The Computational Complexity of Concise Hypersphere Classification

Eduard Eiben [1] Robert Ganian [2] Iyad Kanj [3] Sebastian Ordyniak [4] Stefan Szeider [2]

Abstract

Hypersphere classification is a classical and foundational method that can provide easy-to-process explanations for the classification of real-valued and binary data. However, obtaining an (ideally concise) explanation via hypersphere classification is much more difficult when dealing with binary data than real-valued data. In this paper, we perform the first complexity-theoretic study of the hypersphere classification problem for binary data. We use the fine-grained parameterized complexity paradigm to analyze the impact of structural properties that may be present in the input data as well as potential conciseness constraints. Our results include stronger lower bounds and new fixed-parameter algorithms for hypersphere classification of binary data, which can find an exact and concise explanation when one exists.

1. Introduction

With the rapid advancement of Machine Learning (ML) models to automate decisions, there has been increasing interest in explainable Artificial Intelligence (XAI), where the ML models can explain their decisions in a way humans understand. This has led to the reexamination of ML models that are implicitly easy to explain and interpret with a particular focus on the conciseness of explanations (Doshi-Velez & Kim, 2017; Lipton, 2018; Monroe, 2018; Ribeiro et al., 2018; Shih et al., 2018; Barceló et al., 2020; Chalasani et al., 2020; Darwiche & Hirth, 2020; Blanc et al., 2021; Ignatiev et al., 2021; Wäldchen et al., 2021; Izza et al., 2022).

In this article, we consider a simple classification task—one of the cornerstones of machine learning—from the viewpoint of XAI. However, unlike previous works on explainability, which have typically targeted questions such as identifying a suitable interpretable model for (area-specific) classification (Nori et al., 2021; Shih et al., 2021; Wang et al., 2021) or measuring the accuracy cost of explainability (Laber & Murtinho, 2021; Makarychev & Shan, 2021), the goal of this work is to obtain a comprehensive understanding of the computational complexity of performing binary classification via one of the most fundamental interpretable models.

Consider a set M of either real-valued or binary training feature points, each represented as a d-dimensional feature vector over $[0, 1]$ or $\{0, 1\}$ and labeled as either "blue" (the set V_{B}) or "red" (the set V_{R}). There is, by now, a broad set of more or less opaque classifiers capable of using such a training set to classify unlabeled data, where the suitability of each method depends on the data domain and context; moreover, some classifiers are tailored to real-valued data, while others are designed for binary (or, more generally, categorical) data. In this paper, we consider one of the two arguably simplest—and hence easiest to explain and visualize—types of classifiers, which can be used in both data settings: a hypersphere. More formally, the explanations we consider consist of a cluster center $\vec{c}$ (an element of $[0, 1]^d$ or $\{0, 1\}^d$) and distance ℓ such that each feature vector is at a distance at most ℓ from $\vec{c}$ if and only if it is blue.

The reason for studying the complexity of hypersphere classification does not stem purely from the problem's connection to explainability. Together with classification by a separating hyperplane, hypersphere classification represents one of the most classic explanatory examples of classifiers (see (Cooper, 1962; Wang et al., 2007; 2005) to name a few) which have been extensively studied from both the computational geometry and the machine learning perspectives (Astorino et al., 2016; Astorino & Gaudioso, 2009; Cooper, 1962; Wang et al., 2007; 2005; O'Rourke et al., 1986; Agarwal et al., 2006; Hurtado et al., 2003). Moreover, hypersphere classification is of special importance in one-class classification due to the inherent asymmetry of the provided explanations (Kim et al., 2021). While hyperplane separation can be encoded as a linear program and hence is easily polynomial-time solvable for real-valued and binary data, the computational complexity of hypersphere

[*]Equal contribution [1]Royal Holloway, University of London, UK. [2]Algorithms and Complexity Group, TU Wien, Austria. [3]DePaul University, USA. [4]University of Leeds, UK.. Correspondence to: Eduard Eiben <eduard.eiben@rhul.ac.uk>, Robert Ganian <rganian@gmail.com>, Iyad Kanj <ikanj@depaul.edu>, Sebastian Ordyniak <sordyniak@gmail.com>, Stefan Szeider <sz@ac.tuwien.ac.at>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

classification is far less obvious and has so far remained surprisingly unexplored.

This apparent gap contrasts the situation for several other computational problems arising in the area of machine learning, which have already been targeted by detailed complexity-theoretic studies (Ordyniak & Szeider, 2013; Ganian et al., 2018; Simonov et al., 2019; Dahiya et al., 2021; Ganian & Korchemna, 2021; Ganian et al., 2022; Grüttemeier & Komusiewicz, 2022), carried out using the classical as well as the parameterized-complexity paradigms. In this article, we close the gap by laying bare a detailed map of the problem's computational complexity via the design of novel theoretical algorithms as well as accompanying computational lower bounds.

Contributions. We begin by observing that the hypersphere classification problem is polynomial-time solvable when the input data is real-valued; in particular, this case can be handled via a more sophisticated linear programming encoding than the one used for the classical hyperplane separation problem. However, this approach completely fails when dealing with binary data, warranting a more careful complexity-theoretic study of this case. Our first result shows that hypersphere classification of binary data is not only NP-hard in general but remains NP-hard even when there are only two red vectors. We also obtain an analogous hardness result for instances with two blue vectors.

The fact that the problem's complexity differs between real-valued and categorical data is already interesting. However, the NP-hardness of the latter case does not preclude the existence of efficient algorithms that can solve the problem under additional natural restrictions. Indeed, one of the central themes in modern complexity-theoretic research is the identification of the exact boundaries of tractability. This is frequently achieved through the lens of the *parameterized complexity* paradigm (Downey & Fellows, 2013; Cygan et al., 2015), where we associate each problem instance $\mathcal{I}$ with an integer parameter k (often capturing a certain structural property of the instance) and ask whether the problem of interest can be solved by a "fixed-parameter" algorithm, that is, by an algorithm with runtime of the form $f(k) \cdot |\mathcal{I}|^{\mathcal{O}(1)}$ for some computable function f. This gives rise to a strong form of computational tractability called *fixed-parameter tractability* (FPT).

In the case of our problem of interest, it is easy to observe that hypersphere classification of binary data is fixed-parameter tractable when parameterized by the data dimension d (since the number of possible centers is upper-bounded by 2^d). Moreover, we show that the problem also admits a fixed-parameter algorithm when parameterized by the total number of feature points via a combinatorial reduction to a known tractable fragment of Integer Linear Programming. While these are important pieces of the complexity-theoretic landscape of hypersphere classification, these two initial results are somewhat unsatisfying on their own because (1) they rely on highly restrictive parameterizations, and (2) they ignore a central aspect of explainability, which is *conciseness* or *succinctness* (Ribeiro et al., 2018; Shih et al., 2018; Blanc et al., 2021; Wäldchen et al., 2021; Chalasani et al., 2020; Izza et al., 2022; Ordyniak et al., 2023).

