Journal of Computer and System Sciences 119 (2021) 97–124

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Regularizing conjunctive features for classification

Pablo Barceló [a,b,*], Alexander Baumgartner [c], Victor Dalmau [d], Benny Kimelfeld [e]

[a] *Institute for Mathematical and Computational Engineering, Faculty of Engineering & School of Mathematics, Pontificia Universidad Católica de Chile, Avda. Vicuña Mackenna 4860, Macul, Santiago, Chile*
[b] *IMFD Chile, Chile*
[c] *Instituto de Ciencias de la Ingeniería, Universidad de O'Higgins, Av. Libertador Bernardo O'Higgins 611, Rancagua, Chile*
[d] *Universitat Pompeu Fabra, Roc Boronat, 138, 08018, Barcelona, Spain*
[e] *Technion – Israel Institute of Technology, Faculty of Computer Science, Technion City, Haifa, 3200003, Israel*

ARTICLE INFO

Article history:
Received 1 August 2019
Received in revised form 17 August 2020
Accepted 24 January 2021
Available online 17 February 2021

Keywords:
Classification
Feature generation
Conjunctive queries
Separability
Generalized hypertree width

ABSTRACT

We consider the feature-generation task wherein we are given a database with entities labeled as positive and negative examples, and we want to find feature queries that linearly separate the two sets of examples. We focus on conjunctive feature queries, and explore two problems: (a) deciding if separating feature queries exist (separability), and (b) generating such queries when they exist. To restrict the complexity of the generated classifiers, we explore various ways of regularizing them by limiting their dimension, the number of joins in feature queries, and their generalized hypertreewidth (ghw). We show that the separability problem is tractable for bounded ghw; yet, the generation problem is not because feature queries might be too large. So, we explore a third problem: classifying new entities without necessarily generating the feature queries. Interestingly, in the case of bounded ghw we can efficiently classify without explicitly generating such queries.

1. Introduction

Context Feature engineering is a critical and resource-consuming task in the development of machine-learning solutions in general, and classifiers in particular [46,23,21]. In the framework proposed by Kimelfeld and Ré [28], the general goal is to utilize the knowledge of the underlying relational database to provide automated assistance in feature engineering. One of the fundamental tasks discussed in that framework is that of *separability*—given a database with labeled examples, determine whether a class of queries (e.g., conjunctive queries) is rich enough to provide the features needed for classification; that is, is there a sequence of feature queries and a classifier that separate the examples according to their labels?

We first summarize the framework of Kimelfeld and Ré [28]. The database schema has a special unary relation of *entities* to be classified, known as the *entity schema*. A *feature query* is a query that selects entities, and a *statistic* is a vector of feature queries. Every entity in the database is then assigned a vector, where the ith entry is $+1$ if the entity is selected by the ith feature query in the statistic, and -1 otherwise. A *training database* consists of a database over the entity schema along with a *labeling function* that partitions the entities into *positive examples* and *negative examples*. A *classifier* maps every vector representing an entity into $+1$, denoting the positive class, or -1, denoting the negative class. When evaluating

* Corresponding author at: Institute for Mathematical and Computational Engineering, Faculty of Engineering & School of Mathematics, Pontificia Universidad Católica de Chile, Avda. Vicuña Mackenna 4860, Macul, Santiago, Chile.
E-mail addresses: pbarcelo@uc.cl (P. Barceló), alexander.baumgartner.x@gmail.com (A. Baumgartner), victor.dalmau@upf.edu (V. Dalmau), bennyk@cs.technion.ac.il (B. Kimelfeld).

https://doi.org/10.1016/j.jcss.2021.01.003
0022-0000/

a classifier over a database, the entities are classified into positive and negative cases by transforming each entity into a vector, via the statistic, and then applying the classifier to this vector.

As in Kimelfeld and Ré [28], we focus on features that are *Conjunctive Queries* (CQs) without constants, and on the class of *linear* classifiers. We consider the task of feature generation that aims at automatically proposing feature queries for the statistic. In the separability problem, we are given a training database, and the goal is to determine if there exist a statistic and a classifier that separate the entities according to their labeling. When separability is tractable, we also study the ability to actually produce the statistic, i.e., *feature generation*. As we shall see, determining the existence of a separating statistic does not necessarily mean that we can produce the statistic.

The separability problem is the database variant of the classic separability from Machine Learning (cf., e.g., [2,33]), except that, here, we are given a database and not numeric vectors, and we need to generate the features. The motivation is the practice of automatically generating features as queries, and particularly via joins, which is quite common [1,36,40,29]. While such features often involve *aggregate* queries over the joins, we aim to take a step forward in understanding the theoretical ground for this practice, and we begin with seeking *simple and restricted* queries that are still useful as features in the sense that they provide (approximate) separation.

Separation via simple features may appear to contrast with recent work that focuses on applying *more complex* functions—aiming to eliminate the engineering of specialized features—such as functions for aggregating database information towards statistical models [25] and neural-based feature functions that operate over raw databases [31]. This effort also includes the embedding of database tuples into a (feature) vector space [8] via graph neural networks [45,17] or more specialized tuple-to-vector techniques [34]. However, our approach complements these efforts in several senses. First, classifiers that are based on simple features can help in seeking accompanying *explainable* models, at least when their accuracy is comparable, or just for parts of the database as in the approach of *local surrogate models* [39,38]. Second, we believe that our theoretical analysis can lead to understanding the complexity inherent to the more complex functions, the quality and limitation of the techniques in use (e.g., back-propagation and gradient analysis), and the assumptions that modern approaches implicitly make on the data.

The problem The plain definition of the separability problem allows for feature queries that are arbitrarily complex. This is indeed the case in the proof of coNP-completeness of separability for linear classifiers over CQs shown in [28]. Yet, allowing complex feature queries entails several problems. The first problem is the classic risk of *overfitting*—feature queries seek information that is too specific to the examples, and hence, the learned classifier fails to generalize beyond the training database. The second problem is high *computational complexity*—feature queries might be hard to evaluate (under combined complexity). Finally, complex queries are complicated to interpret and manipulate by human engineers. Following the machine learning terminology, we name this kind of complexity restriction and reduction for learned models as *regularization* [42,41].

Our proposal In this work, we explore regularization at the level of the statistic and feature queries. We consider simplicity constraints on feature CQs and study their implication on the complexity of separability and feature generation. Natural realizations of "simplicity" might involve the *size* of the features—how many joins do they use? and the complexity of their structure—how *cyclic* are they? Hence, the restrictions we consider are twofold:

- bounding the *number of atoms* (join operators), and
- more generally, bounding the *generalized hypertree width* (ghw).

When these bounds are constant, the feature queries can be evaluated efficiently [11]. Restricting the number of atoms is an inherent artifact of common algorithms for feature generation from relational databases, which build joins incrementally up to a limited (small) depth [1,36,40,29,43]. While we are not aware of ghw playing a role in features for machine learning, it has been shown that a very small width is common in "natural" queries [7]. In addition, we explore the more traditional form of regularization—bounding the *dimension* of (i.e., the number of feature CQs in) the statistic, which motivates classic notions of regularization (viewed as the number of nonzero coefficients) [14,35]. We also study the complexity of combining the bound on the dimension with the bounds on the CQ features.

Our results on separability As said, in the absence of any restriction, the separability problem is coNP-complete even for a fixed schema [28]. If we fix the schema *and* pose a constant bound on the number of atoms, then there is only a polynomial number of possible feature queries up to equivalence; in that case, the statistic that consists of all feature CQs (up to equivalence) is itself a separating statistic, if any such statistic exists. In particular, both separability and feature generation become tractable. We show that this tractability continues to hold when the schema is not necessarily fixed, but rather, we keep fixed just the maximal arity of the relations. It remains open whether just bounding the number of atoms in each feature query (and not fixing the schema or its maximal arity) suffices to solve separability in polynomial time. Still, even in this case the problem is feasible by a *fixed-parameter tractable* algorithm [13] (the parameter being the arity of the schema).

When we consider the class of CQs of bounded ghw (for some bound k), we observe an interesting phenomenon: separability is solvable in polynomial time. And yet, we cannot necessarily generate the separating statistic (when it exists),

Table 1
Selected complexity results for the separability problem. We assume that the schema is fixed.

Problem	$\mathcal{L} = \mathbf{CQ}$	$\mathcal{L} = \mathbf{CQ}[m]$	$\mathcal{L} = \mathbf{GHW}(k)$
$\mathcal{L}$-Sep	coNP-complete [28]	PTIME	PTIME
$\mathcal{L}$-Sep$[\ell]$	coNEXPTIME-complete	PTIME	EXPTIME-complete

simply because the feature queries may be too large. Interestingly, it turns out that, while we cannot generate the feature CQs of a separating statistic, we can still classify according to it! To make this formal, we define the *classification* problem: given a training database and an evaluation database (which is simply a database over the entity schema), classify the entities of the evaluation database in a way that is explainable by a learned statistic; i.e., there exists a statistic that agrees with both the training labels (over the training database) and the produced new labels (over the evaluation database). We prove that in the case of bound ghw, the classification problem is solvable in polynomial time. This result is obtained by applying techniques based on the *existential cover game* [11].

Next, we turn to investigating the complexity implications of bounding the dimension of the statistic. We first show a general polynomial-time reduction from a variant of the problem of *Query By Example* (QBE) [44,9,3]: given a database and two tables, is there a query such that the result contains all of the tuples of the first table, and none of the tuples of the second table? The reduction applies to any query language $\mathcal{L}$ that is used for both problems. Using this general reduction, we obtain complexity results about the separability problem for several classes of CQs due to known results about QBE. For other classes of CQs, we first prove their complexity in QBE and then apply our reduction to establish the complexity of separability. In particular, we prove that for every combination of positive constant bounds on the dimension of the classifier and the number of atoms per CQ, separability is NP-complete. For general CQs, the complexity rises to coNEXPTIME-completeness, and EXPTIME-completeness for bounded ghw.

Table 1 shows selected complexity results that we obtain for separability. The classes of feature queries are the one of all CQs (denoted **CQ**), the one of all CQs with at most m atoms (denoted $\mathbf{CQ}[m]$), and the one of all CQs of ghw bounded by k (denoted $\mathbf{GHW}(k)$). The computational problems for each class $\mathcal{L}$ of feature queries is that of general separability ($\mathcal{L}$-Sep) and the separability by a statistic with at most ℓ features ($\mathcal{L}$-Sep$[\ell]$). We assume that the schema is fixed, and throughout the paper, we explain the importance of this assumption, and moreover, when it is necessary. However, no such assumption is needed for the tractability of separability for bounded ghw (i.e., $\mathbf{GHW}(k)$-Sep).

Further results Our analysis so far is applied to *perfect classification*, which means that we seek a statistic and a classifier that classify the examples precisely, no errors allowed. One might wonder if our (positive and negative) complexity results are based on the perfection of the classification. This is *not* the case: most of our complexity results apply to *approximate* classification, where we are given a number $\epsilon \in [0, 1)$ and we allow an ϵ fraction of the examples to be misclassified. In particular, for the hardness results, we prove a general reduction from approximate separability to precise separability which holds for every *fixed* ϵ. We also obtain feasibility results for CQs of bounded ghw and CQs with a fixed number of atoms by revisiting the techniques we use for perfect separability.

We also study the separability problem for more expressive feature queries, in particular FO queries. We observe that FO has the *dimension-collapse property*, which means that every training database that is FO-separable is also separable by a statistics with a single FO feature. This allows us to show that FO-separability has the same complexity as the QBE problem for FO, which is known to be GI-complete [4]. We also provide a characterization based on a definability condition of when a query language has the dimension collapse property. From this we obtain that several relevant fragments of FO also have this property: most notably, the k-variable fragment of FO, for any $k \geq 1$, and the class of existential FO formulas. On the other hand, the class of CQs, the class of CQs of bounded generalized hypertreewidth, and even the existential positive FO formulas do not have such a property. In fact, we prove something stronger: All these languages have the *unbounded-dimension property*, implying that there is no bound on the number of features from the language that are needed to separate training databases.

Note that our work is restricted to the *linear* case of classification, which is commonly viewed as a classic notion for separability, at least as a baseline to compare to more expressive classifier classes (cf., e.g., [2,33]). Moreover, in some cases, such as Lemma 5.8 of Kimelfeld and Ré [28], or Lemma 5.4 of the current paper, a linear separation exists if and only if the class of CQs can distinguish between the positive and the negative examples, regardless of the classifier class; in such cases, the complexity results immediately extend to every superclass of the linear classifiers.

Organization of the paper The rest of the paper is organized as follows. We give basic notation and definitions in Section 2, and define the separability problem in Section 3. In the next three sections, we study the complexity implications of bounding the maximum number of atoms per CQ (Section 4), the generalized hypertree width (Section 5), and the dimension of the statistic (Section 6). In Section 7, we provide results for approximate separability. We discuss feature queries beyond CQs in Section 8 and conclude in Section 9.

