerated into E and E contains only finitely many indices of A_i. The set E has a recursive enumeration $e_0, e_1, \ldots$ which defines by $B_h = A_{e_h}$ a new numbering $B_0, B_1, \ldots$ of S having the desired properties. ∎

Remark 6.2. Proposition 5.6 gives a class which satisfies the Numbering-Based Finite Containment Property but is not clusterable. A variant of the class given in the proof of Proposition 4.3 satisfies the Enumeration-Based Finite Containment Property but is clusterable only relative to maximal oracles.

Let the Numbering-Based Finite Meet Property denote that every A_i meets A_j only for finitely many j. It follows from Proposition 5.8 that a class satisfying the Numbering-Based Finite Meet Property is clusterable with the oracle K. But even this property is not sufficient for clustering without oracles. Example 5.7 satisfies actually the Numbering-Based Finite Meet Property but is not clusterable.

A further example of a class which satisfies the Numbering-Based Finite Meet Property but is not clusterable can be constructed using the following result of Jain and Sharma [12]: there is no learner which identifies all recursively enumerable sets from any text for the set plus an upper bound on its least index. The class

$$\{\{\langle i, x\rangle : x \in W_j\} : j \leq i \wedge W_j \neq \emptyset\}$$

has a numbering witnessing that it satisfies the Numbering-Based Finite Meet Property. But it consists of copies of sets W_j having coded an upper bound of an index of W_j into its first coordinate. This class cannot be learnable or clusterable because one would get a contradiction to the result of Jain and Sharma otherwise.

In the following two conditions are presented which are more restrictive than the Numbering-Based Finite Containment Property and guarantee that a class is clusterable.

Proposition 6.3. *Assume that $A_i \not\subseteq \cup_{j \neq i} A_j$ for all i and that it is decidable whether two sets A_i, A_j intersect. Then $S = \{A_0, A_1, \ldots\}$ is clusterable.*

Proof. The clusterer M uses the fact that one can check disjointness effectively, that is, that $\mathrm{disj}(S)$ is recursive.

Algorithm M. On input $b_0 b_1 \ldots b_s$, M considers all $J \subseteq \{0, 1 \ldots, s\}$ satisfying the following conditions:

1. $A_{i,s} \cap \{b_0, b_1, \ldots, b_s\} \neq \emptyset$ for all $i \in J$;
2. $J \in \mathrm{disj}(S)$;
3. there is no $j \in \{0, 1, \ldots, s\} - J$ such that $A_{j,s} \cap \{b_0, b_1, \ldots, b_s\} \neq \emptyset$ and $J \cup \{j\} \in \mathrm{disj}(S)$.

If there are several sets $J_1, J_2, \ldots, J_n \subseteq \{0, 1, \ldots, s\}$ which satisfy all three conditions then M computes for these sets the number

$$c_m = \max\{h \leq s + 1 : \{b_j : j < h\} \subseteq J_{m,s}\}$$

and outputs that set J_m for which c_m is maximal; if there are still several options, M outputs the one with the least norm.

Verification. Assume that a task $I \in \mathrm{disj}(S)$ is given and that $b_0 b_1 \ldots$ is a text for A_I. Let s be so large that there is a c satisfying the following conditions:

- $s \geq \max(I)$;
- for any $i \in I$ there exists a $h \leq c$ with $b_h \in A_i - \cup_{j \neq i} A_j$;
- $\{b_0, b_1, \ldots, b_c\} \subseteq A_I$.

Then I clearly satisfies the first two search conditions of M. The third is also satisfied since whenever $A_j \cap A_I = \emptyset$ then A_j does not contain any of the elements $b_0, b_1, \ldots, b_s$. Thus, any set $J \neq I$ satisfying all three conditions is not a superset of I. In particular there is an $i \in I - J$ and a $h \leq c$ such that $b_h \in A_i - A_J$. Since $\{b_0, b_1, \ldots, b_c\} \subseteq A_{I,s}$ and $\{b_0, b_1, \ldots, b_c\} \not\subseteq A_J$, M outputs I and not J. So M converges on a text for A_I to I and M is a clusterer for S. ▮

Proposition 6.4. *Assume that $S = \{A_0, A_1, \ldots\}$ satisfies the following three conditions:*

1. *every A_i is infinite;*
2. *if $i \neq j$ then $A_i \cap A_j$ is finite;*
3. *S is uniformly recursive, that is, $\{(i, x) : x \in A_i\}$ is recursive.*

Then S is clusterable. But no two of these three conditions are sufficient for being clusterable.

Proof. On input σ, the clusterer M searches for the J of the least norm satisfying the following properties:

- $J \subseteq \{0, 1, \ldots, |\sigma|\}$;
- $\mathrm{range}(\sigma) \subseteq A_J$;
- $J \in \mathrm{disj}_{|\sigma|}(S)$.