A natural measure of conciseness in our setting is the number of "1" coordinates in a vector; indeed, any explanation produced by a classifier will likely end up ignored by users if such an explanation is incomprehensibly long, relying on too many features. At the same time, depending on the source of the input data, we may often deal with feature vectors that are already concise. Having concise feature vectors does not necessarily guarantee the existence of a concise center (and vice-versa, concise centers may exist for non-concise data); however, at least one of the two independent measures of conciseness can be expected (or even required) to be small in a variety of settings, making them natural choices for parameters in our analysis. In the second part of the article, we show that these two conciseness parameters—a bound econ on the conciseness of the sought-after explanation and a bound dcon on the conciseness of all feature vectors in the training data—can be algorithmically exploited to cope with the NP-hardness of the hypersphere classification problem for binary data.

Toward understanding the complexity of hypersphere classification of binary data through the perspective of conciseness constraints, we begin by considering restrictions on the data conciseness dcon. We obtain a tight classification by showing that the problem is polynomial-time tractable when $\mathsf{dcon} \leq 3$ via a reduction to a tractable fragment of the constraint satisfaction problem and NP-hard otherwise. Moreover, we obtain fixed-parameter algorithms parameterized by dcon plus the number of red or blue points, circumventing the earlier NP-hardness results. When considering the explanation conciseness, we show that hypersphere classification is XP-tractable when parameterized by econ and at the same time provide evidence excluding fixed-parameter tractability even parameterized by econ together with the number of red or blue points. Finally, we obtain a linear-time fixed-parameter algorithm for the problem parameterized by $\mathsf{econ} + \mathsf{dcon}$.

While this settles the complexity of binary-data hypersphere classification from the perspective of conciseness measures, the obtained lower bounds imply that neither measure of conciseness (i.e., neither econ nor dcon) suffices to achieve fixed-parameter tractability on its own. As our final contribution, we consider whether achieving tractability for the problem is possible by exploiting a suitable structural measure of the input data. In particular, following recent

	Conciseness			
Structure	$\emptyset$	econ	dcon	econ + dcon
$\emptyset$	NP-h (Thm 3)	XP (Obs 10), W[2]-h (Thm 11)	NP-h$_{\geq 4}$ (Thm 7)	FPT (Thm 13)
$\|V_R\|$	NP-h$_{\geq 2}$ (Thm 3)	XP (Obs 10), W[2]-h (Thm 11)	FPT (Thm 9)	FPT (Thm 9)
$\|V_B\|$	NP-h$_{\geq 2}$ (Thm 3)	XP (Obs 10), W[1]-h (Thm 12)	FPT (Thm 9)	FPT (Thm 9)
$\|V_R \cup V_B\|$	FPT (Thm 5)	FPT (Thm 5)	FPT (Thm 5)	FPT (Thm 5)
d	FPT (trivial)	FPT (trivial)	FPT (trivial)	FPT (trivial)
tw	XP (Cor 15)	FPT (Cor 16)	FPT (Cor 17)	FPT (Cor 16)

Table 1. The complexity landscape of hypersphere classification with respect to combinations of structural and conciseness parameters: V_R and V_B are the sets of red and blue points, respectively; d is the dimension; tw is the incidence treewidth of the data representation; econ is the conciseness of the explanation, and dcon is the data conciseness. NP-h$_{\geq i}$ means that the problem becomes NP-hard for parameter values of at least 4, while W[j]-h means that the problem is hard for the complexity class W[j] and hence is unlikely to be fixed-parameter tractable (Downey & Fellows, 2013).

successes in closely related areas such as clustering and data completion (Ganian et al., 2022; 2018), we consider the *incidence treewidth* tw of the data representation. Using a non-trivial dynamic programming procedure, we obtain a fixed-parameter algorithm for binary-data hypersphere classification parameterized by either tw + econ or tw + dcon; in other words, each of the two notions of conciseness suffices for tractability of hypersphere clustering for data that is "well-structured," in the sense of having small incidence treewidth. Moreover, as a byproduct of our algorithm, we also obtain the XP-tractability of binary-data hypersphere classification parameterized by tw alone.

A summary of our results is provided in Table 1.

Related Work.

While there is, to the best of our knowledge, no prior work targeting the complexity of hypersphere classification of binary data, there is a significant work on the real-valued variant of the problem by the machine learning community (Cooper, 1962; Wang et al., 2007; 2005; Astorino & Gaudioso, 2009; Astorino et al., 2016), where they studied the optimization version of the problem in which one seeks the smallest bounding sphere that separates the blue points from the red ones. Our results extend to the optimization version of the problem for binary data, as mentioned in Section 7. Many of the above works consider relaxations of the real-valued optimization problem, in which the sphere sought is not of minimum radius (Wang et al., 2007; 2005; Astorino & Gaudioso, 2009; Astorino et al., 2016)—allowing for error or for outliers—and reduce the problem to some fragment of quadratic programming. We point out that the problem is also related to that of finding a minimum bounding sphere to distinguish/discriminate a set of objects (i.e., one-class classification) (Tax & Duin, 1999), which is, in turn, inspired by the Support Vector Machine models introduced in (Vapnik, 1995).

The hypersphere classification of real-valued low-dimensional data has also been studied in the context of point separability within the field of computational geometry. In particular, the separability of two sets of points in $\mathbb{R}^2$ by a circle was studied by O'Rourke, Kosaraju and Megiddo (1986), who established the linear-time tractability of that case. They also observed that their result could be lifted to an $\mathcal{O}(n^d)$-time algorithm for the separability of n d-dimensional data points by a hypersphere; however this is superseded by the $n^{\mathcal{O}(1)}$-time algorithm observed in Proposition 1, which runs in polynomial time even for unbounded values of d. Several authors also studied related point-separation problems in $\mathbb{R}^2$ and $\mathbb{R}^3$, such as separability of points via polyhedra (Megiddo, 1988), L-shapes (Sheikhi et al., 2015) and a variety of other objects (Agarwal et al., 2006; Alegría et al., 2022).

2. Preliminaries

For $\ell \in \mathbb{N}$, we write $[\ell]$ for $\{1, \ldots, \ell\}$. For convenience, we identify each vector $\vec{v} = (v_1, \ldots, v_d)$ with the point $(v_1, \ldots, v_d)$ in d-dimensional space.

Problem Definition and Terminology. For two vectors $\vec{a}, \vec{b} \in \{0,1\}^d$, we denote by $\delta(\vec{a}, \vec{b})$ the Hamming distance between $\vec{a}$ and $\vec{b}$. For a vector $\vec{v} \in \{0,1\}^d$ and $r \in \mathbb{N}$, denote by $B(\vec{v}, r)$ the hypersphere (i.e., ball) centered at $\vec{v}$ and of radius r; that is, the set of all vectors $\vec{x} \in \{0,1\}^d$ satisfying $\delta(\vec{v}, \vec{x}) \leq r$. Similarly, for vectors over $[0,1]^d$ we denote by $B(\vec{v}, r)$ the hypersphere (i.e., ball) centered at $\vec{v}$ and of radius r with respect to the Euclidean distance in $\mathbb{R}^d$.

The problem under consideration in this paper is defined as follows:

BINARY HYPERSPHERE CLASSIFICATION (BHC)

Input:	A set $V = V_R \cup V_B$ of d-dimensional vectors over the binary domain $D = \{0,1\}$, where $V_R \cap V_B = \emptyset$.
Question:	Is there a vector $\vec{c} \in D^d$ and $r \in \mathbb{N}$ such that $V_B \subseteq B(\vec{c}, r)$ and $V_R \cap B(\vec{c}, r) = \emptyset$?