2. Preliminaries

Databases and homomorphisms A *schema* σ is a finite set of relation symbols, each of which has an associated arity $k > 0$. A *fact* over σ is an expression of the form $R(\bar{a})$, where R is a k-ary relation symbol in σ and $\bar{a}$ is a k-tuple of elements taken from a predefined universe. A *database* D over a σ is a finite set of facts over σ. The *domain* of D, denoted $\mathsf{dom}(D)$, is the set of universe elements that occur in the facts of D. We write $|D|$ to denote the size of a reasonable encoding of D.

Let D and D' be databases over σ. A *homomorphism* from D to D' is a mapping $h : \mathsf{dom}(D) \to \mathsf{dom}(D')$ such that for each fact $R(\bar{a}) \in D$ we have that $R(h(\bar{a})) \in D'$. Here, we use the conventional notation $h(\bar{a}) := (h(a_1), \ldots, h(a_k))$. We write $D \to D'$ if there is a homomorphism from D to D'. We also write $(D, \bar{a}) \to (D', \bar{a}')$, where $\bar{a}$ and $\bar{a}'$ are tuples over $\mathsf{dom}(D)$ and $\mathsf{dom}(D')$, respectively, to denote that there is a homomorphism h from D to D' such that $h(\bar{a}) = \bar{a}'$.

Conjunctive queries We consider here conjunctive queries without constants. Formally, a *Conjunctive Query* (CQ) q over a schema σ is a First-Order (FO) formula of the form

$$\exists \bar{y} \left(R_1(\bar{x}_1) \wedge \cdots \wedge R_n(\bar{x}_n) \right), \tag{1}$$

such that the following hold: (1) For each $i \in \{1, \ldots, n\}$ we have that R_i is a k-ary relation symbol in σ and $\bar{x}_i$ is a k-tuple of variables, and (2) $\bar{y}$ is a tuple of variables from $\bar{x}_1, \ldots, \bar{x}_n$. The expressions $R_i(\bar{x}_i)$ are the *atoms* of q. We write $q(\bar{x})$ to denote that $\bar{x}$ is a sequence that consists of all *free variables* of q, i.e., the ones that do not occur in $\bar{y}$. In this work, we mainly deal with *unary* CQs, namely CQs $q(x)$ with a single free variable x.

As usual, we define the evaluation of a CQ in terms of homomorphisms. To do so, we associate with each CQ $q(\bar{x})$ a database

$$D_q = \{R_1(\bar{x}_1), \ldots, R_n(\bar{x}_n)\},$$

which consists precisely of the atoms in q, where variables are treated as elements from the universe. A homomorphism from $q(\bar{x})$ to a database D is then a homomorphism from D_q to D.

The *evaluation* of $q(\bar{x})$ over D is the set

$$q(D) := \{\bar{a} \mid (D_q, \bar{x}) \to (D, \bar{a})\}.$$

If q is unary, then we abuse notation and view $q(D)$ as a set of *elements* rather than unary *tuples*.

When there is no risk of ambiguity, we identify q with D_q; e.g., we write $(q, \bar{x}) \to (D, \bar{a})$ instead of $(D_q, \bar{x}) \to (D, \bar{a})$.

Linear classifiers A *classifier* is a function $\mathcal{H} : \{1, -1\}^n \to \{1, -1\}$, where $n > 0$ is the *arity*. In this paper, we restrict the discussion to the class of *linear* classifiers. Recall that a tuple $\bar{w} = (w_0, w_1, \ldots, w_n)$ of real numbers defines a linear classifier $\Lambda_{\bar{w}}$ in the following way. For $(b_1, \ldots, b_n) \in \{1, -1\}^n$ we have

$$\Lambda_{\bar{w}}(b_1, \ldots, b_n) := \begin{cases} 1 & \text{if } \sum_{1 \leq i \leq n} w_i b_i \geq w_0, \\ -1 & \text{otherwise.} \end{cases}$$

We view a sequence $\langle (\bar{b}_1, y_1), \ldots, (\bar{b}_m, y_m) \rangle$ of vectors in $\{1, -1\}^{n+1}$ as a collection of *examples*, consisting of *positive examples* (where $y_i = 1$) and *negative examples* (where $y_i = -1$). As a shorthand notation, we write such a sequence as $(\bar{b}_i, y_i)_{i=1}^m$ and refer to it as a *training collection*. The training collection $(\bar{b}_i, y_i)_{i=1}^m$ is *linearly separable* if there is a linear classifier $\Lambda_{\bar{w}}$ such that $\Lambda_{\bar{w}}(\bar{b}_i) = y_i$ for all $i \in \{1, \ldots, m\}$; in this case, we say that $\Lambda_{\bar{w}}$ *linearly separates* $(\bar{b}_i, y_i)_{i=1}^m$.

3. The separability problem

Our investigation is in the context of the classification framework introduced by Kimelfeld and Ré [28], which recall next.

The framework An *entity schema* is a schema that includes a distinguished unary relation symbol η used to represent *entities*. To improve readability, we refer to an entity schema simply by σ and denote the corresponding entity symbol by η_σ (but if σ is clear from the context, we simply write η). Let D be a database over an entity schema σ. An *entity* of D is a constant a such that $\eta(a) \in D$. We denote by $\eta(D)$ the set of entities of D.

In this work, a *feature query* is a unary CQ $q(x)$ over an entity schema σ. We are interested in the set of entities selected by $q(x)$ over a database D of schema σ. Hence, without loss of generality, we assume that the atom $\eta(x)$ is always present in feature queries $q(x)$, and therefore it holds that $q(D) \subseteq \eta(D)$. We denote by $\mathbb{1}_{q(D)} : \eta(D) \to \{1, -1\}$ the *indicator function* defined by $q(D)$ over $\eta(D)$; that is, for each $e \in \eta(D)$ we have that $\mathbb{1}_{q(D)}(e) = 1$ if $e \in q(D)$, and $\mathbb{1}_{q(D)}(e) = -1$ otherwise.

A *statistic* over an entity schema σ is a sequence $\Pi = (q_1, \ldots, q_n)$ of feature queries over σ. If D is a database, then we define the mapping $\Pi^D : \eta(D) \to \{1, -1\}^n$ as follows for all entities $e \in \eta(D)$:

$$\Pi^D(e) := \left(\mathbb{1}_{q_1(D)}(e), \ldots, \mathbb{1}_{q_n(D)}(e) \right).$$

Card

id	number	issued
c1	100	Chile
c2	101	USA
c3	102	Spain

Transaction

id	card_number	place
t1	100	Chile
t2	100	Brazil
t3	101	Chile

η

att	λ
c1	-1
c2	1
c3	-1

Fig. 1. The training database (D, λ) from Example 3.1.

A *labeling* λ of a database D over entity schema σ is a function $\lambda : \eta(D) \to \{1, -1\}$ that partitions the set of entities into:

- the set $\{e \in \eta(D) \mid \lambda(e) = 1\}$ of *positive examples*, and
- the set $\{e \in \eta(D) \mid \lambda(e) = -1\}$ of *negative examples*.

A *training database* over σ is a pair (D, λ), where D is a database over σ and λ is a labeling of D.

Definition 3.1 *($\mathcal{L}$-separability).* Let $\mathcal{L}$ be a class of queries and (D, λ) a training database. Then (D, λ) is *$\mathcal{L}$-separable* if there is a statistic $\Pi = (q_1, \ldots, q_n)$ such that each q_i is in $\mathcal{L}$ and $(\Pi^D(e), \lambda(e))_{e \in \eta(D)}$ is linearly separable.

In other words, (D, λ) is $\mathcal{L}$-separable if there is a statistic Π such that each feature query $q \in \Pi$ is in $\mathcal{L}$ and a linear classifier $\Lambda_{\bar{w}}$ that satisfies

$$\Lambda_{\bar{w}}(\Pi^D(e)) = \lambda(e), \qquad \text{for every } e \in \eta(D).$$

In this case, we say that $(\Pi, \Lambda_{\bar{w}})$ *$\mathcal{L}$-separates* (D, λ); or simply that Π is a statistic that $\mathcal{L}$-separates (D, λ) if $\Lambda_{\bar{w}}$ is irrelevant.

Example 3.1 *(Motivated by Example 3.1 in [28]).* Consider an entity schema σ that consists of binary relation symbol `Card`, ternary relation symbol `Transaction`, and unary relation symbol η that represents entities. We have a training database (D, λ) over σ as shown in Fig. 1. The idea is that `Card` collects information about the number and place of issue of credit cards, while `Transaction` collects information about ids of transactions, together where the credit card used and the location.

Then (D, λ) is **CQ**-separable by the statistics $\Pi = (q_1(x), q_2(x))$, for

- $q_1(x) := \exists y \exists z(\texttt{Card}(x, y, z) \wedge \texttt{Transaction}(x', y, z))$, and
- $q_2(x) := \exists y \exists z(\texttt{Card}(x, y, z) \wedge \texttt{Transaction}(x', y, z'))$.

In fact, observe that

$$\Pi^D = \begin{pmatrix} \mathbb{1}_{q_1(D)}(c1) & \mathbb{1}_{q_2(D)}(c1) \\ \mathbb{1}_{q_1(D)}(c2) & \mathbb{1}_{q_2(D)}(c2) \\ \mathbb{1}_{q_1(D)}(c3) & \mathbb{1}_{q_2(D)}(c3) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ -1 & 1 \\ -1 & -1 \end{pmatrix}$$

Let The question is then whether the inequalities below have a solution $w_0, w_1, w_2 \in \mathbb{R}$: such that

$$\begin{aligned} w_1 \mathbb{1}_{q_1(D)}(c1) + w_2 \mathbb{1}_{q_2(D)}(c1) &< w_0 && (\text{since } \eta(c_1) = -1) \\ w_1 \mathbb{1}_{q_1(D)}(c2) + w_2 \mathbb{1}_{q_2(D)}(c2) &\geq w_0 && (\text{since } \eta(c_2) = 1) \\ w_1 \mathbb{1}_{q_1(D)}(c3) + w_2 \mathbb{1}_{q_2(D)}(c3) &< w_0 && (\text{since } \eta(c_3) = -1). \end{aligned}$$

We can readily observe that this is the case by choosing $w_0 = w_2 = 1$ and $w_1 = -1$. □

The separability problem This paper focuses on the *$\mathcal{L}$-separability problem*, or $\mathcal{L}$-Sep for short, for a class $\mathcal{L}$ of queries (usually CQs). Originally proposed in [28], in the problem $\mathcal{L}$-Sep the goal is to determine the existence of a separating statistic Π in $\mathcal{L}$. It is formally defined as follows.

Problem:	$\mathcal{L}$-Sep
Input:	A training database (D, λ)
Question:	Is (D, λ) $\mathcal{L}$-separable?

The following is known about the complexity of the $\mathcal{L}$-Sep problem, when $\mathcal{L}$ is the class **CQ** of all CQs.

Theorem 3.1. [28] *The problem* **CQ**-Sep *is* coNP*-complete. The lower bound holds even if the schema consists of a single binary relation R and the distinguished symbol η.*

Notice that **CQ**-Sep is not in the class NP, as not necessarily a pair (D, λ) that is **CQ**-separable can be separated by a statistic of polynomial size composed by CQs. Theorem 3.1 states, in turn, that non-**CQ**-separability always admits a polynomial size witness. We provide a sketch of how the upper bound is obtained as it is instructive for some of the results later presented in the paper. First, notice We use the following characterization of **CQ**-separability which is implicit in [28].

Lemma 3.2. *The following statements are equivalent for all training databases (D, λ).*

1. *(D, λ) is* **CQ***-separable.*
2. *There are no entities $e, e' \in \eta(D)$ such that $\lambda(e) \neq \lambda(e')$, and yet $e \in q(D) \Leftrightarrow e' \in q(D)$ for all $q(x) \in$* **CQ***.*

Notice that condition (2) is equivalent to stating that there are no entities $e, e' \in \eta(D)$ such that $\lambda(e) \neq \lambda(e')$, and yet $(D, e) \to (D, e')$ and $(D, e') \to (D, e)$. Hence, to check if (D, λ) is *not* **CQ**-separable we can apply the following NP-algorithm:

- Guess entities $e, e' \in \eta(D)$ such that $\lambda(e) \neq \lambda(e')$.
- Guess a homomorphisms from (D, e) to (D, e'), and viceversa.

It follows that **CQ**-Sep is in coNP.