If such a J is found then M outputs J else M outputs $\emptyset$.

First, one can easily see that M is recursive since the search space is limited to $2^{|\sigma|+1}$ candidate sets. Second one considers any task I and any text for it. Every sufficiently long prefix σ of the text satisfies $i \leq |\sigma|$ and $A_i \cap \mathrm{range}(\sigma) \neq \emptyset$ for all $i \in I$. Thus I satisfies for all sufficiently long σ the three search conditions and M converges to a set J with $\mathrm{norm}(J) \leq \mathrm{norm}(I)$. For every $i \in I$, the set A_i is not a subset of $A_{J-\{i\}}$ since A_i is infinite and $A_i \cap A_{J-\{i\}}$ is finite. Thus $I \subseteq J$ and $I = J$ by $\mathrm{norm}(J) \leq \mathrm{norm}(I)$.

The class $\{A \times \mathbb{N} : A \in S_C\}$ for a nonrecursive parameter-set C satisfies Conditions 1 and 2 but is not clusterable. The class of all cofinite sets satisfies Conditions 1 and 3 but is neither learnable [11, Section 3.6.2] nor clusterable. The class S_{gold} satisfies Conditions 2 and 3 but is not clusterable. ∎

7 Geometric Examples

The major topic of the last two sections is to look at sets of clusters which are characterized by basic geometric properties. Therefore the underlying set is no longer $\mathbb{N}$ but the k-dimensional rational vector space $\mathbb{Q}^k$, where $k \in \{1, 2, \ldots\}$ is fixed. The classes considered consist of natural subsets of $\mathbb{Q}^k$. Except for the class $S_{\mathrm{accu},k}$ in Definition 7.2 below, the following holds: the clusters are built from finitely many parameter-points in $\mathbb{Q}^k$; the clusters are connected sets; every task consists of clusters having a positive distance from each other. So the way how to break down a task into clusters is natural and unique.

Recall that a subset $U \subseteq \mathbb{Q}^k$ is affine iff for every fixed $x \in U$ the set $V = \{y \in \mathbb{Q}^k : x + y \in U\}$ is a rational vector space, that is, closed under scalar multiplication and addition. The dimension of U is the dimension of V as a vector space, it is independent on the choice of x.

Example 7.1. *Let $S_{\mathit{aff},k}$ be the class of all affine subspaces of $\mathbb{Q}^k$ which have dimension $k-1$. The class $S_{\mathit{aff},k}$ is clusterable but the class $S_{\mathit{aff},k} \cup \{\mathbb{Q}^k\}$ is not.*

Proof. If one considers a one-one numbering $A_0, A_1, \ldots$ of $S_{\mathrm{aff},k}$, one can easily verify the following properties:

1. every A_i has dimension $k-1$;
2. if $i \neq j$ then $A_i \cap A_j$ is either empty or an affine subspace of dimension up to $k-2$;
3. the set $\{(i, x) : x \in A_i\}$ is recursive.

Conditions 1 and 2 enforce that $A_i \not\subseteq A_{J-\{i\}}$ for every finite set J. Thus one can adapt the clusterer for the class in Proposition 6.4 to a clusterer for $S_{\mathrm{aff},k}$. The verification can also easily be transferred.

The class $S_{\mathrm{aff},k} \cup \{\mathbb{Q}^k\}$ is just the geometric version of the class S_{sing} from Examples 3.5. Let V be a (k-1)-dimensional and W be a 1-dimensional vector space with $V + W = \mathbb{Q}^k$. Furthermore let $U_x = \{x + y : y \in U\}$. Since every U_x is in $S_{\mathrm{aff},k}$ and $\mathbb{Q}^k$ is the disjoint union of all U_x with $x \in W$ it follows that $S_{\mathrm{aff},k} \cup \{\mathbb{Q}^k\}$ is not clusterable. ∎

Definition 7.2. Let k be a positive natural number and $S_{\mathrm{accu},k}$ be a class $\{A_0, A_1, \ldots\}$ of subsets of $\mathbb{Q}^k$ such that

1. every A_i is bounded and has exactly one accumulation point a_i which is in $\mathbb{Q}^k$;
2. whenever q is an accumulation point of $\{a_0, a_1, \ldots\}$ then $q \notin \{a_0, a_1, \ldots\}$;
3. the sequence $a_0, a_1, \ldots$ is recursive and one-one.

Comment. Every set $A_i \cup \{a_i\}$ is compact, but it is not required that $a_i \in A_i$ and therefore the set A_i itself might fail to be compact.

Proposition 7.3. The class $S_{\text{accu},k}$ is clusterable.

Proof. The following machine M witnesses that $S_{\text{accu},k}$ is clusterable.