Throughout the paper, we will refer to the vectors in V_R as "red" and those in V_B as "blue". We also denote by $\mathbf{1}(\vec{v})$ the set of all coordinates i such that $\vec{v}[i] = 1$ and we write $\mathsf{con}(\vec{v})$ for the *conciseness* of $\vec{v}$, i.e., $\mathsf{con}(\vec{v}) = |\mathbf{1}(\vec{v})|$. Moreover, observe that the Hamming distance $\delta(\vec{v}, \vec{c})$ of two vectors $\vec{v}$ and $\vec{c}$ can be written as $\delta(\vec{v}, \vec{c}) = |\mathbf{1}(\vec{c})| + |\mathbf{1}(\vec{v})| - 2|\mathbf{1}(\vec{v}) \cap \mathbf{1}(\vec{c})|$.

Naturally, one may also consider the analogous problem of REAL-VALUED HYPERSPHERE CLASSIFICATION, where the only distinction is that the domain is $[0, 1]$ instead of $\{0, 1\}$. As mentioned in the introduction, this problem can be shown to be polynomial-time solvable and hence is not considered further in our complexity-theoretic analysis.

Proposition 1. REAL-VALUED HYPERSPHERE CLASSIFICATION *can be solved in polynomial time.*

Parameterized Complexity. In parameterized algorithmics (Flum & Grohe, 2006; Downey & Fellows, 2013; Cygan et al., 2015) the running-time of an algorithm is studied with respect to a parameter $k \in \mathbb{N}_0$ and input size n. The basic idea is to find a parameter that describes the structure of the instance such that the combinatorial explosion can be confined to this parameter. In this respect, the most favorable complexity class is FPT(*fixed-parameter tractable*) which contains all problems that can be decided by an algorithm running in time $f(k) \cdot n^{\mathcal{O}(1)}$, where f is a computable function. Algorithms with this running-time are called *fixed-parameter algorithms*. A less favorable outcome is an XP *algorithm*, which is an algorithm running in time $\mathcal{O}(n^{f(k)})$; problems admitting such algorithms belong to the class XP.

Showing that a parameterized problem is hard for the complexity classes $\mathsf{W}[1]$ or $\mathsf{W}[2]$ rules out the existence of a fixed-parameter algorithm under well-established complexity-theoretic assumptions. Such hardness results are typically established via a *parameterized reduction*, which is an analogue of a classical polynomial-time reduction with two notable distinctions: a parameterized reduction can run in time $f(k) \cdot n^{\mathcal{O}(1)}$, but the parameter of the produced instance must be upper-bounded by a function of the parameter in the original instance.

Treewidth. A *nice tree-decomposition* $\mathcal{T}$ of a graph $G = (V, E)$ is a pair (T, χ), where T is a tree (whose vertices are called *nodes*) rooted at a node t_r and χ is a function that assigns each node t a set $\chi(t) \subseteq V$ such that the following hold:

- For every $uv \in E$ there is a node t such that $u, v \in \chi(t)$.
- For every vertex $v \in V$, the set of nodes t satisfying $v \in \chi(t)$ forms a subtree of T.
- $|\chi(\ell)| = 0$ for every leaf ℓ of T and $|\chi(t_r)| = 0$.
- There are only three kinds of non-leaf nodes in T:
  - **Introduce node:** a node t with exactly one child t' such that $\chi(t) = \chi(t') \cup \{v\}$ for some vertex $v \notin \chi(t')$.
  - **Forget node:** a node t with exactly one child t' such that $\chi(t) = \chi(t') \setminus \{v\}$ for some vertex $v \in \chi(t')$.
  - **Join node:** a node t with two children t_1, t_2 such that $\chi(t) = \chi(t_1) = \chi(t_2)$.

The *width* of a nice tree-decomposition (T, χ) is the size of a largest set $\chi(t)$ minus 1, and the *treewidth* of the graph G, denoted $\mathsf{tw}(G)$, is the minimum width of a nice tree-decomposition of G.

We let T_t denote the subtree of T rooted at a node t, and use $\chi(T_t)$ to denote the set $\bigcup_{t' \in V(T_t)} \chi(t')$ and G_t to denote the graph $G[\chi(T_t)]$ induced by the vertices in $\chi(T_t)$. Efficient fixed-parameter algorithms are known for computing a nice tree-decomposition of near-optimal width:

Proposition 2 (Kloks 1994; Korhonen 2021)**.** *There exists an algorithm which, given an n-vertex graph G and an integer k, in time $2^{\mathcal{O}(k)} \cdot n$ either outputs a nice tree-decomposition of G of width at most $2k+1$ and $\mathcal{O}(n)$ nodes, or determines that $\mathsf{tw}(G) > k$.*

Constraint Satisfaction Problems. Let $D = \{0, 1\}$ and let n an integer. An n-ary relation on D is a subset of D^n. An instance I of a *Boolean constraint satisfaction problem* (CSP) is a pair (V, C), where V is a finite set of variables and C is a set of constraints. A *constraint* $c \in C$ consists of a *scope*, denoted by $V(c)$, which is an ordered list of a subset of V, and a relation, denoted by $R(c)$, which is a $|V(c)|$-ary relation on D; $|V(c)|$ is the *arity* of c.

A *solution* to a CSP instance $I = (V, C)$ is a mapping $\tau : V \to D$ such that $\langle \tau(v_1), \ldots, \tau(v_{|V(c)|}) \rangle \in R(c)$ for every $c \in C$ with $V(c) = \langle v_1, \ldots, v_{|V(c)|} \rangle$. A CSP instance is *satisfiable* if and only if it has at least one solution.

3. NP-hardness of BHC Restricted to Two Red or Blue Vectors

In this section, we show that BHC remains NP-hard even when one of the two sets (V_B or V_R) has size at most two. In particular, let us denote by 2RED-BHC the restriction of BHC to instances in which the number of red vectors is two (i.e., $|V_\mathrm{R}| = 2$), and by 2BLUE-BHC the restriction of BHC to instances in which the number of blue vectors is two (i.e., $|V_\mathrm{B}| = 2$).

Theorem 3. 2RED-BHC *and* 2BLUE-BHC *are* NP-*complete.*

Proof Sketch. Proving membership in NP is straightforward and is omitted. We begin with the following problem, which is known to be NP-hard (Frances & Litman, 1997):

Minimum Radius (MR)

Input:	A set V of $(2n)$-dimensional binary vectors where $n \in \mathbb{N}$.
Question:	Is there a $(2n)$-dimensional center vector $\vec{c}$ such that $V \subseteq B(\vec{c}, n)$?

Denote by Rest-MR the restriction of MR to instances in which V contains the $(2n)$-dimensional all-zero vector $\vec{0}_{2n}$ and we ask for a center vector $\vec{c}$ that contains exactly n ones. We first show that Rest-MR remains NP-hard via a polynomial-time Turing-reduction from MR to Rest-MR.