Version of the separability problem studied in the paper We study the complexity of $\mathcal{L}$-Sep for subfamilies $\mathcal{L}$ enforcing various natural restrictions on the feature CQs. In addition, we consider two regularization variants of the separability problem:

- The *dimension* of (i.e., the number of features in) Π is bounded by a constant ℓ. We denote this variant by $\mathcal{L}$-Sep$[\ell]$.
- The dimension is not bounded, but rather is given as input. We denote this variant by $\mathcal{L}$-Sep$[*]$.

Hence, we have three variants of the separability problem, namely $\mathcal{L}$-Sep, $\mathcal{L}$-Sep$[\ell]$, and $\mathcal{L}$-Sep$[*]$.

4. Bounded number of feature atoms

As we will see next, one can overcome the high complexity of separability (and related problems), at least under the yardstick of *parameterized complexity* [13], by fixing the number of atoms allowed in feature CQs.

For every fixed $m \geq 1$, we denote by **CQ**$[m]$ the class of CQs with at most m atoms (not counting atom $\eta(x)$ which we assume appears in every feature query $q(x)$). The following simple observation allows us to obtain a better understanding of the complexity of the separability problem when restricted to feature queries in **CQ**$[m]$.

Proposition 4.1. *For every fixed $m \geq 1$, there is an algorithm that determines if a given training database (D, λ) is* **CQ**$[m]$*-separable, and if so, constructs a pair $(\Pi, \Lambda_{\bar{w}})$ that* **CQ**$[m]$*-separates (D, λ). The running time of the algorithm is bounded by $|D|^c \cdot 2^{s(k)}$ for a constant $c \geq 1$ and polynomial $s : \mathbb{N} \to \mathbb{N}$, where $k \geq 1$ is the maximal arity of a relation in D.*

Proof. Observe that (D, λ) is **CQ**$[m]$-separable iff it is separable by the statistic Π that contains all feature queries $q(x)$ in **CQ**$[m]$ that mention only relation symbols that appear in D. Let us denote by r the number of relation symbols in Σ that appear in D. The number of different feature CQs in Π is then bounded by

$$r^m \cdot (mk)^{mk} \;=\; r^m \cdot 2^{m \log m \, k \log k},$$

corresponding to the number of ways in which one can choose the relation symbols used in the atoms of a CQ in Π and the disposition of at most mk variables in such atoms. In particular, since m is fixed, the statistic Π can be constructed in time $r^m \cdot 2^{p(k)}$, for some polynomial $p : \mathbb{N} \to \mathbb{N}$.

Now, for each CQ $q(x)$ in Π, we can compute $q(D)$ in time $O(|D|^m)$, and thus the indicator function $\mathbb{1}_{q(D)} : \eta(D) \to \{1, -1\}$ can be computed in time $O(|D|^{m+1})$. Hence, the set of tuples of the form $\Pi^D(e)$, for $e \in \eta(D)$, can be computed in time $O(|D|^{m+1} \cdot r^m \cdot 2^{p(k)}) = O(|D|^{2m+1} \cdot 2^{p(k)})$, which is $|D|^{2m+1} \cdot 2^{p'(k)}$ for some polynomial $p' : \mathbb{N} \to \mathbb{N}$.

Finally, we need to determine whether $(\Pi^D(e), \lambda(e))_{e\in\eta(D)}$ is indeed linearly classifiable. Recall that linear separability can be solved in polynomial time, by a reduction to the problem of finding a solution to a linear program (which is known to be tractable by a landmark result in combinatorial optimization [27,24]). This procedure also finds a linear classifier $\Lambda_{\bar{w}}$ that separates the training collection $(\Pi^D(e), \lambda(e))_{e\in\eta(D)}$ in case it exists. Thus, checking if (D, λ) is **CQ**$[m]$-separable, and, if so, computing a pair $(\Pi, \Lambda_{\bar{w}})$ that **CQ**$[m]$-separates (D, λ), can be done in time $|D|^c \cdot 2^{s(k)}$ for a constant $c \geq 1$ and polynomial $s : \mathbb{N} \to \mathbb{N}$. This concludes the proof. □

From Proposition 4.1, the problem **CQ**$[m]$-Sep can be solved in time $|D|^{O(1)} \cdot f(k)$, for a computable function $f : \mathbb{N} \to \mathbb{N}$, where k is the maximal arity of a relation symbol mentioned in D. In the terminology of parameterized complexity, this means that the problem is *Fixed-Parameter Tractable* (FPT), with the parameter being the maximum arity k of a relation symbol in the schema (or simply the *arity of the schema* from now on). Summing up:

Corollary 4.2. *For all fixed $m \geq 1$, the problem* **CQ**$[m]$-Sep *is* FPT *with the parameter being the arity of the schema.*

The restriction on the number of atoms allowed in statistics is necessary for obtaining the positive result stated in Corollary 4.2. In fact, Theorem 3.1 states that **CQ**-Sep is coNP-hard even if the schema is of a fixed arity; hence, the problem cannot be FPT if the parameter is the arity of the schema (assuming PTIME $\neq$ NP).

Recall from Theorem 3.1 that **CQ**-Sep, and hence also **CQ**$[m]$-Sep, can be solved in coNP. It remains an interesting open problem whether **CQ**$[m]$-Sep is NP-hard for some fixed $m \geq 1$, or can be solved in polynomial time. Nevertheless, there is a way to restrict the problem in order to ensure tractability: bounding the arity of the schema by a fixed constant. As explained in the proof of Proposition 4.1, the implication of this restriction is that the number of different feature CQs that one can form in this case (up to equivalence) is polynomial in the size of the input. Still, we can do better than fixing the arity, since the argument remains valid if we assume only that the maximal *number of occurrences per variable* in the feature CQs is bounded by a constant. Formally, for fixed $m, p \geq 1$ let **CQ**$[m, p]$ be the class of CQs with at most m atoms and in which each variable occurs at most p times. Then:

Proposition 4.3. **CQ**$[m, p]$-Sep *can be solved in polynomial time, for every fixed m and p.*

Importantly, the results stated in Corollary 4.2 and Proposition 4.3 are obtained via a constructive proof that allows to perform the following tasks with the same tractability guarantees, if the input (D, λ) is indeed **CQ**$[m]$-separable.

- *Feature generation:* Construct a pair $(\Pi, \Lambda_{\bar{w}})$ that **CQ**$[m]$-separates the training database (D, λ).
- *Classification:* Apply $(\Pi, \Lambda_{\bar{w}})$ to a given evaluation database for performing the actual classification.

As shown next, things become more complicated if, instead of the number of atoms, we bound the *generalized hypertree-width* of feature queries.

5. Bounded generalized hypertree-width

In this section, we investigate the complexity implications of regularizing statistics by bounding the generalized hypertree-width of the feature CQs.

5.1. Background

Hypertree-width We start by introducing the classes of CQs of bounded *generalized hypertree-width* [16] (also known as *coverwidth* [11]). We adopt the definition of Chen and Dalmau [11], which better suits non-Boolean queries. A *tree decomposition* of a CQ $q = \exists \bar{y} \bigwedge_{1\leq i\leq n} R_i(\bar{x}_i)$ is a pair (T, χ), where T is a tree and χ is a mapping that assigns a subset of the existentially quantified variables in $\bar{y}$ to each node $t \in T$, such that:

1. For all $1 \leq i \leq m$, the variables in $\bar{x}_i \cap \bar{y}$ are contained in $\chi(t)$, for some $t \in T$.
2. For all variables y in $\bar{y}$, the node set $\{t \in T \mid y \in \chi(t)\}$ induces a connected subtree of T.

The *width* of node t in (T, χ) is the minimal size of an $I \subseteq \{1, \ldots, m\}$ such that $\bigcup_{i\in I} \bar{x}_i$ covers $\chi(t)$. The width of (T, χ) is the maximal width of the nodes of T. The *generalized hypertree-width* (*ghw* for short) of q is the minimum width of its tree decompositions.

For a fixed k, we denote by **GHW**(k) the class of CQs of ghw at most k. In contrast to the case of general CQs, the evaluation problem for CQs in **GHW**(k) can be solved in polynomial time [15]. Notice that each CQ in **CQ**$[k]$ is also in **GHW**(k), but not viceversa.

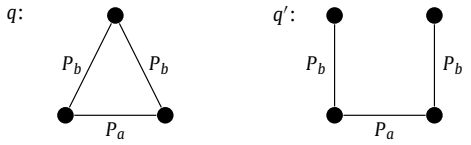

Fig. 2. The CQs q and q' from Example 5.1.

The existential cover game There is a link between the evaluation of CQs in **GHW**(k) and a version of the pebble game, known as *existential cover game* [11], that we recall below. The existential k-cover game (for k a natural number) is played by *Spoiler* and *Duplicator* on pairs $(D, \bar{a})$ and $(D', \bar{b})$, where D and D' are databases and $\bar{a}$ and $\bar{b}$ are n-ary ($n \geq 0$) tuples over $\mathsf{dom}(D)$ and $\mathsf{dom}(D')$, respectively. In each round of the game, Spoiler places (resp., removes) a pebble on (resp., from) an element of $\mathsf{dom}(D)$, and Duplicator responds by placing (resp., removing) its corresponding pebble on an element of (resp., from) $\mathsf{dom}(D')$. The number of pebbles is not bounded, but Spoiler is constrained as follows: At any round p of the game, if $c_1, \ldots, c_\ell$ ($\ell \leq p$) are the elements marked by Spoiler's pebbles in D, there must be a set of at most k facts in D that contain all such elements (this is why the game is called k-cover, as pebbled elements are *covered* by no more than k facts).

Duplicator wins if she has a *winning strategy*, that is, she can indefinitely continue playing the game in such a way that after each round, if $c_1, \ldots, c_\ell$ are the elements that are marked by Spoiler's pebbles in D and $d_1, \ldots, d_\ell$ are the elements marked by the corresponding pebbles of Duplicator in D', then

$$\big((c_1, \ldots, c_\ell, \bar{a}), (d_1, \ldots, d_\ell, \bar{b})\big)$$

is a *partial homomorphism* from D to D'. That is, for every atom $R(\bar{c}) \in D$, where each element c of $\bar{c}$ appears in $(c_1, \ldots, c_\ell, \bar{a})$, it is the case that $R(\bar{d}) \in D'$, where $\bar{d}$ is the tuple obtained from $\bar{c}$ by replacing each element c of $\bar{c}$ by its corresponding element d in $(d_1, \ldots, d_\ell, \bar{b})$. We write $(D, \bar{a}) \to_k (D', \bar{b})$ if Duplicator wins.

Notice that $\to_k$ "approximates" $\to$ as follows:

$$\to \subsetneq \cdots \subsetneq \to_{k+1} \subsetneq \to_k \subsetneq \cdots \subsetneq \to_1 \qquad (k \geq 1).$$

Example 5.1. Fig. 2 shows two CQs q and q'. The schema consists of binary relation symbols P_a and P_b. Nodes represent variables, and an edge labeled P_a between x and y represents the presence of atoms $P_a(x, y)$ and $P_a(y, x)$. (Same for P_b). All variables are existentially quantified. Clearly, $q \not\to q'$. In addition, it can easily be established that $q \to_1 q'$. □

The approximations provided by $\to_k$ over $\to$ are convenient complexity-wise: Checking whether $(D, \bar{a}) \to (D', \bar{b})$ is NP-complete, but $(D, \bar{a}) \to_k (D', \bar{b})$ can be solved efficiently (as long as k is fixed) by applying standard greatest fixed-point algorithms based on local consistency notions developed in the area of constraint satisfaction.

Proposition 5.1. [11] *For all fixed $k \geq 1$, whether $(D, \bar{a}) \to_k (D', \bar{b})$ can be determined in polynomial time.*

Moreover, there is a close connection between $\to_k$ and the evaluation of CQs in **GHW**(k).

Proposition 5.2. [11] *$(D, \bar{a}) \to_k (D', \bar{b})$ if and only if for every CQ $q(\bar{x})$ in* **GHW**(k) *we have that*

$$(q, \bar{x}) \to (D, \bar{a}) \implies (q, \bar{x}) \to (D', \bar{b}).$$

In particular, for all CQs $q(\bar{x})$ in **GHW**(k)*, databases D, and tuples $\bar{a}$, it holds that $\bar{a} \in q(D)$ if and only if $(q, \bar{x}) \to_k (D, \bar{a})$.*

5.2. Separability

In contrast to the case of arbitrary CQs, the separability problem for the classes of CQs of bounded ghw is tractable. We prove this result by applying techniques based on the existential cover game.

Theorem 5.3. *For all fixed $k \geq 1$, the problem* **GHW**(k)-Sep *is solvable in polynomial time.*

The proof is based on the following lemma which corresponds to the relativization of Lemma 3.2 from **CQ** to **GHW**(k).