Algorithm M. On input $b_0 b_1 \ldots b_s$, let

$$\begin{aligned} H_s = \{i \leq s : (\exists h \leq s)\,(\forall j \leq h, j \neq i) \\ [b_h \notin \{b_0, b_1, \ldots, b_i\} \wedge d(a_i, b_h) < d(a_j, b_h)]\}; \\ J_s = \{i \in H_s : (\exists h)\,(\forall j \in H_s - \{i\})\,[b_h \in A_{i,s} - A_{j,s}]\}. \end{aligned}$$

and output J_s.

Verification. Since a_i is an accumulation point of A_i but not of $\{a_0, a_1, \ldots\}$ and since $a_i \neq a_j$ whenever $i \neq j$, there is for every i a threshold $\epsilon_i > 0$ such that

- for all $q \in \mathbb{Q}^k$ there is at most one i with $d(a_i, q) < \epsilon_i$;
- for almost all $q \in A_i$, $d(a_i, q) < \epsilon_i$.

Consider now a text $b_0 b_1 \ldots$ for a set A_I with $I \in \text{disj}(S_{\text{accu},k})$. Let $H = \cup_s H_s$ where H_s, J_s are the sets constructed by the algorithm with input $b_0 b_1 \ldots b_s$. There are only finitely many $q \in A_I$ which do not satisfy $d(a_i, q) < \epsilon_i$ for an $i \in I$ and there is a stage $t \geq \max(I)$ such that $\{b_0, b_1, \ldots, b_t\}$ contains all these q. It follows that $H \subseteq \{0, 1, \ldots, t\}$. Note that the intersection $A_i \cap A_j$ is finite for any different i, j since the sets A_i, A_j are bounded and have different accumulation points. So all sufficiently large s satisfy the following conditions:

- $H = H_s$;
- for all different $i, j \in H$, $A_i \cap A_j = A_{i,s} \cap A_{j,s}$;
- for all $i \in I$ there is a $h \leq s$ such that $b_h \in A_i - (\cup_{j \in H} A_j \cup \{b_0, b_1, \ldots, b_i\})$ and $d(b_h, a_i) < \epsilon_i$.

It follows that on one hand $I \subseteq H_s$ and on the other hand that every $j \in H_s - I$ satisfies $A_I \cap A_j \subseteq A_{I,s}$. Since $b_0 b_1 \ldots$ is a text for A_I, it follows that $J_s = I$ and M is a clusterer for $S_{\text{accu},k}$. ∎

The class $S_{\text{accu},k}$ is clusterable but the machine M makes use of the sequence $a_0, a_1, \ldots$ as an auxiliary source of information. Nevertheless, this information is implicit. One can build a program for it into the machine M which simulates this program in order to get some further information on $S_{\text{accu},k}$.

8 Clustering with Additional Information

Freivalds and Wiehagen [7] introduced a learning model where the learner receives in addition to the graph of the function to be learned an upper bound on the size of some program for this function – this additional information increases the learning power and enables to learn the class of all recursive functions.

Similarly, the machine M receiving adequate additional information can solve every clustering task for the class $S_{\mathrm{conv},k}$ defined below. But without that additional information, $S_{\mathrm{conv},k}$ is not clusterable. So the main goal of this section is to determine which pieces of additional information are sufficient to cluster certain geometrically defined classes where clustering without additional information is impossible.

Definition 8.1. For a given positive natural number k, the class $S_{\mathrm{conv},k}$ contains all subsets of $\mathbb{Q}^k$ which are the rational points in the convex hull of a finite subset of $\mathbb{Q}^k$.

Note that $S_{\mathrm{conv},k}$ has the following nice properties which will be used in the proofs implicitly: every cluster is ϵ-connected for all $\epsilon > 0$; any two clusters A_i, A_j have either a point in common or have a positive distance from each other where the distance is defined as $d(A_i, A_j) = \inf\{d(x, y) : x \in A_i, y \in A_j\}$. Furthermore, note that every set $A_i \in S_{\mathrm{conv},k}$ has a finite subset $D = \{x_0, x_1, \ldots, x_n\}$ such that

$$\begin{aligned} A_i = \{&q_0 x_0 + q_1 x_1 + \ldots + q_n x_n : q_0, q_1, \ldots, q_n \in \mathbb{Q} \wedge \\ &q_0, q_1, \ldots, q_n \geq 0 \wedge q_0 + q_1 + \ldots + q_n = 1\}. \end{aligned}$$

The set D is unique if one takes for n the smallest possible value. The set A_i generated from D as above is called the rational convex hull of D, written $\mathrm{hull}(D)$.

Proposition 8.2. *The class $S_{conv,k}$ is semiclusterable but not clusterable.*

Proof. A semiclusterer M for $S_{\mathrm{conv},k}$ works as follows.