At this point, to complete the proof of the theorem it suffices to exhibit a polynomial-time reduction from Rest-MR to 2RED-BHC (an analogous reduction is also be used for 2BLUE-BHC). Given an instance V of Rest-MR, we construct an instance V^+ of 2Red-BHC such that V is a Yes-instance of Rest-MR if and only if V^+ is a Yes-instance of 2RED-BHC. Without loss of generality, we may assume that (the $(2n)$-dimensional all 1's vector) $\vec{1}_{2n} \in V$ since V is a Yes-instance of Rest-MR if and only if $V \cup \{\vec{1}_{2n}\}$ is. The previous statement is true since $\vec{0}_{2n} \in V \subseteq B(\vec{c}, n)$, where $\vec{c}$ contains exactly n 1's, if and only if $(V \cup \{\vec{1}_{2n}\}) \subseteq B(\vec{c}, n)$.

To construct V^+, we extend each $(2n)$-dimensional vector $\vec{v} \in V$ by adding two coordinates, that we refer to as coordinates q_{2n+1} and q_{2n+2}, and setting their values to 0 and 1, respectively; let $\vec{v}^+$ denote the extension of $\vec{v}$. Let V_b be the resulting set of (extended) vectors from V, and let $V_r = \{\vec{0}_{2n+2}, \vec{1}_{2n+2}\}$, where $\vec{0}_{2n+2}, \vec{1}_{2n+2}$ are the $(2n+2)$-dimensional all-zero and all-one vectors, respectively. Finally, let $V^+ = V_b \cup V_r$. We can now show that V is a Yes-instance of Rest-MR if and only if V^+ is a Yes-instance of 2Red-BHC. □

4. Basic Parameterizations for BHC

We follow up on Theorem 3 by considering the two remaining obvious parameterizations of the problem, notably d and $|V|$. The former case is trivial since it bounds the size of the input.

Observation 4. BHC *is* FPT *parameterized by* d.

Next, we give a fixed-parameter algorithm for BHC parameterized by the total number of vectors.

Theorem 5. BHC *is* FPT *parameterized by the number of red vectors plus blue vectors.*

Proof Sketch. Let k be the total number of red vectors plus blue vectors. For convenience, we will consider the matrix representation of the input, in which the vectors are represented as the rows of a matrix M. Observe that, since there are k rows of binary coordinates in M, the total number of different column configurations of M is at most 2^k. The idea behind the fixed-parameter algorithm is to encode the problem as an instance of an Integer Linear Program (ILP) with 2^{k+1} variables—two for every column type. One such variable will capture the number of "1"s the center uses in the columns belonging to that type, while the other simply captures the number of "0"s of the center in these columns. The main reason why this suffices is that the exact positions of these "0"s and "1"s within these column types is irrelevant when considering the distance between an arbitrary point and a center. The constraints simply ensure that the distance between the center and each blue point is strictly smaller than the distance between the center and each red point.

It is well known that such an ILP instance can then be solved in FPT time using the classical result of Lenstra (H. W. Lenstra, 1983; Kannan, 1987; Frank & Tardos, 1987). Once the coordinates of the desired (hypersphere) center have been determined, the radius of the hypersphere can be set as the maximum Hamming distance between the center and the blue vectors in M. □

5. The Complexity of BHC with Conciseness

In this section, we perform a detailed analysis of the complexity of BHC with respect to conciseness. We will distinguish between data conciseness and explanation conciseness. Data conciseness is the maximum number of 1's appearing in any red or blue vector of the instance I and is denoted $\mathsf{dcon}(I)$; that is, $\mathsf{dcon}(I) = \max_{\vec{v} \in V_R \cup V_B} \mathsf{con}(\vec{v})$. The explanation conciseness on the other hand is the maximum number of 1's appearing in the sought-after vector $\vec{c}$. To capture this aspect of the problem, we define a new version of BHC that imposes a bound econ on the explanation conciseness of the vector $\vec{c}$. Formally, let EC-BHC be defined analogously to BHC, but where we are additionally given an integer econ and the question is whether there exists a vector $\vec{c} \in D^d$ of conciseness at most econ and $r \in \mathbb{N}$ such that $V_B \subseteq B(\vec{c}, r)$ and $V_R \cap B(\vec{c}, r) = \emptyset$.

5.1. Data Conciseness

In this subsection, we analyse the parameterized complexity of BHC parameterized by the conciseness of the data $\mathsf{dcon}(I)$. We start by showing that instances I satisfying $\mathsf{dcon}(I) \leq 3$ can be solved in polynomial-time.

Theorem 6. *The restriction of* BHC *to instances I satisfying* $\mathsf{dcon}(I) \leq 3$ *can be solved in time* $\mathcal{O}(|V|d)$.

Proof Sketch. The main idea behind the algorithm is a case distinction based on the minimum distance M_r of any red vector from a solution vector $\vec{c}$. Note first that if we fix a solution $\vec{c}$ for I, then $|\mathbf{1}(\vec{v})| - 2|\mathbf{1}(\vec{v}) \cap \mathbf{1}(\vec{c})|$ can be used instead of the Hamming distance to compare the distances of two vectors from $\vec{c}$. Altogether, we obtain four cases for $M_r = \min_{\vec{r}\in V_r} |\mathbf{1}(\vec{r})| - 2|\mathbf{1}(\vec{r}) \cap \mathbf{1}(\vec{c})|$, i.e., (1) $M_r \leq -1$, (2) $M_r = 0$, (3) $M_r = 1$, and (4) $M_r \geq 2$. While (1) and (4) are trivial to solve, (2) and (3) are solved via a reduction to a Boolean CSP instance that can be solved in polynomial-time because its relational language is closed under a majority operation. To illustrate the ideas for (1), note that if $M_r \leq -1$, then it is easy to show that any solution vector must be 1 on all coordinates that has a 1 for any blue vector. But this means that the instance has a solution if and only if the vector $\vec{c}$ that is 1 at all coordinates in $\bigcup_{v\in V_{\mathrm{B}}} \mathbf{1}(\vec{v})$ and otherwise 0 is a solution for I; this is because setting the coordinates outside of $\bigcup_{v\in V_{\mathrm{B}}} \mathbf{1}(\vec{v})$ to 1 only reduces the distance of $\vec{c}$ to vectors in V_{R}. ☐

We now show that BHC is already NP-complete for instances with $\mathsf{dcon}(I) \geq 4$; in fact, this holds even when restricted to the class of instances where $\mathsf{dcon}(I)$ is precisely 4.

Theorem 7. BHC *is* NP-*complete even when restricted to instances I satisfying* $\mathsf{dcon}(I) = 4$.

We prove Theorem 7 via a reduction from the CSP problem using a constraint language Γ_4 that is NP-hard by Schaefer's theorem (Schaefer, 1978; Chen, 2009). Γ_4 is the Boolean constraint language containing the following two Boolean 4-ary relations: the *red* relation R_{R} containing all tuples having at most 2 ones and the *blue* relation R_{B} containing all tuples having at least 3 ones.

Lemma 8. *CSP*(Γ_4) *is* NP-*complete.*

With Lemma 8 in hand, we establish Theorem 7 by designing a polynomial-time reduction from CSP(Γ_4). We note that, as mentioned already in the proof of Theorem 3, inclusion in NP is trivial.