Lemma 5.4. *The following statements are equivalent for all training databases (D, λ).*

1. *(D, λ) is* **GHW**(k)*-separable.*

2. *There are no entities* $e, e' \in \eta(D)$ *such that* $\lambda(e) \neq \lambda(e')$, *and yet* $e \in q(D) \Leftrightarrow e' \in q(D)$ *for all* $q(x) \in \mathbf{GHW}(k)$.

Proof. The fact that $1 \to 2$ is straightforward. We now prove that $2 \to 1$. For each $e \in \eta(D)$, we define a query

$$q_e(x) := \bigwedge_{e' \in \eta(D)} q_e^{e'}(x), \tag{2}$$

where $q_e^{e'}(x) = q(x)$ is an arbitrary CQ in $\mathbf{GHW}(k)$ such that $e \in q(D)$ and $e' \notin q(D)$—if such $q(x)$ exists at all—and it is $\eta(x)$ otherwise. Then $q_e(x)$ can be reformulated as an equivalent CQ in $\mathbf{GHW}(k)$. This is because each conjunct in $q_e(x)$ is in $\mathbf{GHW}(k)$, and $\mathbf{GHW}(k)$ is closed under taking conjunctions (see, e.g., [6]).

We denote by $\preceq$ the binary relation over $\eta(D)$ such that $e \preceq e'$ iff $e' \in q_e(D)$. It is easy to see that $\preceq$ is reflexive and transitive, that is, it is a preorder. Recall that an *equivalence class* of $\preceq$ over $\eta(D)$ is an equivalence class of the equivalence relation "$e \preceq e'$ and $e' \preceq e$". We overload notation and write $E \preceq F$, for equivalence classes E, F over $\eta(D)$ defined by $\preceq$, iff there are elements $e \in E$ and $f \in F$ such that $e \preceq f$. Since $\preceq$ is a partial order, there is a topological sort of such equivalence classes with respect to $\preceq$. Let $E_1, E_2, \ldots, E_m$ be one such a topological sort.

For each E_i, we select an arbitrary entity $e_i \in E_i$. It is not hard to see then that the following hold for each $i \in \{1, \ldots, m\}$ and entity $e \in E_i$:

- $e \in q_{e_i}(D)$, and
- $e \notin q_{e_j}(D)$ for each $j \in \{1, \ldots, m\}$ with $i < j$.

It follows from Kimelfeld and Ré [28] that these properties imply that the statistics $\Pi = (q_{e_1}, \ldots, q_{e_m})$ separates (D, λ). Since each q_{e_i} can be reformulated as an equivalent CQ in $\mathbf{GHW}(k)$, we conclude that (D, λ) is $\mathbf{GHW}(k)$-separable. □

Proposition 5.2 establishes that the condition of Lemma 5.4, stating that for all $q(x) \in \mathbf{GHW}(k)$ it is the case that $e \in q(D) \Leftrightarrow e' \in q(D)$, is equivalent to saying that

$$(D, e) \to_k (D, e') \text{ and } (D, e') \to_k (D, e).$$

Hence, the following test checks for $\mathbf{GHW}(k)$-separability.

Test:	$\mathbf{GHW}(k)$-separability
Input:	A training database (D, λ)
Condition:	Accept if $(D, e) \not\to_k (D, e')$ or $(D, e') \not\to_k (D, e)$, for all $e, e' \in \eta(D)$ with $\lambda(e) \neq \lambda(e')$

Proposition 5.5. *A training database* (D, λ) *is* $\mathbf{GHW}(k)$*-separable iff the* $\mathbf{GHW}(k)$*-separability test accepts* (D, λ).

From Proposition 5.1, the $\mathbf{GHW}(k)$-separability test can be performed in polynomial time, which yields Theorem 5.3.

While Theorem 5.3 establishes the tractability of $\mathbf{GHW}(k)$-Sep, the proof is *not* constructive, that is, it does not show how to efficiently construct a statistic that realizes $\mathbf{GHW}(k)$-separability. As shown next, this is not coincidental: separability and feature generation behave differently for $\mathbf{GHW}(k)$.

5.3. Feature generation

We now look at the problem of generating a statistics that $\mathbf{GHW}(k)$-separates a training database (D, λ). It follows from Chen and Dalmau [11] that there is an exponential time algorithm that takes as input an entity $e \in \eta(D)$ and constructs a CQ $q'_e(x)$ in $\mathbf{GHW}(k)$ that is equivalent to $q_e(x)$, where $q_e(x)$ is as defined in Equation (2) in the proof of Lemma 5.4. On the other hand, Lemma 5.4 states that if (D, λ) is $\mathbf{GHW}(k)$-separable, then it is separable by a statistic that contains only queries of the form $q'_e(x)$ for $e \in \eta(D)$. Therefore, if (D, λ) is $\mathbf{GHW}(k)$-separable, then there exists a statistic Π with polynomially many features, each of which is of at most exponential size, such that Π $\mathbf{GHW}(k)$-separates (D, λ). This statistic Π can be constructed in exponential time from (D, λ). Summing up:

Proposition 5.6. *For all fixed k, there is an exponential-time algorithm that determines whether a given training database* (D, λ) *is* $\mathbf{GHW}(k)$*-separable, and if so, generates a statistic* Π *that:*

- $\mathbf{GHW}(k)$*-separates* (D, λ)*;*
- *has a dimension linear in the number of entities in* $\eta(D)$*;*
- *consists of CQs of size at most exponential in that of D.*

As it turns out, the size of statistic Π in Proposition 5.6 is essentially optimal.

Theorem 5.7. *Let $k \geq 1$ be fixed. For all $n, m \geq 1$ there is a training database (D, λ) with $|D| = O(n+m)$ such that:*

- *(D, λ) is $\mathbf{GHW}(k)$-separable.*
- *For all statistics $\Pi = (q_1, \ldots, q_p)$ that linearly separate (D, λ), it is the case that (a) $p \geq m$, and (b) at least one of the q_is, for $i \in \{1, \ldots, p\}$, has $\Omega(2^n)$ atoms.*

Before proving the theorem, we introduce an important result. Recall from Proposition 5.2 that, for every $k \geq 1$, if $(D, \bar{a}) \not\to_k (D', \bar{b})$, where D and D' are databases and $\bar{a}$ and $\bar{b}$ are n-ary tuples over $\mathsf{dom}(D)$ and $\mathsf{dom}(D')$, respectively, then there is a CQ $q(\bar{x}) \in \mathbf{GHW}(k)$ such that $D \models q(\bar{a})$ but $D' \not\models q(\bar{b})$. In the case when $D = D'$, we call a CQ $q(\bar{x})$ satisfying such conditions a $\mathbf{GHW}(k)$*-separator for* $(D, \bar{a}, \bar{b})$. The following result states that $\mathbf{GHW}(k)$-separators are, in some cases, necessarily of exponential size. Moreover, this holds even when $\bar{a}$ and $\bar{b}$ consist of single elements.

Proposition 5.8 *(Implicit in [30]). Fix $k \geq 1$. There is a family of databases D_n and elements $a_n, b_n \in \mathsf{dom}(D_n)$, for $n \geq 0$, such that*

- *$(D_n, a_n) \not\to_k (D_n, b_n)$,*
- *the size of D_n is polynomial in n, and*
- *each $\mathbf{GHW}(k)$-separator for (D_n, a_n, b_n) has $\Omega(2^n)$ atoms.*

We would like to remark that while this result is not explicitly stated in [30], its proof can be almost directly extracted from the proof of the main result in such a paper; namely, that checking, for a fixed $k \geq 1$, whether $(D, a) \to_k (D_b)$ is a PTIME-complete problem.

We now prove Theorem 5.7. The intuition is as follows. For each $n, m \geq 1$, we start by taking the tuple (D_n, a_n, b_n) as given in the statement of Proposition 5.8. We then extend D_n with a distinguished set of $m-1$ constants interpreted in a very specific way over the schema. The resulting database is denoted D_n^*. We need to prove two things. First, that each statistics Π formed by CQs in $\mathbf{GHW}(k)$ that linearly separates (D_n^*, λ) must have at least m features. Second, that at least one CQ in Π must be of exponential size in n. The first part follows more or less straightforwardly from the way in which distinguished constants are interpreted over the schema: the statistics Π needs to have a different feature CQ for each distinguished constant, as otherwise it would not be able to separate the given entities. One more feature CQ is then needed to separate a_n from b_n. For the second part, we show that if Π separates (D_n^*, λ) then there must be at least one feature CQ from $\mathbf{GHW}(k)$ in Π that provides a $\mathbf{GHW}(k)$-separator for (D_n, a_n, b_n). By assumption, this feature CQ must be of at least exponential size in n.

Proof of Theorem 5.7. Fix $n, m \geq 1$, and let $(D_n, \{a_n\}, \{b_n\})$ be as given in the statement of Proposition 5.8. By assumption, there is a $\mathbf{GHW}(k)$-separator $q_n(x)$ for $(D_n, \{a_n\}, \{b_n\})$.

Without loss of generality, we can assume that $(D_n, b_n) \to (D_n, a_n)$; if not, we build a new database D'_n that is obtained by taking two disjoint copies D_n^1 and D_n^2 of D_n, and then fusing the elements that correspond to a_n and b_n in D_n^1 and D_n^2, respectively, into a single element a'_n. It is clear then that if b'_n is the element that corresponds to b_n in D_n^1, then $(D'_n, b'_n) \to (D'_n, a'_n)$. In addition, if (D_n, a_n, b_n) has a $\mathbf{GHW}(k)$-separator then so does $(D'_n, \{a'_n\}, \{b'_n\})$. Finally, as we observe next, for each $\mathbf{GHW}(k)$-separator $q'(x)$ of (D'_n, a'_n, b'_n) there is a $\mathbf{GHW}(k)$-separator $q(x)$ of $(D_n, \{a_n\}, \{b_n\})$ such that the atoms of $q(x)$ are a subset of the atoms in $q'(x)$. This implies that each $\mathbf{GHW}(k)$-separator for (D'_n, a'_n, b'_n) must also have $\Omega(2^n)$ atoms.

Claim 1. *For each $\mathbf{GHW}(k)$-separator $q'(x)$ of (D'_n, a'_n, b'_n) there is a $\mathbf{GHW}(k)$-separator $q(x)$ of (D_n, a_n, b_n) such that the atoms of $q(x)$ are a subset of the atoms in $q'(x)$.*

Proof. Consider an arbitrary $\mathbf{GHW}(k)$-separator $q'(x)$ for (D'_n, a'_n, b'_n). Then there is homomorphism h' from q' to D'_n such that $h'(x) = a'_n$, but no homomorphism g' from q' to D'_n such that $g'(x) = b'_n$. By definition, h' must satisfy two properties.

- At least some variable $y \neq x$ from q' satisfies that $h'(y) \in D_n^1$ (otherwise, there is also a homomorphism g' from q' to D'_n such that $g'(x) = b'_n$, which is a contradiction).
- Each atom $R(\bar{z})$ in q' is mapped to D_n^1 or D_n^2 by h' (since D_1^n and D_2^n only share the element a'_n).

Let q_1 and q_2 be the queries that are obtained by taking precisely those atoms in q' that are mapped by h' to D_n^1 and D_n^2, respectively. Notice then that there is a homomorphism g' from q_2 to D'_n such that $g'(x) = b'_n$ (by simply mimicking the result of h' in D_n^1). In addition, $q_1(x)$ satisfies the following properties.

- First, $q_1 \in \mathbf{GHW}(k)$. This follows by taking a tree decomposition (T, χ') of $q'(x)$ whose width is k, turning each bag $\chi'(t)$, for $t \in T$, into a new bag $\chi(t)$ by removing all the variables not in q_1, and then noticing that for any set A of atoms of q' that covers a bag $\chi'(t)$, for $t \in T$, the restriction of A to the atoms that only mention variables in q_1 also covers $\chi(t)$.

- Second, $q_1(x)$ is a **GHW**(k)-separator for (D_n, a_n, b_n). In fact, we know that h' is a homomorphism from q_1 to D_n^1 such that $h'(x) = a'_n$. Therefore, there is a homomorphism h_a from q_1 to D_n such that $h_a(x) = a_n$. In addition, $b \notin q(D_n)$. Assume otherwise, i.e., there is a homomorphism h_b from q_1 to D_n such that $h_b(x) = b_n$. Then there is also a homomorphism g from q_1 to D'_n such that $g(x) = b'_n$. It follows that the mapping f from q' to D'_n defined as $f(y) = g(y)$, if y belongs to q_1, and $f(y) = g'(y)$, otherwise, is a homomorphism from q' to D'_n such that $f(x) = b'_n$. This is a contradiction.