Algorithm M. On input σ, M searches for the first $i \in \{0, 1, \ldots, |\sigma|\}$ such that $\mathrm{range}(\sigma) \subseteq \mathrm{hull}(A_{i,|\sigma|})$. If this i is found, then M outputs $\{i\}$ else M outputs $\emptyset$.

Verification. Let I be the clustering task and i be the least index of a set with $A_I \subseteq A_i$. Given any text for A_I, every sufficiently long prefix σ of the text satisfies the following three conditions.

- $|\sigma| \geq i$;
- $\mathrm{range}(\sigma) \not\subseteq A_j$ for all $j < i$;
- $\mathrm{hull}(A_{i,|\sigma|}) = A_i$.

It is easy to see that $M(\sigma) = \{i\}$ for the input σ. Therefore, M converges to i and $S_{\mathrm{conv},k}$ is semiclusterable.

Gold's Condition. Note that every singleton in $\mathbb{Q}^k$ belongs to $S_{\mathrm{conv},k}$ and that there are also infinite clusters. Then given an M and an infinite A_i, M has a stabilizing sequence $\sigma \in (A_i)^*$. So M either fails on the clustering task I representing all singletons $\{x\}$ with $x \in \mathrm{range}(\sigma)$ or M fails on the clustering task $\{i\}$ representing A_i. ∎

Proposition 8.3. *The class $S_{conv,k}$ is clusterable with additional information if for any task I one of the following pieces of information is also provided to the machine M:*

- *the number $|I|$ of clusters of the task;*
- *a positive lower bound ϵ for $\gamma = \min(\{1\} \cup \{d(A_i, A_j) : i, j \in I \wedge i \neq j\})$;*
- *the minimal number p of points which are needed to generate all the convex sets A_i with $i \in I$.*

Proof. The algorithm tries to identify in the limit the following pieces of information:

- finite sets $E_0, E_1, \ldots, E_m$;
- for each $l \in \{0, 1, \ldots, m\}$ an index j_l such that $A_{j_l} = \mathrm{hull}(E_l)$.

The final conjecture of the algorithm will then be the set $J = \{j_0, j_1, \ldots, j_m\}$.

The algorithm uses the notion of ϵ-component. Given $\epsilon > 0$, a subset $E \subseteq U$ is an ϵ-component of U if the following two conditions hold:

- for any $x, y \in E$ there is a sequence $z_0, z_1, \ldots, z_h$ of elements of E such that $x = z_1, y = z_h$ and $d(z_l, z_{l+1}) < \epsilon$ for all l with $1 \leq l < m$;
- $d(x, y) \geq \epsilon$ for any $x \in E$ and $y \in U - E$.

Note that for every ϵ and finite set U, the partition of U into ϵ-components is unique.

Algorithm M. On input σ, the clusterer goes into the first case applicable.

- If $|I|$ is given and there is a maximal $\epsilon \in \{\frac{1}{|\sigma|}, \frac{2}{|\sigma|}, \ldots\}$ such that $\mathrm{range}(\sigma)$ has exactly $|I|$ ϵ-components then let $m = |I|$ and $F_0, F_1, \ldots, F_{m-1}$ be these components. For $l = 0, 1, \ldots, m-1$, let E_l be the smallest subset of F_l with $\mathrm{hull}(E_l) = \mathrm{hull}(F_l)$.
- If ϵ is given then let m be the number of ϵ-components $F_0, F_1, \ldots, F_{m-1}$ of $\mathrm{range}(\sigma)$. For $l = 0, 1, \ldots, m-1$, let E_l be the smallest subset of F_l with $\mathrm{hull}(E_l) = \mathrm{hull}(F_l)$.
- If p is given and there are a number $m \in \{0, 1, \ldots, p-1\}$, an $\epsilon \in \{\frac{1}{|\sigma|}, \frac{2}{|\sigma|}, \ldots\}$ and $E_0, E_1, \ldots, E_{m-1} \subseteq \mathrm{range}(\sigma)$ such that
  - $p = |E_0| + |E_1| + \ldots + |E_{m-1}|$ and

– the sets $\mathrm{hull}(E_l) \cap \mathrm{range}(\sigma)$ are the ϵ-components of $\mathrm{range}(\sigma)$,

then fix m and the sets $E_0, E_1, \ldots, E_{m-1}$.

– If none of the previous cases hold then let $m = |\sigma|$ and $E_0, E_1, \ldots, E_{m-1}$ be the m singleton subsets of $\mathrm{range}(\sigma)$.

Now find for each $l \in \{0, 1, \ldots, m-1\}$ the least $s \geq |\sigma|$ such that there is a j_l with $\mathrm{hull}(E_l) = \mathrm{hull}(A_{j_l,s})$; if there are several candidates for this j_l, then choose the least one. Having found m and $j_0, j_1, \ldots, j_{m-1}$, the output is the set $J = \{j_0, j_1, \ldots, j_{m-1}\}$.