Proof Sketch for Theorem 7. Let $I = (V, C)$ be the given instance of CSP(Γ_4). We denote by C_r/C_b the set of all constraints c in C with $R(c) = R_{\mathrm{R}}/R(c) = R_{\mathrm{B}}$; note that $C = C_r \cup C_b$. We will construct the instance $I' = (V_{\mathrm{R}}, V_{\mathrm{B}}, d)$ of BHC as follows. First, we introduce one coordinate d_v for every variable $v \in V$. Moreover, for every constraint $c \in C_r$, we introduce the red vector $\vec{r}_c$ that is 1 on all coordinates that correspond to variables within the scope of c and is 0 otherwise, i.e., $\vec{r}_c$ is 1 exactly on the coordinates in $\{ d_v \mid v \in S(c) \}$ and 0 at all other coordinates. Similarly, for every constraint $c \in C_b$, we introduce the blue vector $\vec{b}_c$ that is 1 on all coordinates that correspond to variables within the scope of c and 0 otherwise.

Finally, we will introduce two gadgets which will enforce that in every solution $\vec{c}$ of I', it holds that:

(1) there is a red vector $\vec{r} \in V_{\mathrm{R}}$ such that $\vec{c}$ is 1 on at least two coordinates, where $\vec{r}$ is also 1; and
(2) there is a blue vector $\vec{b} \in V_{\mathrm{B}}$ such that $\vec{c}$ is not 1 on all coordinates where $\vec{b}$ is 1.

Towards enforcing (1), we add two new blue vectors $\vec{v}_1^b$ and $\vec{v}_2^b$ together with 8 new coordinates $d_1^b, \ldots, d_8^b$ such that $\vec{v}_1^b$ is 1 exactly at the coordinates $d_1^b, \ldots, d_4^b$ and $\vec{v}_2^b$ is 1 exactly at the coordinates $d_5^b, \ldots, d_8^b$. Moreover, for every i and j with $1 \leq i < j \leq 8$, we introduce a red vector v_{ij}^r that is 1 exactly at the coordinates c_i and c_j plus two additional fresh coordinates.

Towards enforcing (2), we add one new blue vector $\vec{u}_i^b$ together with four fresh coordinates $e_1^i, \ldots, e_4^i$ for every i with $1 \leq i \leq 4$ such that $\vec{u}_i^b$ is 1 exactly on the coordinates $e_1^i, \ldots, e_4^i$. Finally, we add one new red vector $\vec{u}^r$ that is 1 exactly at the coordinates e_1^1, e_1^2, e_1^3, and e_1^4.

We can now complete the proof by showing the equivalence between the original instance I of CSP(Γ_4) and the constructed instance I' of BHC. ☐

5.2. Data Conciseness Plus $|V_{\mathrm{R}}|$ or $|V_{\mathrm{B}}|$

Here we show that if in addition to the input conciseness one also parameterizes by the minimum of the numbers of red vectors and blue vectors, then BHC becomes fixed-parameter tractable.

Theorem 9. BHC *is fixed-parameter tractable parameterized by* $\mathsf{dcon}(I) + \min\{|V_{\mathrm{B}}|, |V_{\mathrm{R}}|\}$.

Proof. Let $I = (V_{\mathrm{R}}, V_{\mathrm{B}}, d)$ be the given instance of BHC. It suffices to show that BHC is fixed-parameter tractable parameterized by $\mathsf{dcon}(I) + |V_{\mathrm{B}}|$ and also by $\mathsf{dcon}(I) + |V_{\mathrm{R}}|$. To avoid any confusion, we remark that it is well known (and easy to see) that establishing fixed-parameter tractability w.r.t. the sum $\alpha + \beta$ of two numbers is equivalent to establishing fixed-parameter tractability w.r.t. the product $\alpha \cdot \beta$ of the same numbers.

The main observation behind the algorithm (for the case $\mathsf{dcon}(I) + |V_{\mathrm{B}}|$) is that the total number of coordinates, where any blue vector can be 1 is at most $|V_{\mathrm{B}}|\mathsf{dcon}(I)$; let $B = \bigcup_{\vec{b}\in V_{\mathrm{B}}} \mathbf{1}(\vec{b})$ be the set of all those coordinates. Since any solution $\vec{c}$ can be assumed to be 0 at any coordinate outside of B, we can solve I by "guessing" (i.e., branching to find) a solution in time $\mathcal{O}(2^{\mathsf{dcon}(I)|V_{\mathrm{B}}|}|V|d)$. More specifically, for every subset B' of the at most $2^{\mathsf{dcon}(I)|V_{\mathrm{B}}|}$ subsets of B, we check in time $\mathcal{O}(|V|d)$ whether the vector $\vec{c}$ that is 1 exactly at the coordinates in B' is a solution for I. If one of those vectors is a solution, then we output it, otherwise we can correctly return that I is a No-instance.

The algorithm for the case where we parameterize by $\mathsf{dcon}(I) + |V_{\mathrm{R}}|$ is almost identical with the only difference being the observation that the set $R = \bigcup_{\vec{r} \in V_{\mathrm{R}}} \mathbf{1}(\vec{r})$ has size at most $2^{\mathsf{dcon}(I)|V_{\mathrm{R}}|}$ and that any solution $\vec{c}$ can be assumed to be 1 at every coordinate in R. $\square$

5.3. Explanation Concisness

Recall that EC-BHC is defined analogously as BHC, but one is additionally given an integer econ and is asked for a solution $\vec{c}$ for BHC with conciseness at most econ. Note that a simple brute-force algorithm that enumerates all potential solution vectors $\vec{c}$ with at most econ 1's shows that EC-BHC is in XP parameterized by econ.

Observation 10. EC-BHC *can be solved in time* $\mathcal{O}(d^{\mathsf{econ}}|V|d)$.

Therefore, it becomes natural to ask whether this can be improved to fixed-parameter tractability. The following two theorems show that this is unlikely to be the case, even if we additionally assume $|V_{\mathrm{B}}| = 1$ or $|V_{\mathrm{R}}| = 1$.

Theorem 11. EC-BHC *is* W[2]*-hard parameterized by the conciseness* econ *of the solution even if* $|V_{\mathrm{R}}| = 1$.

Proof Sketch. We provide a parameterized reduction from the UNIFORM HITTING SET problem, which given a set U of elements, a family $\mathcal{F} \subseteq 2^U$ of subsets of U with $|F| = \ell$ for every $F \in \mathcal{F}$ and an integer k, asks whether $\mathcal{F}$ has a *hitting set* $H \subseteq U$ of size at most k, i.e., $H \cap F \neq \emptyset$ for every $F \in \mathcal{F}$. UNIFORM HITTING SET is W[2]-complete parameterized by k (Downey & Fellows, 2013).

Let $I = (U, \mathcal{F}, k)$ be an instance of UNIFORM HITTING SET with sets of size ℓ. We construct an equivalent instance $I' = (V_{\mathrm{R}}, V_{\mathrm{B}}, d, \mathsf{econ})$ of EC-BHC as follows. We set $\mathsf{econ} = k$. For every $u \in U$, we introduce the (element) coordinate d_u and for every i with $1 \leq i \leq \ell$, we introduce the (dummy) coordinate d'_i. Moreover, for every $F \in \mathcal{F}$, we add the blue vector $\vec{b}_F$ to V_{B}, which is 1 on all coordinates d_u with $u \in F$ and 0 at all other coordinates. Finally, we introduce the red vector $\vec{r}$ that is 1 at all dummy coordinates d'_i and 0 at all element coordinates. This completes the construction of I', which can clearly be achieved in polynomial-time.We can now show that I is a Yes-instance of UNIFORM HITTING SET if and only if I' is a Yes-instance of EC-BHC. $\square$

Theorem 12. EC-BHC *is* W[1]*-hard parameterized by the conciseness* econ *of the solution even if* $|V_{\mathrm{B}}| = 1$.