This finishes the proof of the claim as we can define $q(x)$ to be $q_1(x)$. □

We now continue with the proof of Theorem 5.7. Let us define a database D_n^* which extends D_n with fresh constants $b', a'_1, ..., a'_{m-1}$ and fresh facts

$$\kappa_1(a'_1), \ldots, \kappa_{m-1}(a'_{m-1}),$$

assuming that the κ_is are unary relation symbols not in the schema of D_n. We define $\eta(D_n^*) := \mathsf{dom}(D_n^*)$, and the labeling $\lambda : \eta(D_n^*) \to \{1, -1\}$ so that $\lambda(e) = 1$ if $e \in q_n(D_n) \cup \{a'_1, ..., a'_{m-1}\}$ and $\lambda(e) = -1$ otherwise. Notice then that $\lambda(a_n) = 1$ and $\lambda(b_n) = 1$.

We prove the following claim.

Claim 2. *It is the case that (D_n^*, λ) is* **GHW**(k)*-separable by a statistics with m features, but not by a statistics with $m-1$ features.*

Proof. Consider the statistic $\Pi = (\kappa_1, \ldots, \kappa_{m-1}, q_n)$, which is formed by queries in **GHW**(k). We claim that $(\Pi, \Lambda_{\bar{w}})$ separates (D_n^*, λ), where $\bar{w} = (1 - m, 1, \ldots, 1)$. In fact, notice by definition that for each $e \in \eta(D_n^*) = \mathsf{dom}(D_n^*)$ it is the case that $\Lambda_{\bar{w}}(\Pi^{D_n^*}(e)) = 1$ iff e belongs to the evaluation of at least some CQ in Π over D_n^*. We prove that the latter holds iff $\lambda(e) = 1$.

Let us first consider the elements e with $\lambda(e) = 1$. If $e \in q_n(D_n^*)$ then this is trivial. In addition, if $e = a'_i$, for $i \in \{1, \ldots, m-1\}$, then $e \in \kappa_i(D_m^*)$. Consider now an arbitrary element e with $\lambda(e) = -1$. Then by definition $e \notin \kappa_i(D_n^*)$, for each $i \in \{1, \ldots, m-1\}$. Also, $e \notin q_n(D_n^*)$ as we prove next. Assume, for the sake of contradiction, that this is not the case.

- Suppose first that $e \in \mathsf{dom}(D_n)$. Then $e \in q_n(D_n)$. This is because the restriction of D_n^* to the symbols in the schema of D_n homomorphically maps to D_n, and, thus, if $e \in q_n(D_n^*)$ we would have that $e \in q_n(D_n)$. But then $\lambda(e) = 1$ by definition, which is a contradiction.
- Suppose now that $e = b'$. Since the only atom that mentions b' in D_n^* is $\eta(b')$, and, in addition, $\eta(D_n^*) = \mathsf{dom}(D_n^*)$, we would have that any element in D_n^* belongs to $q_n^*(D_n^*)$; in particular, b_n. But then $b_n \in q_n(D_n)$ from the previous item, which is a contradiction as $\lambda(b_n) = -1$.

We prove next that (D_n^*, λ) is not **GHW**(k)-separable by a statistics with $m-1$ features. Assume to the contrary that such a statistics $\Pi = (q^1, \ldots, q^{m-1})$ exists. For each $i \in \{1, \ldots, m-1\}$ it is the case that $\lambda(a'_i) \neq \lambda(b')$, and, therefore,

$$\Lambda_{\bar{w}}(\Pi^{D_n^*}(a'_i)) \neq \Lambda_{\bar{w}}(\Pi^{D_n^*}(b')).$$

Hence, for each $i \in \{1, \ldots, m-1\}$ there is at least some $q(x)$ in Π such that $q(D_n^*) \cap \{b', a'_i\}$ is either $\{b'\}$ or $\{a'_i\}$. But $q(D_n^*) \cap \{b', a'_i\} = \{b'\}$ is not possible, as otherwise we would have $q(D_n^*) = \mathsf{dom}(D_n^*)$ by a previous observation. Hence for each $i \in \{1, \ldots, m-1\}$ there exists a $q(x)$ in Π such that $q(D_n^*) \cap \{b', a'_i\} = \{a'_i\}$. It is easy to see then that $q(x)$ must be equivalent to $\kappa_i(x)$, and hence $q(D_n^*) = \{a'_i\}$. In fact, if $q(x)$ selects a'_i then it can only be equivalent to $\kappa_i(x)$ or $\eta(x)$. But the latter is not possible as otherwise $q(D_n^*) = \mathsf{dom}(D_n^*)$.

Hence, Π is equivalent to the statistic $(\kappa_1, \ldots, \kappa_{m-1})$. For each κ_i we have that $a_n, b_n \notin \kappa_i(D_n^*)$. But then

$$\Lambda_{\bar{w}}(\Pi^{D_n^*}(a_n)) = \Lambda_{\bar{w}}(\Pi^{D_n^*}(b_n)),$$

thus contradicting the fact that Π separates (D_n^*, λ) as $\lambda(a_n) = 1$ and $\lambda(b_n) = -1$. This finishes the proof of the claim. □

Assume then that (D_n^*, λ) is **GHW**(k)-separable by a statistics Π with $p \geq m$ features. It remains to show that at least one feature query in Π has to have at least $\Omega(2^n)$ atoms. Since $(D_n, b_n) \to (D_n, a_n)$, it is also the case that $(D_n^*, b_n) \to (D_n^*, a_n)$. Hence, there must be some query $q_n^*(x)$ in Π such that $a_n \in q_n^*(D_n^*)$ but $b_n \notin q_n^*(D_n^*)$; otherwise Π could not separate the element a_n with $\lambda(a_n) = 1$ from the element b_n with $\lambda(b_n) = -1$. We can then establish the following.

Claim 3. *If we remove from $q_n^*(x)$ all atoms of the form $\eta(z)$ and $\kappa_i(z)$, for $i \in \{1, \ldots, m-1\}$, we obtain a* **GHW**(k)*-separator for (D_n, a_n, b_n).*

Proof. Since $\eta(D_n^*) = \mathsf{dom}(D_n^*)$, an atom $\eta(z)$ is equal to the trivial condition $z \in \mathsf{dom}(D_n^*)$ and can be removed. Let us assume then that

$$q_n^*(x) := \exists \bar{y} \left(\kappa_{i_1}(z_1) \wedge \cdots \wedge \kappa_{i_p}(z_p) \wedge R_1(\bar{x}_1) \wedge \cdots \wedge R_t(\bar{x}_t)\right),$$

where $1 \leq i_1, \ldots, i_p \leq m-1$ and the R_ℓs come from the schema of D_n. We know that $a_n \in q_n^*(D_n^*)$; hence $x \neq z_j$, for each $j \in \{1, \ldots, p\}$. Thus,

$$q_n^*(x) := \exists \bar{y}_0 \exists z_1, \ldots, z_m \left(\kappa_{i_1}(z_1) \wedge \cdots \wedge \kappa_{i_p}(z_p) \wedge R_1(\bar{x}_1) \wedge \cdots \wedge R_t(\bar{x}_t)\right),$$

where $\bar{y}_0$ is $\bar{y}$ without $z_1, \ldots, z_p$. By definition, for each $j \in \{1, \ldots, p\}$, the element a'_{i_j} only appears twice in D_n^*; namely, in the facts $\kappa_{i_j}(a'_{i_j})$ and $\eta(a'_{i_j})$. Therefore, z_{i_j} cannot appear in any of the atoms $R_\ell(\bar{x}_\ell)$, for $1 \leq \ell \leq t$. This means that we can safely remove each atom of the form $\kappa_{i_j}(z_{i_j})$ from q_n^*, as it is expressing a trivial condition; i.e., that there is an element in the interpretation of κ_{i_j} over D_n^*. The resulting query $q(x)$ satisfies that $q(D_n^*) = q_n^*(D_n^*)$, and, therefore, $q(x)$ is a **GHW**(k)-separator for (D_n^*, a, b). □

Therefore, the $q_n^*(x)$s must have $\Omega(2^n)$ atoms, as otherwise we would have a family of **GHW**(k)-separators for the tuples (D_n, a_n, b_n) that neither has $\Omega(2^n)$ atoms. This is a contradiction. □

We are thus faced with an apparently contradictory situation: while we can efficiently check for the existence of a statistic that **GHW**(k)-separates the input (D, λ), materializing such a statistic might be infeasible. Interestingly, for *classifying* unseen entities, this statistic does not need to be materialized—we can perform this task efficiently by applying techniques based on the existential cover game. Next, we formalize this statement and prove it.

5.4. Classification

In this section, we discuss the problem of classifying an evaluation database based on a training database, without necessarily materializing a statistic. Formally, in this problem we are given a training database (D, λ) and an evaluation database D', which is a database over the same schema as D. The goal is to classify the entities of D' according to *some* statistic and linear classifier that separate D.

Problem:	$\mathcal{L}$-Cls
Input:	An $\mathcal{L}$-separable training database (D, λ) and an evaluation database D'
Output:	A labeling λ' of D' such that there is $(\Pi, \Lambda_{\bar{w}})$ that $\mathcal{L}$-separates both (D, λ) and (D', λ')

We prove the following:

Theorem 5.9. **GHW**(k)-Cls *can be solved in polynomial time for all fixed* $k \geq 1$.

Before proving this result we provide a high-level idea of the proof. Consider an input given by (D, λ) and D'. We make use of the CQs $q_e(x)$, for $e \in \eta(D)$, as well as several concepts defined in the proof of Lemma 5.4. It can be shown that the training database (D, λ) is **GHW**(k)-separable, then it is **GHW**(k)-separable by a statistic $\Pi = (q_{e_1}(x), \ldots, q_{e_m}(x))$, where $e_i \in E_i$ for an arbitrary topological sort $E_1, \ldots, E_m$ of the equivalence classes defined by $\preceq$. We can then construct in polynomial time a linear classifier $\Lambda_{\bar{w}}$ and a labeling λ' of $\eta(D')$ such that $(\Pi, \Lambda_{\bar{w}})$ **GHW**(k)-separates (D, λ) and (D', λ'). The crucial observation is that this step does not require building the statistic Π. To prove this we use known properties of the existential cover game.

Proof of Theorem 5.9. Consider an input for **GHW**(k)-Cls that consists of a **GHW**(k)-separable training database (D, λ) over an entity schema σ and an evaluation database D' over σ. We need to construct a labeling λ' of $\eta(D')$ such that there exists $(\Pi, \Lambda_{\bar{w}})$ that **GHW**(k)-separates both (D, λ) and (D', λ').

Let us consider again the CQs $q_e(x)$, for $e \in \eta(D)$, defined in the proof of Lemma 5.4. From the definition of $q_e(x)$ it follows that for all $e' \in \eta(D)$ we have that $e' \in q_e(D)$ iff for all CQs $q(x) \in$ **GHW**(k) it is the case that $e \in q(D)$ implies $e' \in q(D)$. In turn, from Proposition 5.2 we get that the latter holds iff $(D, e) \to_k (D, e')$. Therefore, the problem of testing whether $e' \in q_e(D)$, given $e, e' \in \eta(D)$, can be solved in polynomial time due to Proposition 5.1.

Let $E_1, \ldots, E_m$ be an arbitrary topological sort of the equivalence classes defined by $\preceq$ over $\eta(D)$. From the proof of Lemma 5.4 it follows that the training database (D, λ) is **GHW**(k)-separable by any statistic $\Pi = (q_{e_1}(x), \ldots, q_{e_m}(x))$ such that $e_i \in E_i$.

The topological sort $E_1, \ldots, E_m$ and, thus, also the elements $e_1, \ldots, e_m$, can be constructed in polynomial time from (D, λ). This is due to the fact that the relation $\preceq$ over $\eta(D)$ can be constructed efficiently (as we have already mentioned that $e \preceq e'$ iff $e' \in q_e(D)$ iff $(D, e) \to_k (D, e')$, which is decidable in polynomial time). In addition, it follows from [28] that one can construct in polynomial time a linear classifier $\Lambda_{\bar{w}}$ such that $(\Pi, \Lambda_{\bar{w}})$ separates (D, λ), without actually constructing Π, but rather just using $\preceq$.

Algorithm 1 Classification algorithm **GHW**(k)-Cls.