Verification. It is easy to verify that M is computable and is defined on every σ. Fix a task $I \in \mathrm{disj}(S_{\mathrm{conv},k})$ and a text for A_I. In case of additional information of the second type, let δ be the given lower bound ϵ for γ; otherwise let $\delta = 1$ if $|I| = 1$ and $\delta = \min\{d(A_i, A_j) : i, j \in I \wedge i \neq j\}$ if $|I| > 1$. Assume that a prefix σ of the given text is so long that for each $i \in I$, the following conditions hold:

– for all $j \in \{0, 1, \ldots, i\}$, $\mathrm{hull}(A_{j,|\sigma|}) = A_i$ iff $A_j = A_i$;
– $\mathrm{hull}(\mathrm{range}(\sigma) \cap A_{i,|\sigma|}) = A_i$;
– for all $x \in A_i$ there is a $y \in \mathrm{range}(\sigma)$ such that $d(x, y) < 0.1 \cdot \delta$;
– $\frac{1}{|\sigma|} < 0.1 \cdot \delta$.

Then one can verify that the algorithm will come up with a lower bound ϵ for δ such that the ϵ-components of $\mathrm{range}(\sigma)$ coincide with the δ-components. Furthermore, the parameter m is the cardinality $|I|$ and the sets $E_0, E_1, \ldots, E_{m-1}$ are sets of minimal cardinality such that

$$\{\mathrm{hull}(E_0), \mathrm{hull}(E_1), \ldots, \mathrm{hull}(E_{m-1})\} = \{A_i : i \in I\}.$$

Since σ is a sufficiently long prefix of the text, the output of the algorithm is a finite set J with $\{A_j : j \in J\} = \{A_i : i \in I\}$. It follows that M solves the clustering task I. ∎

The last results of the present work deal with conditions under which nonconvex geometrical objects can be clustered. The first approach is to look at unions of convex objects which are still connected. For $k = 1$, this class is the same as $S_{\mathrm{conv},1}$. But for $k = 2$, this class is larger. There the type of additional information used for clustering $S_{\mathrm{conv},k}$ is no longer sufficient. Given both the number of clusters and the number of vertices as additional information, it is possible to cluster the natural subclass $S_{\mathrm{polygon},2}$ of all classes considered. But if one permits holes inside the clusters, this additional information is no longer sufficient. An alternative parameter is the k-dimensional area covered by a geometric object. In Example 8.8 a natural class $S_{\mathrm{area},k}$ is introduced which can be clustered with the area of a clustering task given as additional information. The class $S_{\mathrm{area},2}$ contains $S_{\mathrm{polygon},2}$ and the class from Example 8.7 as subclasses.

Definition 8.4. A polygon p is given by n vertices $q_1, q_2, \ldots, q_n \in \mathbb{Q}^2$ and is the union of n sides which are the convex hulls of $\{q_1, q_2\}, \{q_2, q_3\}, \ldots, \{q_n, q_1\}$. The sides do not cross each other and exactly two sides contain one vertex. Every side has positive length and the angle between the two sides meeting at a vertex is never 0, 180 or 360 degrees. Let $p_0, p_1, \ldots$ be an enumeration of the polygons and let P_i be the set of all points in $\mathbb{Q}^2$ which are on the polygon p_i or in its interior. Let n_i denote the minimum number of vertices to define the i-th polygon p_i and $S_{\text{polygon},2}$ be the class $\{P_0, P_1, \ldots\}$.

Remark 8.5. Note that every polygon has the same number of sides as vertices. The length 0 of sides and the angle of 180 degrees are forbidden in order to make the representation unique up to some permutation of the vertices. The angles of 0 and 360 degrees are forbidden in order to avoid irregularities.

The following fact will be used below. Assume that $P_i \subseteq P_j$, $n_i \leq n_j$ and every side of p_j contains at least $n_j + 2$ points of P_i. Then $P_i = P_j$. To see this, consider any side T of p_j. Let $c_0, c_1, \ldots, c_{n_j}, c_{n_j+1}$ be $n_j + 2$ points on $T \cap P_i$. These points are all on p_i since they are on p_j and $P_i \subseteq P_j$. There is a $u \in \{0, 1, \ldots, n_j\}$ such that no vertex of p_i is properly between c_u and c_{u+1}. Then the convex hull of $\{c_u, c_{u+1}\}$ is part of a side U_T of P_i. So every side T of p_j has with some side U_T of p_i at least two points in common. The only way to have $P_i \neq P_j$ is to assume that there are two sides $T, \tilde{T}$ of P_j such that $U_T = U_{\tilde{T}}$. Let $d_1 \in T \cap U_T$ and $d_2 \in \tilde{T} \cap U_{\tilde{T}}$. Then U_T contains $\text{hull}(\{d_1, d_2\})$ and is a subset of P_j although not a side of P_j. Since U_T touches two sides of P_j and goes through the interior of P_j, U_T splits P_j into two halves each of them having some sides different from U_T. On these sides are points of $p_i \cap p_j$ and so P_i would also be split into two halves, a contradiction. Thus $U_T \neq U_{\tilde{T}}$; U_T and $U_{\tilde{T}}$ have at most one point in common for different sides $T, \tilde{T}$ of P_j. In particular, if $T, \tilde{T}$ are neighbouring then the vertex between them is also in $U_T \cap U_{\tilde{T}}$. It follows that $U_T = T$, $U_{\tilde{T}} = \tilde{T}$, $p_i = p_j$ and $n_i = n_j$.