Proof Sketch. We will provide a parameterized reduction from the MULTI-COLORED INDEPENDENT SET problem, which given an undirected graph $G = (V, E)$, where V is partitioned into k vertex sets $V_1, \dots, V_k$ with $|V_i| = n$ and $G[V_i]$ is a clique and an integer k, asks whether G has an independent set of size at least k; note that such an independent set must contain exactly one vertex from each V_i. MULTI-COLORED INDEPENDENT SET is well-known to be W[1]-complete (Downey & Fellows, 2013).

Let $I = (G, V_1, \dots, V_k, k)$ be an instance of MULTI-COLORED INDEPENDENT SET with $|V_i| = n$ and $V = \bigcup_{i=1}^{k} V_i$. We construct an equivalent instance $I' = (V_{\mathrm{R}}, V_{\mathrm{B}}, d, \mathsf{econ})$ of EC-BHC as follows. We set $\mathsf{econ} = k$. For every $v \in V$, we introduce the (vertex) coordinate d_v and for every i with $1 \leq i \leq nk - 2k + 1$, we introduce the (dummy) coordinate d'_i. Moreover, for every $e = \{u, v\} \in E(G)$, we add the red vector $\vec{r}_e$ to V_{R}, which is 1 on the coordinates d_u and d_v as well as the coordinate d'_i for every i with $1 \leq i \leq nk - 2k + 1$. We also add the red vector $\vec{r}$ to V_{R} that is 1 at the coordinates d'_i with $1 \leq i \leq nk - 2k + 1$. Finally, we introduce the blue vector $\vec{b}$ that is 1 at all vertex coordinates d_v and 0 at all dummy coordinates. This completes the construction of I', which can clearly be achieved in polynomial-time. We can now show that I is a Yes-instance of MULTI-COLORED INDEPENDENT SET if and only if I' is a Yes-instance of EC-BHC. $\square$

5.4. Data and Explanation Conciseness

As our final result in this section, we show that EC-BHC is fixed-parameter tractable when parameterized by data and explanation conciseness combined.

Theorem 13. EC-BHC *can be solved in time* $\mathcal{O}(\mathsf{dcon}(I)^{\mathsf{econ}}|V|d)$ *and is therefore fixed-parameter tractable parameterized by* $\mathsf{econ} + \mathsf{dcon}$.

Proof. Let $I = (V_{\mathrm{R}}, V_{\mathrm{B}}, d, \mathsf{econ})$ with $V = V_{\mathrm{R}} \cup V_{\mathrm{B}}$ be the given instance of EC-BHC. The main idea behind the algorithm is as follows. We start by initializing the solution vector $\vec{c}$ to the all-zero vector. We then check in time $\mathcal{O}(|V|d)$ whether $\vec{c}$ is already a solution. If so, we are done. Otherwise, there must exist a red vector $\vec{r} \in V_{\mathrm{R}}$ and a blue vector $\vec{b} \in V_{\mathrm{B}}$ such that $\delta(\vec{r}, \vec{c}) \leq \delta(\vec{b}, \vec{c})$ and therefore: $|\mathbf{1}(\vec{r})| + |\mathbf{1}(\vec{c})| - 2|\mathbf{1}(\vec{r}) \cap \mathbf{1}(\vec{c})| \leq |\mathbf{1}(\vec{b})| + |\mathbf{1}(\vec{c})| - 2|\mathbf{1}(\vec{b}) \cap \mathbf{1}(\vec{c})|$, or in short $|\mathbf{1}(\vec{r})| - 2|\mathbf{1}(\vec{r}) \cap \mathbf{1}(\vec{c})| \leq |\mathbf{1}(\vec{b})| - 2|\mathbf{1}(\vec{b}) \cap \mathbf{1}(\vec{c})|$. It follows that any vector $\vec{c}'$ with $\mathbf{1}(\vec{c}) \subseteq \mathbf{1}(\vec{c}')$ and $\delta(\vec{r}, \vec{c}') > \delta(\vec{b}, \vec{c}')$ has to be obtained from $\vec{c}$ by flipping at least one coordinate in $B = \mathbf{1}(\vec{b}) \setminus (\mathbf{1}(\vec{r}) \cup \mathbf{1}(\vec{c}))$ from 0 to 1; note that $|B| \leq \mathsf{dcon}(I)$. We can therefore branch on the coordinates of B, and for every such choice $b \in B$, we continue with the vector $\vec{c}'$ obtained from $\vec{c}$ after flipping the coordinate b from 0 to 1. We stop if either we have reached a solution or if the number of 1's in the current vector $\vec{c}$ exceeds the conciseness upper bound econ. In other words, we can solve the problem using a branching algorithm that has at most $|\mathbf{1}(\vec{b})| \leq \mathsf{dcon}(I)$ many choices per branch, uses time $\mathcal{O}(|V|d)$ per search-tree node, and makes at most econ branching decisions before it stops. Therefore, the run-time of the algorithm is $\mathcal{O}(\mathsf{dcon}(I)^{\mathsf{econ}}|V|d)$. $\square$

6. A Treewidth-Based Algorithm for BHC

Let the *incidence graph* G_I of an instance $I = (V_\mathrm{R}, V_\mathrm{B}, d)$ of BHC be the bipartite graph defined as follows. First of all, $V(G_I) = V_\mathrm{R} \cup V_\mathrm{B} \cup [d]$. As for the edge set, there is an edge $\vec{v}c \in E(G_I)$ between a vector $\vec{v} \in V_\mathrm{R} \cup V_\mathrm{B}$ and a coordinate $c \in [d]$ if and only if $c \in \mathbf{1}(\vec{v})$. We identify the vertices of G_I with the vectors in $V_\mathrm{R} \cup V_\mathrm{B}$ and the coordinates in $[d]$. That is, for a set of vertices X in G_I, we often say "vectors in X" or "coordinates in X" to mean the vectors/the coordinates associated with the vertices in X.

This section is dedicated to proving the following technical theorem, which implies all the claimed tractability results concerning the treewidth of the incidence graph:

Theorem 14. *Given an instance $I = (V_\mathrm{R}, V_\mathrm{B}, d)$ of* BHC *and a nice tree-decomposition $\mathcal{T} = (T, \chi)$ of G_I of width w, there is an algorithm solving I in time $(2\min\{\mathsf{econ}_{min}, \mathsf{dcon}(I)\})^{2w+2} \cdot (|V| + d)^{\mathcal{O}(1)}$, where econ_{min} is the minimum conciseness of any center. Moreover, if I is* Yes*-instance, then the algorithm outputs a center with conciseness econ_{min} and minimum radius among all such centers.*

Proof Sketch. We begin by enumerating each choice of center conciseness $\lambda = 0, 1, 2, \ldots, d$ and radius $r = 0, 1, 2, \ldots, d$, and aim to construct a solution with exactly this conciseness and radius. The algorithm is a bottom-up dynamic programming along the nice tree-decomposition $\mathcal{T}$. We first describe the records that we need to compute for every node t of T. Given the description of the records, we need to show that for each of the node types (i.e., leaf/introduce/forget/join), we can compute the records from the records of their children. Finally, we need to also show that given the records for the root node of the tree-decomposition, we can decide whether I is a Yes-instance and if so output a center vector $\vec{c}$ such that $|\mathbf{1}(\vec{c})| = \lambda$, $V_\mathrm{B} \subseteq B(\vec{c}, r)$, and $V_\mathrm{R} \cap B(\vec{c}, r) = \emptyset$.