Require: An **GHW**(k)-separable training database (D, λ) and an evaluation database D'
1: $([e_1], \ldots, [e_m]) :=$ topological sort of the equivalence classes defined by $\rightarrow_k$ over $\eta(D)$
2: $\Lambda_{\bar{w}} = (w_0, \ldots, w_m) :=$ linear classifier such that $(\Pi, \Lambda_{\bar{w}})$ separates (D, λ), where $\Pi = (q_{e_1}(x), \ldots, q_{e_m}(x))$ ▷ It can be computed efficiently without computing Π
3: **for** each $f \in \eta(D')$ and $i \in \{1, \ldots, m\}$ **do**
4: **if** $(D, e_i) \rightarrow_k (D', f)$ **then**
5: $\mathbb{1}_{q_{e_i}(D')}(f) = 1$
6: **else**
7: $\mathbb{1}_{q_{e_i}(D')}(f) = -1$
8: **end if**
9: **end for**
10: **for** each $f \in \eta(D')$ **do**
11: **if** $\sum_{1 \leq i \leq m} w_i \cdot \mathbb{1}_{q_{e_i}(D')}(f) \geq w_0$ **then**
12: $\lambda'(f) = 1$
13: **else**
14: $\lambda'(f) = -1$
15: **end if**
16: **end for**
17: **return** $\lambda' : \eta(D') \rightarrow \{1, -1\}$

We define a labeling λ' of $\eta(D')$ such that for each $f \in \eta'(D)$ it holds that $\lambda'(f) = \Lambda_{\bar{w}}(\Pi^{D'}(f))$. Clearly, $(\Pi, \Lambda_{\bar{w}})$ **GHW**(k)-separates (D, λ) and (D', λ'). We need to show that λ' can be constructed in polynomial time, or equivalently, that given $f \in \eta(D')$ we can compute $\lambda'(f) = \Lambda_{\bar{w}}(\Pi^{D'}(f))$ in polynomial time. By definition,

$$\Lambda_{\bar{w}}(\Pi^{D'}(f)) = 1 \;\Leftrightarrow\; \sum_{1 \leq i \leq m} w_i \cdot \mathbb{1}_{q_{e_i}(D')}(f) \geq w_0,$$

assuming that $\bar{w} = (w_0, \ldots, w_m)$. The latter can be checked in polynomial time, since computing $\mathbb{1}_{q_{e_i}(D')}(f)$ boils down to checking $(D, e_i) \rightarrow_k (D', f)$. The pseudo-code of the procedure is shown in Algorithm 1. □

In summary, in this section we have established that it takes polynomial time to decide whether a given training database (D, λ) is **GHW**(k)-separable (Theorem 5.3). At the same time, it may be infeasible to actually materialize the separating statistic, since it might be too large (Theorem 5.7). Then again, to classify entities of a given evaluation database D', we do not need to materialize such a statistic, and in fact, this classification can be carried out in polynomial time (Theorem 5.9).

6. Bounding the dimension

While the separability and classification problems become tractable if we restrict to statistics formed by CQs in **GHW**(k), for each fixed $k \geq 1$, there is one aspect of such statistics that complicates its applicability: as stated in Theorem 5.7, the number of feature queries required to separate a training database (D, λ) might depend on the number of elements in $\mathsf{dom}(D)$. This problem is not exclusive to the class **GHW**(k); in fact, a similar negative result can be proved for statistics based on the general class of CQs.

To address this issue, we study the separability problem for the restricted class of statistics that allow a bounded number of features only. Recall that this problem is denoted $\mathcal{L}$-Sep[$*$], for $\mathcal{L}$ a class of CQs. The input consists of a training database (D, λ) and an integer $\ell \geq 1$, and the task is to decide if there is a statistic formed by at most ℓ feature queries in $\mathcal{L}$ that separates (D, λ). If, in addition, the number ℓ of features is fixed, we denote the problem by $\mathcal{L}$-Sep[ℓ].

As we show next, the study of $\mathcal{L}$-Sep[$*$] and $\mathcal{L}$-Sep[ℓ] is directly related to *query-by-example* problem (QBE). This allows us to apply the wide arsenal of results and tools for QBE [44,9,6] in order to understand the complexity of $\mathcal{L}$-Sep[$*$]. We first introduce QBE.

6.1. The query-by-example problem

We start by defining the notion of **GHW**(k)*-explanation* for a query language $\mathcal{L}$.

Definition 6.1 *($\mathcal{L}$-explanations).* Let D be a database, and assume that S^+ and S^- are relations over D of *positive* and *negative* examples, respectively. An *$\mathcal{L}$-explanation* for (D, S^+, S^-) is a query $q(\bar{x})$ in $\mathcal{L}$ such that $S^+ \subseteq q(D)$ and $q(D) \cap S^- = \emptyset$.

Notice that when S^+ and S^- are singletons, then an $\mathcal{L}$-explanation for (D, S^+, S^-) is an *$\mathcal{L}$-explanation* for (D, S^+, S^-), where $\mathcal{L}$-explanations are the natural generalization of the notion defined before Proposition 5.8 to any query language $\mathcal{L}$.

The QBE problem for the class $\mathcal{L}$ is then defined as follows.

Problem:	$\mathcal{L}$-QBE
Input:	A database D and relations S^+ and S^- over D
Question:	Is there an $\mathcal{L}$-explanation for (D, S^+, S^-)?

The following is known regarding the complexity of QBE:

Theorem 6.1. [44,9,6] *The following statements hold:*

- **CQ**-QBE *is* coNEXPTIME-*complete.*
- **GHW**(k)-QBE *is* EXPTIME-*complete, for each* $k \geq 1$.

The lower bounds continue to hold even if the schema is fixed and S^+, S^- *are nonempty unary relations such that* $S^- = dom(D) \setminus S^+$.

6.2. Separability for bounded dimension

One of the crucial properties used in the study of separability is that a training database (D, λ) is **CQ**-separable iff there are no entities $e, e' \in \eta(D)$ such that $\lambda(e) \neq \lambda(e')$, yet e and e' are "indistinguishable" by CQs [28]. As the next example shows, this does not hold under the current restriction on the dimension of the statistic.

Example 6.1. Let σ be an entity schema with two unary symbols R and S and the entity symbol η. Consider the database

$$D = \{R(a), S(a), S(c), \eta(a), \eta(b), \eta(c)\}$$

over σ. We define a labeling $\lambda : \eta(D) \to \{1, -1\}$ such that

$$\lambda(a) = \lambda(b) = 1 \text{ and } \lambda(c) = -1.$$

It is not hard to see that (D, λ) is not **CQ**-separable by a statistic with one feature. This is in spite of the fact that a can be distinguished from c by the CQ $R(x)$, and b can be distinguished from c by the CQ $S(x)$. On the other hand, (D, λ) is **CQ**-separable by a statistic with two features; namely, $\Pi = (R(x), S(x))$. □

On the other hand, we can design a simple "guess-and-check" algorithm that solves $\mathcal{L}$-Sep[$*$], for an arbitrary class $\mathcal{L}$ of CQs, if we know how to solve $\mathcal{L}$-QBE.

Test:	$(\mathcal{L}, \ell)$-separability
Input:	A training database (D, λ)
Condition:	Accept if for each $e \in \eta(D)$ there is a vector $\bar{\kappa}_e \in \{1, -1\}^\ell$ such that

- $(\bar{\kappa}_e, \lambda(e))_{e \in \eta(D)}$ is linearly separable; and
- for all $j \in \{1, \ldots, \ell\}$ an $\mathcal{L}$-explanation for (D, S_j^+, S_j^-) exists, where $S_j^+ = \{e \mid \bar{\kappa}_e[j] = 1\}$, $S_j^- = \{e \mid \bar{\kappa}_e[j] = -1\}$, and $\bar{\kappa}_e[j]$ is the j-th component of $\bar{\kappa}_e[j]$

It is easy to see that the following holds for every class $\mathcal{L}$ of CQs.

Lemma 6.2. *A training database is* $\mathcal{L}$*-separable by a statistic with at most* ℓ *features if and only if the* $(\mathcal{L}, \ell)$*-separability test accepts* (D, λ).

Proof. Assume that (D, λ) is $\mathcal{L}$-separable by a statistic Π with l features. Then $\bar{\kappa}_e$ can be chosen to be $\Pi^D(e)$, for each $e \in \eta(D)$.

On the other hand, since the $(\mathcal{L}, \ell)$-separability test accepts (D, λ), there is an $\mathcal{L}$-explanation $q_j(x)$ for (D, S_j^+, S_j^-), where $S_j^+ = \{e \mid \bar{\kappa}_e[j] = 1\}$ and $S_j^- = \{e \mid \bar{\kappa}_e[j] = -1\}$, for all $j \in \{1, \ldots, \ell\}$. Such $\mathcal{L}$-explanations form a statistic $\Pi = (q_1(x), \ldots, q_\ell(x))$. From the linear separability of $(\bar{\kappa}_e, \lambda(e))_{e \in \eta(D)}$ it follows then that there is a linear classifier $\Lambda_{\bar{w}}$ such that $\Lambda_{\bar{w}}(\bar{\kappa}_e) = \lambda(e)$, for all $e \in \eta(D)$. □

It is not hard to see, by applying Theorem 6.1, that for every $\ell \geq 1$ the (**CQ**, ℓ)-separability test can be carried out in coNEXPTIME, while for every fixed $k \geq 1$, the (**GHW**(k), ℓ)-separability test can be carried out in EXPTIME. Then, from Lemma 6.2 we obtain an upper bound for the complexity of **CQ**-Sep[$*$] and **GHW**(k)-Sep[$*$], for every $k \geq 1$.

Algorithm 2 Algorithm for $\mathcal{L}$-Sep[$*$].

Require: A training database (D, λ) and an integer $\ell \in \mathbb{N}$
1: **for** each $e \in \eta(D)$ and each vector $\bar{\kappa}_e \in \{1, -1\}^\ell$ **do**
2: **if** $(\bar{\kappa}_e, \lambda(e))_{e \in \eta(D)}$ is linearly separable and for all $j \in \{1, \ldots, \ell\}$ an $\mathcal{L}$-explanation for (D, S_j^+, S_j^-) exists **then**
3: (D, λ) is $\mathcal{L}$-separable by a statistics with at most ℓ features
4: **end if**
5: **end for**
6: (D, λ) is not $\mathcal{L}$-separable by a statistics with at most ℓ features

Proposition 6.3. *The following statements hold:*

- **CQ**-Sep[$*$] *is in* coNEXPTIME, *while*
- **GHW**(k)-Sep[$*$] *is in* EXPTIME *for every* $k \geq 1$.

Proof. Let $\mathcal{L} = \mathbf{CQ}$ or $\mathcal{L} = \mathbf{GHW}(k)$. From Lemma 6.2 we can use Algorithm 2 below for solving $\mathcal{L}$-Sep[$*$].

The algorithm can clearly be implemented in exponential time if we are granted access to $\mathcal{L}$-QBE to check whether an $\mathcal{L}$-explanation for (D, S_j^+, S_j^-) exists. We obtain from Theorem 6.1 the $\mathcal{L}$-QBE subroutine can be implemented in coNEXPTIME for **CQ**, and in EXPTIME for **GHW**(k). Since the input (D, S_j^+, S_j^-) to $\mathcal{L}$-QBE in Algorithm 2 is of polynomial size, it easily follows then that the algorithm can also be implemented in coNEXPTIME for **CQ**, and in EXPTIME for **GHW**(k). □

It can be shown that these bounds are optimal by using a general reduction from QBE for any class $\mathcal{L}$ of CQs (under a mild assumption on $\mathcal{L}$). This reduction actually states something stronger: The lower bound for our problems continue to hold even if the number of features $\ell \geq 1$ is fixed.

Lemma 6.4. *Let $\mathcal{L}$ be a class of CQs such that, for every schema σ, all CQs with only one atom over σ belong to $\mathcal{L}$.*

Fix $\ell \geq 1$. Then $\mathcal{L}$-QBE reduces in polynomial time to $\mathcal{L}$-Sep[ℓ], when the former is restricted to inputs (D, S^+, S^-) such that S^+, S^- are nonempty unary relations with $S^- = dom(D) \setminus S^+$.

Proof. Let D be a database over some schema σ and assume that S^+, S^- are nonempty unary relations over D such that $S^- = \mathsf{dom}(D) \setminus S^+$. Define an entity schema σ' that extends σ with the entity symbol η and $\ell - 1$ fresh unary symbols $\kappa_1, \ldots, \kappa_{\ell-1}$. We construct a database D' over σ' that extends D with fresh constants $c^-, c_1, ..., c_{\ell-1}$ and facts $\kappa_1(c_1), \ldots, \kappa_{\ell-1}(c_{\ell-1})$. We define

$$\eta(D') = S^+ \cup S^- \cup \{c^-, c_1, ..., c_{\ell-1}\} = \mathsf{dom}(D'),$$

and a labeling $\lambda : \eta(D') \to \{1, -1\}$ in such a way that $\lambda(e) = 1$ if $e \in S^+ \cup \{c_1, ..., c_{\ell-1}\}$ and $\lambda(e) = -1$ if $e \in S^- \cup \{c^-\}$.

By construction of D', for any CQ $q(x)$ over σ':

1. If $c^- \in q(D')$ then $q(D') = \eta(D') = \mathsf{dom}(D')$.
2. For each $i \in \{1, \ldots, \ell - 1\}$, if $c_i \in q(D')$ then either $q(D') = \{c_i\}$ or $q(D') = \eta(D') = \mathsf{dom}(D')$.