Note that this property no longer holds if one permits a set of polygons instead of a single polygon. So there is a polygon p_j such that one can find, for any finite subset $F \subseteq P_j$, a set $\{i_1, i_2\} \in \text{disj}(S_{\text{polygon},2})$ with $F \subseteq P_{\{i_1,i_2\}} \subseteq P_j$. More precisely, let p_j be given by $(0,0), (0,1), (1,1), (1,2), (2,2), (2,1), (3,1), (3,0)$ and F be any finite subset of P_j. Then take $q = \min(\{y : (\exists x)\,[(x,y) \in F \wedge y > 1]\})$ and take i_1, i_2 representing the rectangles given by $(0,0), (0,1), (3,1), (3,0)$ and $(1,q), (1,2), (2,2), (2,q)$.

Figure 1 illustrates the last counterexample. More information on polygons can be found in textbooks on geometry like [15].

Proposition 8.6. *The class $S_{polygon,2} = \{P_0, P_1, \ldots\}$ is clusterable with additional information in the sense that it is clusterable from the following input provided to a clusterer for task I in addition to a text for P_I: the cardinality $|I|$ and the number $\sum_{i \in I} n_i$. Clustering is impossible if only one of these two pieces of information is available.*

Proof. Assume that the algorithm M knows $|I|$ and $\sum_{i \in I} n_i$ and receives as input a prefix σ of a text for A_I. Then M searches for the $J \subseteq \{0, 1, \ldots, |\sigma|\}$ of least norm which satisfies the following conditions:

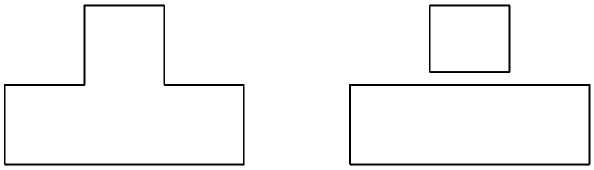

Fig. 1. Two clusters with 8 vertices and 8 sides.

1. $J \in \mathrm{disj}(S_{\mathrm{polygon},2})$;
2. $|J| = |I|$;
3. $\sum_{j \in J} n_j = \sum_{i \in I} n_i$;
4. $\mathrm{range}(\sigma) \subseteq P_J$;
5. the vertices of the p_j with $j \in J$ are in $\mathrm{range}(\sigma)$;
6. if T is a side of p_j and $j \in J$ then $|T \cap \mathrm{range}(\sigma)| \geq \sum_{i \in I} n_i + 2$.

M outputs J if J is found and $\emptyset$ otherwise.

For the verification, it is easy to see that M is recursive. Now consider any clustering task $I \in \mathrm{disj}(S_{\mathrm{polygon},2})$. Since I satisfies the search conditions for all sufficiently long prefixes σ of the text, the clusterer converges to a J with $\mathrm{norm}(J) \leq \mathrm{norm}(I)$, $P_I \subseteq P_J$, $|J| = |I|$ and $\sum_{j \in J} n_j = \sum_{i \in I} n_i$. If $i \in I$ then $P_i \subseteq P_J$. If $P_i \not\subseteq P_j$ for any single $j \in J$ then the P_j with $j \in J$ intersecting P_i would have a positive distance from each other; but since P_i is connected some points of P_i would not be in any P_j with $j \in J$. Thus this case cannot happen. Furthermore, if P_j is disjoint from P_I then the vertices of P_j do never show up in the input and thus $j \notin J$. It follows that there is a one-one correspondence between the $i \in I$ and $j \in J$ such that $P_i \subseteq P_j$. Since $\sum_{j \in J} n_j = \sum_{i \in I} n_i$ there are $i \in I$ and $j \in J$ with $P_i \subseteq P_j$ and $n_i \leq n_j$. Furthermore $P_j \cap P_{I-\{i\}} = \emptyset$, thus all points of P_j which have shown up in the input are actually from P_i. It follows for every side T of p_j that $n_j + 2 \leq |T \cap \mathrm{range}(\sigma)| \leq |T \cap P_i|$. Thus, by Remark 8.5, $P_i = P_j$ and $n_i = n_j$. In particular, there are no $i \in I$, $j \in J$ with $P_i \subseteq P_j$ and $n_i < n_j$. Since $|I| = |J|$ and $\sum_{j \in J} n_j = \sum_{i \in I} n_i$, one can conclude that there are also no $i \in I$, $j \in J$ with $P_i \subseteq P_j$ and $n_i > n_j$. So $n_i = n_j$ whenever $i \in I, j \in J, P_i \subseteq P_j$. This gives by the previous considerations that $P_i = P_j$ whenever $i \in I, j \in J, P_i \subseteq P_j$. In particular $P_J = P_I$ and M is a clusterer for $S_{\mathrm{polygon},2}$ which succeeds whenever it receives on the input, besides a text for P_I, also the numbers $|I|$ and $\sum_{i \in I} n_i$.