We begin by describing the record Γ_t for each node $t \in T$. We can think about Γ_t as a map that maps a tuple $\mathcal{C} = (c_\text{past}, c_\text{future}, C_\text{bag}, V_\text{bag}) \in \mathbb{N} \times \mathbb{N} \times 2^{\chi(t)} \times \mathbb{N}^{|\chi(t)|}$ to either a vector $\vec{c}_t = \{0,1\}^d$ with $\mathsf{con}(\vec{c}_t) = \lambda$ or $\perp$. The intuition behind the record is that the tuple $(c_\text{past}, c_\text{future}, C_\text{bag}, V_\text{bag})$ is mapped to an arbitrary vector $\vec{c}_t$ such that

1. c_past is the number of non-zero coordinates of $\vec{c}_t$ on already "forgotten" coordinates, i.e., $c_\text{past} = |\mathbf{1}(\vec{c}_t) \cap \chi(T_t) \setminus \chi(t)|$;
2. c_future is the number of non-zero coordinates of $\vec{c}_t$ on coordinates that are not yet introduced, i.e., $c_\text{future} = |\mathbf{1}(\vec{c}_t) \cap [d] \setminus \chi(T_t)|$;
3. $C_\text{bag} = \mathbf{1}(\vec{c}_t) \cap \chi(t)$;
4. V_bag contains, for every vector $\vec{v} \in \chi(t)$, the number of ones on "forgotten" coordinates in $\mathbf{1}(\vec{c}_t)$, that is $V_\text{bag}(\vec{v}) = |\mathbf{1}(\vec{c}_t) \cap \mathbf{1}(\vec{v}) \cap (\chi(T_t) \setminus \chi(t))|$;
5. no forgotten red vector is at distance at most r from $\vec{c}_t$, i.e., $V_\mathrm{R} \cap (\chi(T_t) \setminus \chi(t)) \cap B(\vec{c}_t, r) = \emptyset$; and
6. all forgotten blue vectors are at distance at most r from $\vec{c}_t$, i.e., $(V_\mathrm{B} \cap (\chi(T_t) \setminus \chi(t))) \subseteq B(\vec{c}_t, r)$.

We say that a vector $\vec{c}_t$ that satisfies all the above properties is *compatible* with $(c_\text{past}, c_\text{future}, C_\text{bag}, V_\text{bag})$ for t. Moreover, $(c_\text{past}, c_\text{future}, C_\text{bag}, V_\text{bag})$ is mapped to $\perp$ if and only if no vector in $\{0,1\}^d$ is compatible with $(c_\text{past}, c_\text{future}, C_\text{bag}, V_\text{bag})$.

First note that if t is the root node, then $\chi(t)$ is empty and $\chi(T_t) \setminus \chi(t)$ contains all vectors in the instance. Hence, if any tuple is mapped to a vector in the root, then the vector is a solution by properties 5 and 6 above.

We say that a tuple $\mathcal{C} = (c_\text{past}, c_\text{future}, C_\text{bag}, V_\text{bag})$ is *achievable for* Γ_t if the following holds:

- $c_\text{past} + c_\text{future} + |C_\text{bag}| = \lambda$; and
- for all vectors $\vec{v} \in \chi(t)$: $V_\text{bag}(\vec{v}) \leq \min\{\lambda, |\mathbf{1}(\vec{v}) \cap (\chi(T_t) \setminus \chi(t))|\}$.

Note that if $\mathcal{C}$ is not achievable for Γ_t, then no vector with conciseness λ can be compatible with $\mathcal{C}$. Hence, the table Γ_t will only contain the achievable tuples for Γ_t. We observe that $|\Gamma_t| \leq \lambda^2 \cdot 2^{\chi(t)} \cdot (\min\{\lambda, \mathsf{dcon}(I)\})^{|\chi(t)|}$. We can now compute the records in a leaf-to-root fashion at each of the four different types of nodes in $\mathcal{T}$. $\square$

Combining Theorem 14 and Proposition 2, we get the following three corollaries:

Corollary 15. BHC *and* EC-BHC *are in* XP *parameterized by* $\mathsf{tw}(G_I)$.

Corollary 16. EC-BHC *is fixed-parameter tractable parameterized by* $\mathsf{tw}(G_I) + \mathsf{econ}$.

Corollary 17. BHC *and* EC-BHC *are fixed-parameter tractable parameterized by* $\mathsf{tw}(G_I) + \mathsf{dcon}(I)$.

7. Concluding Remarks

In this paper, we studied hypersphere classification problems from a parameterized complexity perspective, focusing strongly on conciseness. We considered conciseness in terms of the sought-after explanation and in terms of the feature vectors in the training data. Our algorithmic and lower-bound results draw a comprehensive complexity map of hypersphere classification. This map pinpoints the exact complexity of the various combinations of parameters which can either measure the structural properties of the input data or the conciseness of data or explanations.

All our lower and upper complexity bounds are essentially tight, with a single exception: While we show that hypersphere classification without conciseness restrictions is XP-tractable when parameterized by treewidth alone, whether the problem is fixed-parameter tractable or W[1]-hard under this parameterization is left open.

Finally, we remark that all our results carry over to the case where one aims to find a minimum-radius separating hypersphere (instead of merely deciding whether one exists) that classifies the training data. This problem has also been extensively studied (Cooper, 1962; Wang et al., 2007; 2005; Astorino & Gaudioso, 2009; Astorino et al., 2016).

Acknowledgments

Robert Ganian acknowledges support from the Austrian Science Fund (FWF, project Y1329). Iyad Kanj acknowledges support from DePaul University through a URC grant 602061. Stefan Szeider acknowledges support from the Austrian Science Fund (FWF, projects P32441), and from the Vienna Science and Technology Fund (WWTF, project ICT19-065). Sebastian Ordyniak acknowledges support from the Engineering and Physical Sciences Research Council (EPSRC, project EP/V00252X/1).

References

Agarwal, P. K., Aronov, B., and Koltun, V. Efficient algorithms for bichromatic separability. *ACM Trans. Algorithms*, 2(2):209–227, 2006.

Alegría, C., Orden, D., Seara, C., and Urrutia, J. Separating bichromatic point sets in the plane by restricted orientation convex hulls. *Journal of Global Optimization*, pp. 1–34, 2022.

Astorino, A. and Gaudioso, M. A fixed-center spherical separation algorithm with kernel transformations for classification problems. *Comput. Manag. Sci.*, 6(3):357–372, 2009.

Astorino, A., Fuduli, A., and Gaudioso, M. Nonlinear programming for classification problems in machine learning. In *AIP Conference Proceedings*, volume 1776, 2016.

Barceló, P., Monet, M., Pérez, J., and Subercaseaux, B. Model interpretability through the lens of computational complexity. In *Proceedings of the Thirty-Third Annual Conference on Neural Information Processing Systems NeurIPS*, pp. 15487–15498, 2020.