We claim that there is an $\mathcal{L}$-explanation for (D, S^+, S^-) iff (D', λ) is $\mathcal{L}$-separable by a statistics with ℓ features. Assume first that $q(x)$ is an $\mathcal{L}$-explanation for (D, S^+, S^-). Let $q_i(x) := \kappa_i(x)$, for each $1 \leq i \leq \ell - 1$. Then the statistic $\Pi = (q_1, \ldots, q_{\ell-1}, q)$ belongs to $\mathcal{L}$ by hypothesis. Moreover, $(\Pi, \Lambda_{\bar{w}})$ separates (D', λ), where $\bar{w} = (1 - l, 1, \ldots, 1)$. The proof of this fact mimics the one presented by the first two paragraphs of the proof of Claim 2 in the proof of Theorem 5.7, so we omit it.

Assume, on the other hand, that Π separates (D', λ), where Π is a statistic with ℓ features from $\mathcal{L}$. It can be proved then that there are at least $\ell - 1$ distinct feature queries $q_1, \ldots q_{\ell-1}$ in Π, such that for each $1 \leq j \leq \ell - 1$ and $e \in S^+ \cup S^- \cup \{c^-\}$ it holds that $e \notin q_j(D)$. The proof of this fact mimics the one presented by the last two paragraphs of the proof of Claim 2, so we again omit it. Now, aside from $\{q_1, \ldots q_{\ell-1}\}$, there is only one more feature query $q(x)$ in Π. By our previous observation, it must be the case that $e \in q(D') \Leftrightarrow e' \notin q(D')$ for each $e \in S^+$ and $e' \in S^- \cup \{c^-\}$ (as otherwise there would be entities $e \in S^+$ and $e' \in S^- \cup \{c^-\}$ such that $\Pi^{D'}(e) = \Pi^{D'}(e')$, contradicting the fact that Π separates (D', λ)). By property (1) then, it must be the case that $e \in q(D')$ for each $e \in S^+$, and $e' \notin q(D')$ for each $e' \in S^- \cup \{c^-\}$. This means that $S^+ \subseteq q(D')$ and $(S^- \cup \{c^-\}) \cap q(D') = \emptyset$.

It remains to show that we can restrict q so that it only contains symbols from σ, i.e., if q' is the query obtained from q by removing atoms of the form $\eta(x)$ and $\kappa_i(x)$, then $q'(D') = q(D')$. The proof of this fact mimics the proof of Claim 3, so we omit it. □

In view of Theorem 6.1 and Lemma 6.4, we obtain the following:

Theorem 6.5. *It is the case that:*

- **CQ**-SEP[$*$] *is* coNEXPTIME-*complete.*
- **GHW**(k)-SEP[$*$] *is* EXPTIME-*complete, for each* $k \geq 1$.

The lower bounds continue to hold even for the $\mathcal{L}$-SEP[ℓ] *problem, for any fixed* $\ell \geq 1$, *where* $\mathcal{L}$ *is either* **CQ** *or* **GHW**(k).

The lower bounds for **CQ**-SEP[ℓ] and **GHW**(k)-SEP[ℓ] established in the previous theorem hold even for a fixed schema. This is based on the fact that the lower bounds in Theorem 6.1 hold over a fixed schema, and the reduction from QBE to $\mathcal{L}$-SEP[ℓ] provided in the proof of Lemma 6.4 enlarges the schema of the input database for QBE with only ℓ extra unary symbols (for fixed $\ell \geq 1$).

6.3. Generating a statistic

Next we establish lower bounds on the number of atoms required by feature queries under the assumption that statistics are of a bounded dimension.

Theorem 6.6. *Fix* $\ell \geq 1$. *For every* $n \geq 1$ *there is a training database* (D, λ) *such that:*

1. $|D|$ *is polynomial in* n,
2. (D, λ) *is* **CQ**-*separable,*
3. *for every statistics* $\Pi = (q_1, \ldots, q_\ell)$ *that* **CQ**-*separates* (D, λ), *at least one* q_i *has* $\Omega(2^n)$ *atoms.*

This holds true if we restrict to the class of statistics formed by CQs in **GHW**(k), *but then, at least one* q_i *has* $\Omega(2^{2^n})$ *atoms.*

Proof. We start with the case of **CQ**. We use the following result.

Proposition 6.7. *[44] There is a family* $(D_n, S_n^+, S_n^-)_{n\geq 0}$ *of tuples of databases* D_n *and unary relations* S_n^+ *and* S_n^- *over* $\mathsf{dom}(D_n)$ *with* $S_n^- = \mathsf{dom}(D_n) \setminus S_n^+$, *such that*

- *the size of* D_n *is polynomial in* n,
- *there is a* **CQ**-*explanation for* (D_n, S_n^+, S_n^-), *and*
- *the smallest such* **CQ**-*explanation has* $\Omega(2^n)$ *atoms.*

Let (D_n, S_n^+, S_n^-) be as defined in the previous proposition. We proceed in the same way than in the proof of Lemma 6.4: that is, we construct a database D that extends D_n with fresh constants $c^-, c_1, \ldots, c_{\ell-1}$ and fresh facts $\kappa_1(c_1), \ldots, \kappa_{\ell-1}(c_{\ell-1})$, and then define

$$\eta(D) = S_n^+ \cup S_n^- \cup \{c^-, c_1, \ldots, c_{\ell-1}\} = \mathsf{dom}(D),$$

and a labeling $\lambda : \eta(D) \to \{1, -1\}$ in such a way that $\lambda(e) = 1$ if $e \in S^+ \cup \{c_1, \ldots, c_{\ell-1}\}$ and $\lambda(e) = -1$ if $e \in S^- \cup \{c^-\}$.

From the proof of Lemma 6.4, we know that there is an **CQ**-explanation for (D_n, S_n^+, S_n^-) iff (D, λ) is **CQ**-separable by a statistics with exactly ℓ features. It suffices to inspect one direction of the proof of Lemma 6.4, namely, where we assume that there is a statistic Π with ℓ features in **CQ** that separates (D, λ). There we show that Π must contain a query $q(x)$ such that, after removing all the atoms of the form $\kappa_i(x_i)$ from q, it becomes an $\mathcal{L}$-explanation for (D_n, S_n^+, S_n^-). Therefore, $q(x)$ must have at least $\Omega(2^n)$ atoms from Proposition [44].

The proof for **GHW**(k) is analogous, this time using Proposition 5.8. □

In summary, while bounding the dimension of statistics for general CQs and CQs of bounded ghw is positive from a generalization point of view, it also creates new problems that affect the practicality of the approach: (1) The complexity of separability becomes prohibitively high, and (2) feature queries can grow exponentially large (or even double exponentially if we bound their ghw).

6.4. Bounded dimension and number of feature atoms

Let us go back to the restriction on statistics introduced in Section 4: fixing the number of atoms allowed in feature CQs. Recall that this restriction is well-behaved in terms of separability; in fact, the problem becomes FPT, with the parameter being the arity of the schema (see Corollary 4.2). In addition, this restriction prevents statistics from growing too large in terms of the size of the data. In fact, the number of different CQs in **CQ**[m]—the class of CQs with at most m atoms—depends exclusively on m and the underlying schema σ (in particular, in the number $r \geq 1$ of relation symbols in σ and the maximum arity $k \geq 1$ of any such a relation symbol).

Yet, the number of different CQs in **CQ**[m] is exponential in the combined size of m and k, and thus could still be quite large for practical purposes. It might be reasonable then in this case to also bound the number of feature queries allowed

in statistics. This calls for the study of **CQ**[m]-Sep[$*$] and **CQ**[m]-Sep[ℓ], that is, the separability problem for statistics based on the class of CQs with at most m atoms wherein the number of features is bounded or corresponds to the fixed $\ell \geq 1$, respectively.

It is not hard to see that **CQ**[m]-Sep[$*$] is FPT, with the parameter being the *size* of the schema, which is the upper bound between the number of symbols in the schema and the maximum arity over all such symbols. The proof of this fact is constructive in the sense that it yields a pair $(\Pi, \Lambda_{\bar{w}})$ that **CQ**[m]-separates the input training database (D, λ) where Π has at most ℓ features. Therefore, the classification problem **CQ**[m]-Cls[$*$] is also FPT.

Proposition 6.8. *For each $m \geq 1$ both* **CQ**[m]-Sep[$*$] *and* **CQ**[m]-Cls[$*$] *are* FPT, *with the parameter being the size of the schema.*

Notice the difference with Corollary 4.2, which establishes that **CQ**[m]-Sep is FPT with the parameter being the *arity* of the schema only. As we show next, the extra requirement on the parameter is necessary (under conventional complexity assumptions).

Proposition 6.9. *For each $m \geq 1$ the problem* **CQ**[m]-Sep[$*$] *is* NP*-complete even for fixed arity schemas.*

We do not provide a proof of this result now, as we actually prove a stronger result later in item (2) of Proposition 6.12.

Therefore, if for any $m \geq 1$ the problem **CQ**[m]-Sep[$*$] is FPT with the parameter being the arity of the schema, then P = NP. The reason why **CQ**[m]-Sep[$*$] is NP-hard is because it involves *choosing* a set of at most ℓ feature CQs in **CQ**[m], for a given $\ell \geq 1$, that separates the input (D, λ). Notice that this establishes an interesting difference with the problem **CQ**[m]-Sep, which we do not know whether it is NP-hard.

Interestingly, the intractability holds even if the number of features is fixed (but the arity of the schema is not).

Theorem 6.10. *The problem* **CQ**[m]-Sep[ℓ] *is* NP*-complete, for each fixed $\ell \geq 1$.*

We now explain the proof of the NP-hardness in Theorem 6.10. Recall that Lemma 6.4 provides a general way of obtaining lower bounds for separability with a fixed number of features via a reduction from a restricted version of QBE. However, unlike the case of **CQ** and **GHW**(k), for $k \geq 1$, for which the complexity of QBE is well understood, the complexity of QBE for **CQ**[m], for $m \geq 1$, has not been studied in the literature. We show it to be NP-complete below, even in the restricted setting required by Lemma 6.4, which is a surprisingly negative result. In fact, the problem is NP-complete even for the class **CQ**[1] of single-atom CQs.

Proposition 6.11. **CQ**[m]-QBE *is* NP*-complete for each fixed $m \geq 1$. The lower bound holds even if the input is of the form (D, S^+, S^-) and S^+, S^- are nonempty unary relations such that $S^- = dom(D) \setminus S^+$.*

The lower bound in Theorem 6.10 follows directly then from Lemma 6.4 and Proposition 6.11. The rest of this section is devoted to proving Proposition 6.11.

Proof of Proposition 6.11. Consider the following problem, which we call ConstrainedPartition. Its input $\mathcal{I}$ consists of an integer $n \geq 1$ given in unary, a collection of *positive constraints* of the form $X \subseteq Y_1 \vee X \subseteq Y_2 \vee \cdots \vee X \subseteq Y_r$, where each Y_i is a partition of $[n] = \{1, \ldots, n\}$, and a collection of *negative constraints* $X \not\subseteq Z$, where Z is a partition of $[n]$. (We assume that partitions of $[n]$ are given as reflexive, symmetric, and transitive binary relations on $[n]$). The problem consists in determining if there is a *solution* for $\mathcal{I}$, i.e., a partition X of $[n]$ that satisfies all positive and negative constraints specified in $\mathcal{I}$.

The strategy behind the proof of Proposition 6.11 is as follows.

(a). We first show that ConstrainedPartition is NP-complete.
(b). We then show that ConstrainedPartition reduces in polynomial time to the restriction of **CQ**[1]-QBE defined by inputs of the form (D, S^+, S^-), where S^+, S^- are nonempty unary relations such that $S^- = \mathsf{dom}(D) \setminus S^+$.
(c). We then reduce in polynomial time the latter problem to **CQ**[m]-QBE, for each $m \geq 1$, for inputs as specified in (2).

Proof of (a). We reduce from SAT. Given a CNF formula F, we construct an instance of the ConstrainedPartition problem in the following way. Let n be $4|V|$, assuming that $V = \{v_1, v_2, \ldots\}$ are the variables of F. We associate with every literal ℓ of F a partition X_ℓ on $[n]$ as follows.

- If $\ell = v_i$, then the classes of X_ℓ are $\{4i-3, 4i\}, \{4i-2, 4i-1\}$, and another class containing the rest of elements in $[n]$.
- If $\ell = \neg v_i$, then the classes of X_ℓ are $\{4i-3, 4i-1\}, \{4i-2, 4i\}$, and another class containing the rest of elements in $[n]$.

In addition, we associate with every set A of literals from F the partition X_A defined as $\bigcap_{\ell \in A} X_\ell$. Notice that it follows directly from the definition that $A \subseteq B$ iff $X_B \subseteq X_A$, for every pair A, B of sets of literals.