Now it is shown that besides the text also the other two pieces of information given to M are needed. That is, M cannot succeed while receiving only one of them.

If only the additional information $|I|$ is given, then consider a stabilizing sequence σ for the rectangle P_i with vertices $(0,0), (0,2), (1,2), (1,0)$. Since $\mathrm{range}(\sigma)$ is finite, there are rationals q_1, q_2 with $0 < q_1 < q_2 < 1$ such that no point of the form (q, r) with $q_1 < q < q_2$ is in $\mathrm{range}(\sigma)$. Thus σ is also a stabilizing sequence for the P_j given by the polygon through the vertices

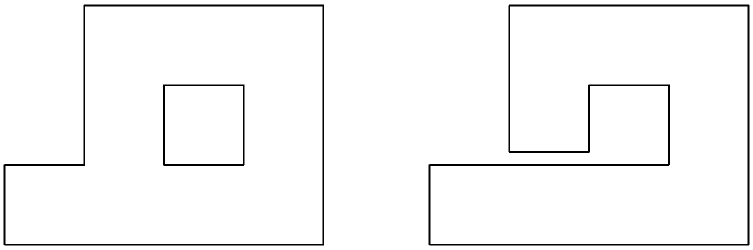

Fig. 2. Opening a hole while preserving 10 vertices and 10 sides.

$(0,0),(0,2),(q_1,2),(q_1,1),(q_2,1),(q_2,2),(1,2),(1,0)$ and $P_j \subset P_i$. So the clusterer fails to identify either the task $\{i\}$ or the task $\{j\}$.

If only the additional information $\sum_{i\in I} n_i$ is given, one can take $I = \{i\}$ such that p_i, P_i is given by $(0,0),(0,1),(1,1),(1,2),(2,2),(2,1),(3,1),(3,0)$ and $n_i = 8$ as done at the third paragraph of Remark 8.5. Now let $\sigma \in P_i^*$ be a stabilizing sequence for P_i and $q = \min(\{y : (\exists x)\,[(x,y) \in range(\sigma) \wedge y > 1]\})$. Then σ is also a stabilizing sequence for a cluster consisting of the two rectangles which are given as $(0,0),(0,1),(3,1),(3,0)$ and $(1,q),(1,2),(2,2),(2,q)$. See Figure 1 for an illustration. ∎

Example 8.7. *Let $B_{i,j} = p_j \cup (P_i - P_j)$ and $m_{i,j} = n_i + n_j$ if $P_j \subseteq P_i - p_i$; otherwise let $B_{i,j} = P_i$ and $m_{i,j} = n_i$. Let $S_{hole,2}$ consist of all sets $B_{i,j}$. Then it is impossible to cluster $S_{hole,2}$ if besides a text the only pieces of additional information supplied are $|I|$ and $\sum_{(i,j)\in I} m_{i,j}$.*

Proof. The counterexample here is an adaptation of the counterexample from Proposition 8.6. The idea is just to connect the two parts by a bridge and only to cut out the lower connection.

Now take i,j such that the polygons p_i, p_j are given by $(0,0),(0,1),(1,1),(1,3),(4,3),(4,0)$ and $(1,2),(2,2),(2,3),(1,3)$. Note that $m_{i,j} = 10$. Let $\sigma \in B_{i,j}^*$ be a stabilizing sequence for $B_{i,j}$ and

$$q = \min(\{y : (\exists x)\,[(x,y) \in range(\sigma) \wedge y > 1]\}).$$

Then σ is also a stabilizing sequence for a polygon $P_h \subseteq B_{i,j}$ given by $(0,0),(0,1),(1,3),(2,3),(2,2),(2,q),(1,q),(1,3),(4,3),(4,0)$. The polygon P_h has also 10 vertices and is obtained by connecting the hole with the outside world. Figure 2 illustrates this counterexample. ∎

Alternatively one might not restrict the dimension but require that the class under consideration is the union of convex hulls of finite sets which have a positive k-dimensional area. Then this area is a natural parameter for clustering with additional information.