Blanc, G., Lange, J., and Tan, L. Provably efficient, succinct, and precise explanations. In *Proceedings of the Thirty-Fourth Conference on Neural Information Processing Systems NeurIPS*, pp. 6129–6141, 2021.

Chalasani, P., Chen, J., Chowdhury, A. R., Wu, X., and Jha, S. Concise explanations of neural networks using adversarial training. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, pp. 1383–1391. PMLR, 2020.

Chen, H. A rendezvous of logic, complexity, and algebra. *ACM Comput. Surv.*, 42(1):2:1–2:32, 2009.

Cooper, P. W. The hypersphere in pattern recognition. *Inf. Control.*, 5(4):324–346, 1962.

Cygan, M., Fomin, F. V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., and Saurabh, S. *Parameterized Algorithms*. Springer, 2015.

Dahiya, Y., Fomin, F. V., Panolan, F., and Simonov, K. Fixed-parameter and approximation algorithms for PCA with outliers. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139, pp. 2341–2351, 2021.

Darwiche, A. and Hirth, A. On the reasons behind decisions. In *ECAI 2020 - 24th European Conference on Artificial Intelligence*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pp. 712–720, 2020.

Doshi-Velez, F. and Kim, B. A roadmap for a rigorous science of interpretability. *CoRR*, abs/1702.08608, 2017. URL `http://arxiv.org/abs/1702.08608`.

Downey, R. G. and Fellows, M. R. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

Flum, J. and Grohe, M. *Parameterized Complexity Theory*, volume XIV of *Texts in Theoretical Computer Science. An EATCS Series*. Springer, Berlin, 2006.

Frances, M. and Litman, A. On covering problems of codes. *Theory of Computing Systems*, 30:113–119, 1997.

Frank, A. and Tardos, É. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987.

Ganian, R. and Korchemna, V. The complexity of bayesian network learning: Revisiting the superstructure. In *Proceedings of the Thirty-Fourth Conference on Neural Information Processing Systems NeurIPS*, pp. 430–442, 2021.

Ganian, R., Kanj, I. A., Ordyniak, S., and Szeider, S. Parameterized algorithms for the matrix completion problem. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, pp. 1642–1651, 2018.

Ganian, R., Hamm, T., Korchemna, V., Okrasa, K., and Simonov, K. The complexity of k-means clustering when little is known. In *International Conference on Machine Learning (ICML)*, volume 162, pp. 6960–6987, 2022.

Grüttemeier, N. and Komusiewicz, C. Learning bayesian networks under sparsity constraints: A parameterized complexity analysis. *J. Artif. Intell. Res.*, 74:1225–1267, 2022.

H. W. Lenstra, J. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4): 538–548, 1983.

Hurtado, F., Seara, C., and Sethia, S. Red-blue separability problems in 3D. In *Computational Science and Its Applications*, pp. 766–775, Berlin, Heidelberg, 2003. Springer Berlin.

Ignatiev, A., Marques-Silva, J., Narodytska, N., and Stuckey, P. J. Reasoning-based learning of interpretable ML models. In Zhou, Z. (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence IJCAI*, pp. 4458–4465, 2021.

Izza, Y., Ignatiev, A., Narodytska, N., Cooper, M. C., and Marques-Silva, J. Provably precise, succinct and efficient explanations for decision trees. *CoRR*, abs/2205.09569, 2022.

Kannan, R. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12 (3):415–440, 1987.

Kim, S., Lee, K., and Jeong, Y. Norm ball classifier for one-class classification. *Ann. Oper. Res.*, 303(1):433–482, 2021.

Kloks, T. *Treewidth, Computations and Approximations*, volume 842 of *Lecture Notes in Computer Science*. Springer, 1994.

Korhonen, T. A single-exponential time 2-approximation algorithm for treewidth. In *62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 184–192. IEEE, 2021.

Laber, E. S. and Murtinho, L. On the price of explainability for some clustering problems. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139, pp. 5915–5925, 2021.

Lipton, Z. C. The mythos of model interpretability. *Commun. ACM*, 61(10):36–43, 2018.

Makarychev, K. and Shan, L. Near-optimal algorithms for explainable k-medians and k-means. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139, pp. 7358–7367, 2021.

Megiddo, N. On the complexity of polyhedral separability. *Discrete & Computational Geometry*, 3(4):325–337, 1988.

Monroe, D. Ai, explain yourself. *Commun. ACM*, 61(11): 11–13, 2018.

Nori, H., Caruana, R., Bu, Z., Shen, J. H., and Kulkarni, J. Accuracy, interpretability, and differential privacy via explainable boosting. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139, pp. 8227–8237, 2021.

Ordyniak, S. and Szeider, S. Parameterized complexity results for exact bayesian network structure learning. *J. Artif. Intell. Res.*, 46:263–302, 2013.

Ordyniak, S., Paesani, G., and Szeider, S. The parameterized complexity of finding concise local explanations. In Elkind, E. (ed.), *The 32nd International Joint Conference on Artificial Intelligence (IJCAI-23), August 19–25, 2023, Macao, S.A.R.* International Joint Conferences on Artificial Intelligence Organization, 2023. Main Track, to appear.

O'Rourke, J., Kosaraju, S. R., and Megiddo, N. Computing circular separability. *Discrete & computational geometry*, 1:105–114, 1986.

Ribeiro, M. T., Singh, S., and Guestrin, C. Anchors: High-precision model-agnostic explanations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 1527–1535. AAAI Press, 2018.

Schaefer, T. J. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing, May 1-3, 1978, San Diego, California, USA*, pp. 216–226. ACM, 1978.

Sheikhi, F., Mohades, A., de Berg, M., and Davoodi, M. Separating bichromatic point sets by l-shapes. *Comput. Geom.*, 48(9):673–687, 2015.

Shih, A., Choi, A., and Darwiche, A. A symbolic approach to explaining bayesian network classifiers. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI*, pp. 5103–5111, 2018.

Shih, S., Tien, P., and Karnin, Z. GANMEX: one-vs-one attributions using gan-based model explainability. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139, pp. 9592–9602, 2021.

Simonov, K., Fomin, F. V., Golovach, P. A., and Panolan, F. Refined complexity of PCA with outliers. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pp. 5818–5826, 2019.

Tax, D. M. J. and Duin, R. P. W. Data domain description using support vectors. In *The European Symposium on Artificial Neural Networks*, 1999.

Vapnik, V. N. *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg, 1995.

Wäldchen, S., MacDonald, J., Hauch, S., and Kutyniok, G. The computational complexity of understanding binary classifier decisions. *J. Artif. Intell. Res.*, 70:351–387, 2021.

Wang, J., Neskovic, P., and Cooper, L. N. Pattern classification via single spheres. In *proceedings of 8th International Conference on Discovery Science*, volume 3735 of *Lecture Notes in Computer Science*, pp. 241–252. Springer, 2005.

Wang, J., Neskovic, P., and Cooper, L. N. Bayes classification based on minimum bounding spheres. *Neurocomputing*, 70(4-6):801–808, 2007.

Wang, Z., Zhang, W., Liu, N., and Wang, J. Scalable rule-based representation learning for interpretable classification. In *Proceedings of the Thirty-Fourth Conference on Neural Information Processing Systems (NeurIPS)*, pp. 30479–30491, 2021.