Example 8.8. *Let $S_{area,k} = \{A_0, A_1, \ldots\}$ be the class of finite unions of members of $S_{conv,k}$ which are connected and have a positive k-dimensional area. Without loss of generality the set $\{(i,x) : x \in A_i\}$ and the function mapping i to the area of A_i are recursive. Then there is a clusterer for $S_{area,k}$ which uses the area of the members of a cluster as additional information. But $S_{area,k}$ cannot be clustered without additional information.*

Proof. Assume that $A_I \subset A_J$ and let $x \in A_J - A_I$ be given. The point x has a positive distance r from A_I. But the area of $A_J \cap R$ where R is the k-dimensional cube of side-length $0.1 \cdot r/k$ with center x is positive. It follows that the area of A_J is at least the sum of the areas of A_I and $R \cap A_J$. So whenever two sets A_I, A_J have the same area, they are either equal or incomparable. Thus one can use the following algorithm.

For any given task I, M receives the additional information q and a prefix σ of a text for A_I. Then M outputs the first J such that $J \in \mathrm{disj}_{|\sigma|}(S_{\mathrm{area},k})$, $\mathrm{range}(\sigma) \subseteq A_J$ and A_J has the k-dimensional area q.

It is easy to see that M is recursive and total. Furthermore, M converges to the least J with $\mathrm{norm}(J) = \mathrm{norm}(I)$, A_J having the area q and $A_I \subseteq A_J$. It follows from the arguments above that $A_I = A_J$ and M satisfies the required properties. ∎

9 Conclusion

Clustering is a process which makes important use of prior assumptions. Indeed, not every set of points in an underlying space is a potential cluster; geometric conditions for instance play an important role in the definition of the class of admissible clusters. Whereas such conditions have been taken into account in previous studies, none of those has investigated the consequences of the more fundamental requirement that clustering is a computable process. This paper shows that recursion-theoretic and geometric conditions can both yield substantial insights on whether or not clustering is possible. It also explores to which extent clustering depends on computational properties, by characterizing the power of oracles for clustering. It is expected that further studies of the interaction between topological, recursion-theoretic and geometrical properties will turn out to be fruitful.

References

1. Leonard Adleman and Manuel Blum. Inductive inference and unsolvability. *Journal of Symbolic Logic*, 56:891–900, 1991.
2. Michael R. Anderberg. *Cluster Analysis for Applications.* Academic Press, 1973.
3. Dana Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135, 1980.
4. Dick de Jongh and Makoto Kanazawa. Angluin's theorem for indexed families of r.e. sets and applications. In *Proceedings 9th Annual Confeference on Computational Learning Theory*, pages 193–204. ACM Press, New York, NY, 1996.

5. Richard Duda, Peter Hart and David Stork. *Pattern Classification.* Wiley, second edition, 2001.
6. Lance Fortnow, William Gasarch, Sanjay Jain, Efim Kinber, Martin Kummer, Stuart A. Kurtz, Mark Pleszkoch, Theodore A. Slaman, Robert Solovay and Frank Stephan. Extremes in the degrees of inferability. *Annals of Pure and Applied Logic*, 66:231–276, 1994.
7. Rūsiņš Freivalds and Rolf Wiehagen. Inductive inference with additional information. *Elektronische Informationsverarbeitung und Kybernetik* 15:179–185, 1979.
8. Mark Fulk. Prudence and other conditions on formal language learning. *Information and Computation*, 85:1–11, 1990.
9. E. Mark Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
10. Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data.* Prentice-Hall, 1988.
11. Sanjay Jain, Daniel Osherson, James Royer and Arun Sharma. *Systems that Learn: An Introduction to Learning Theory.* MIT Press, Cambridge, Mass., second edition, 1999.
12. Sanjay Jain and Arun Sharma. Learning with the knowledge of an upper bound on program size. *Information and Computation*, 102:118–166, 1993.
13. Sanjay Jain and Arun Sharma. On the non-existence of maximal inference degrees for language identification. *Information Processing Letters*, 47:81–88, 1993.
14. Jon Kleinberg. An impossibility theorem for clustering. *Advances in Neural Information Processing Systems 15* (NIPS 2002), MIT Press, Cambridge, 446–453, 2003.
15. Friedhelm Kürpig and Oliver Niewiadomski. *Grundlehre Geometrie. Begriffe, Lehrsätze, Grundkonstruktionen.* Vieweg, Braunschweig, 1992.
16. Martin Kummer and Frank Stephan. On the structure of the degrees of inferability. *Journal of Computer and System Sciences*, (Special Issue COLT 1993), 52:214–238, 1996.
17. Pitu B. Mirchandani and Richard L. Francis (editors). *Discrete Location Theory.* Wiley, 1990.
18. Piergiorgio Odifreddi. *Classical Recursion Theory.* North-Holland, Amsterdam, 1989.
19. Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition.* Academic Press, 1998.