The instance $\mathcal{I}$ of ConstrainedPartition contains then the following constraints.

- For every variable v_i in V, we have that $\mathcal{I}$ contains the positive constraint $(X \subseteq X_{v_i}) \vee (X \subseteq X_{\neg v_i})$.
- For every clause C in F, it is the case that $\mathcal{I}$ contains the negative constraint $X \not\subseteq \bigcap_{\ell \in C} X_{\neg \ell}$.

Let s be any assignment on the variables of F and define $X = X_A$, where A is the set of all literals satisfied by s. Since $A \subseteq B$ iff $X_B \subseteq X_A$, it follows that

1. X satisfies all positive constraints in $\mathcal{I}$, and
2. X satisfies all negative constraints in $\mathcal{I}$ iff s satisfies F.

Therefore, if F is satisfiable then our instance $\mathcal{I}$ of ConstrainedPartition has a solution.

On the other hand, assume that the ConstrainedPartition instance $\mathcal{I}$ has some solution X. Let A be a set of literals constructed in the following way. For each variable v_i, we know that the constraint $(X \subseteq X_{v_i}) \vee (X \subseteq X_{\neg v_i})$ is satisfied. Then choose a literal $\ell_i = v_i$ or $\ell_i = \neg v_i$ such that $X \subseteq X_{\ell_i}$ and include ℓ_i in A. It follows by (1) that X_A satisfies every positive constraint in $\mathcal{I}$. Furthermore, since $X \subseteq X_\ell$ for every $\ell \in A$ it follows that $X \subseteq X_A$. Hence, X_A is also a solution of the instance $\mathcal{I}$ due to the form of the negative constraints. Consequently, we can directly obtain from A an assignment satisfying F.

Proof of (b). Consider an input $\mathcal{I}$ to ConstrainedPartition. We assume without loss of generality that $\mathcal{I}$ contains the positive constraint $X \subseteq [n] \times [n]$. From $\mathcal{I}$ we construct a database D as follows.

- Let $\mathcal{C}$ be the collection of all positive constrains in $\mathcal{I}$. For each $c \in \mathcal{C}$ of the form

  $$X \subseteq Y_1 \vee X \subseteq Y_2 \vee \cdots \vee X \subseteq Y_r,$$

  we add to D a tuple of the form $R(c, \bar{y}_i)$, for each $i \in \{1, \ldots, r\}$, where $\bar{y}_i$ is an n-ary tuple of fresh elements representing the partition Y_i of $[n]$, i.e., if $\bar{y}_i = (y_1, \ldots, y_n)$ then for each $1 \leq j \leq k \leq n$ we have that $y_j = y_k$ iff $(j, k) \in Y_i$.
- Let $\mathcal{D}$ be the collection of all negative constrains in $\mathcal{I}$. For each $d \in \mathcal{D}$ of the form $X \not\subseteq Z$, we add to D the tuple $R(d, \bar{z})$, where as before $\bar{z}$ is an n-ary tuple of fresh elements representing the partition Z on $[n]$.

Also, we define $S^+ := \{c \mid c \in \mathcal{C}\}$ and $S^- = \mathsf{dom}(D) \setminus S^+$. Clearly, (D, S^+, S^-) can be constructed in polynomial time from $\mathcal{I}$. We show next that $\mathcal{I}$ has a solution iff (D, S^+, S^-) has a **CQ**[1]-explanation.

Let X be a partition of $[n]$ and $\bar{x}$ a tuple of elements representing X. It follows directly from the definitions that if x_0 is an element not in $\bar{x}$ then the evaluation of a query of the form $R(x_0, \bar{x})$ over D contains a tuple of the form $(e, \bar{y})$, where $e \in \mathcal{C} \cup \mathcal{D}$ and $\bar{y}$ is a tuple representing partition Y of $[n]$, iff $X \subseteq Y$. This directly implies the following claim.

Claim 4. *Let $\bar{x}$ a tuple of elements representing partition X of $[n]$. Then X is a solution for I iff $q(D) = S^+$, where $q(x_0) = \exists \bar{x} R(x_0, \bar{x})$.*

We now prove the correctness of the construction. Assume first that $\mathcal{I}$ has a solution X. By Claim 4, the CQ $q(x_0) = \exists \bar{x} R(x_0, \bar{x})$ is a **CQ**[1]-explanation for (D, S^+, S^-), where $\bar{x}$ is a tuple of elements representing partition X and x_0 does not occur in $\bar{x}$.

Assume now that there is a **CQ**[1]-explanation for (D, S^+, S^-). Then this explanation can be assumed to be of the form

$$q(x_i) := \exists \bar{y}\, R(x_0, \ldots, x_n), \quad i \in \{0, \ldots, n\},$$

where the x_js, for $j \in \{0, \ldots, n\}$, are not necessarily distinct variables and $\bar{y}$ is a tuple that contains all the x_js that are different from x_i. It remains to show that $i = 0$ and $x_0 \neq x_j$ for each $j \in \{1, \ldots, n\}$. In fact, consider the constraint $c = X \subseteq [n] \times [n]$ in $\mathcal{I}$. Then $R(c, \bar{y}) \in D$, where $\bar{y}$ is a tuple of fresh elements representing partition $[n] \times [n]$. Hence $c \in q(D)$ since $q(x_i)$ is a **CQ**[1]-explanation for (D, S^+, S^-). But c only appears in the first coordinate of the tuples in D, and thus $i = 0$. For the same reason, $x_0 \neq x_j$ for each $j \in \{1, \ldots, n\}$. Therefore, there is a **CQ**[1]-explanation for (D, S^+, S^-) of the form $\exists \bar{x} R(x_0, \bar{x})$. Then X is a solution for $\mathcal{I}$ from Claim 4, where X is the partition of $[n]$ represented by $\bar{x}$.

Proof of (c). It remains to show that **CQ**[1]-QBE reduces to **CQ**[m]-QBE. Let D be a database and S^+, S^- nonempty unary relations such that $S^- = \mathsf{dom}(D) \setminus S^+$. Consider m disjoint copies of the input (D, S^+, S^-), namely $(D_1, S_1^+, S_1^-), \ldots, (D_m, S_m^+, S_m^-)$. Now, construct the database D' as the union $\bigcup_{i=1}^m D_i$ extended by $m - 1$ fresh unary relations

$$\#_i(D') = \mathsf{dom}(D') \setminus \mathsf{dom}(D_i), \qquad \text{for } 2 \leq i \leq m,$$

and unary relations $T^+ = S_1^+$ and $T^- = \mathsf{dom}(D') \setminus S_1^+$.

Claim 5. *There is a* **CQ**[1]*-explanation for* (D, S^+, S^-) *iff there is a* **CQ**[m]*-explanation for* (D', T^+, T^-).

First, assume that there is a query $q(x)$ in **CQ**[1] such that $S^+ \subseteq q(D)$ and $q(D) \cap S^- = \emptyset$. It follows that $(S_1^+ \cup \cdots \cup S_m^+) \subseteq q(D')$ and $q(D') \cap (S_1^- \cup \cdots \cup S_m^-) = \emptyset$. Extending $q(x)$ by the $m-1$ atoms $\#_2(x), \ldots, \#_m(x)$ gives the desired explanation, separating T^+ from T^-.

On the other hand, assume that there is a query $q(x)$ in **CQ**[m] such that $T^+ \subseteq q(D')$ and $q(D') \cap T^- = \emptyset$. From

$$S_1^+ \subseteq q(D') \text{ and } q(D') \cap (S_2^+ \cup \cdots \cup S_m^+) = \emptyset,$$

it follows that $q(x)$ must contain the $m-1$ atoms $\#_2(x), \ldots, \#_m(x)$. Let $q'(x)$ be the query in **CQ**[1] which is obtained from $q(x)$ by removing any atom of the form $\#_i(y)$, for $2 \leq i \leq m$, where y is either existentially quantified or $y = x$. It must be the case that $q'(x)$ is of the form $\exists \bar{y} R(x, \bar{y})$, for R some relation symbol. We claim that $q'(D')$ separates S_1^+ from S_1^- over D_1, and thus $q(x)$ is a **CQ**[1]-explanation for (D, S^+, S^-). In fact, by definition $S_1^+ \subseteq q(D')$, which implies that $S_1^+ \subseteq q'(D')$. In addition, $q'(D') \cap S_1^- = \emptyset$. Assume otherwise. Then also $q(D') \cap S_1^- \neq \emptyset$ as every element in S_1^- satisfy each atom $\#_j(x)$ which is in $q(x)$ but not in $q'(x)$. □

6.5. Fixed number of variable occurrences

Recall from Proposition 4.3 that we can ensure tractability of separability, for statistics with an unbounded number of features, by fixing *both* the number of atoms and the number of occurrences of variables in feature queries; that is, **CQ**[m, p]-Sep is in PTIME, for fixed $m, p \geq 1$.

In the current scenario this continues to hold only if we fix the number $\ell \geq 1$ of features allowed in statistics. In turn, if the number ℓ is given as part of the input the problem becomes NP-hard.

Proposition 6.12. *Fix* $m, p \geq 1$. *The following holds:*

1. *The problems* **CQ**[m, p]-Sep[ℓ] *and* **CQ**[m, p]-Cls[ℓ] *are in* PTIME, *for each fixed* $\ell \geq 1$.
2. *The problem* **CQ**[m, p]-Sep[$*$] *is* NP*-complete. This holds even for fixed arity schemas.*

Proof. We start with (1). Since all feature queries in **CQ**[m, p] can be generated in polynomial time (modulo renaming of variables) and ℓ is fixed, we can also generate in polynomial time all possible statistics containing at most ℓ feature queries in **CQ**[m, p]. Also, for each such statistic Π, we can find, in polynomial time, a linear classifier separating (D, λ) or report that none exists. This is because, since m is fixed, we can construct $\Pi^D(e)$ for every $e \in \eta(D)$ in polynomial time.

We now prove (2). First note that, since m is fixed, the $(\mathcal{L}, \ell)$-separability test from section 6.2 implies that **CQ**[m, p]-Sep[$*$] is in NP. To prove NP-hardness we consider the following problem, which we shall call BoundedLinSep: given a collection $\langle(\overline{b_i}, y_i)_{i=1}^m\rangle$ of Boolean examples and $\ell \geq 1$, decide whether $\langle(\overline{b_i}, y_i)_{i=1}^m\rangle$ is linearly separable by a linear classifier $\Lambda_{\bar{w}}$ such that $\overline{w} = (w_0, w_1, \ldots, w_n)$ contains at most ℓ non-zero values besides w_0.

First, we prove that there is a polynomial time reduction from VertexCover to BoundedLinSep, and hence that BoundedLinSep is NP-hard. Let (G, ℓ) be an instance of VertexCover, where $G = (V, E)$ is a graph and $\ell \geq 1$. Let $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$ and construct the instance $(\langle(\overline{b_i}, y_i)_{i=1}^{m+1}\rangle, \ell)$ of BoundedLinSep, where $\langle(\overline{b_i}, y_i)_{i=1}^{m+1}\rangle$ is defined as follows.

- For every $i = 1, \ldots, m$ we have $y_i = 1$ and $\overline{b_i} = (b_i^1, \ldots, b_i^n)$, where $b_i^j = 1$ if e_i is incident to v_j and 0 otherwise.
- In addition, $y_{m+1} = -1$ and $\overline{b_{m+1}} = (0, \ldots, 0)$.

The correctness of the reduction is a direct consequence of the following proposition.

Proposition 6.13. *For every* $S \subseteq \{1, \ldots, n\}$, *it is the case that* $\{v_j \mid j \in S\}$ *is a vertex cover of* G *iff there is linear classifier* $\Lambda_{(w_0, w_1, \ldots, w_n)}$ *such that* $w_j = 0$ *for every* $j \geq 1$ *not in* S.

Proof. Assume first that $\{v_j \mid j \in S\}$ is a vertex cover and define $\overline{w} = (w_0, w_1, \ldots, w_n)$, where $w_j = 1$ if $j \in S \cup \{0\}$ and 0 otherwise. It follows easily that $\Lambda_{\bar{w}}$ linearly classifies $(\overline{b_i}, y_i)_{i=1}^{m+1}$. Conversely, assume that $\{v_j \mid j \in S\}$ is not a vertex cover and let e_i be an edge not covered by it. It follows that $\Lambda_{\bar{w}}(\overline{b_i}) = \Lambda_{\bar{w}}(\overline{b_{m+1}})$ for every $\overline{w} = (w_0, w_1, \ldots, w_n)$ such that $w_j = 0$ for every $j \geq 1$ not in S. □

To complete the proof, we show that there is a polynomial time reduction from BoundedLinSep to **CQ**[m, p]-Sep[$*$]. Let $\langle(\overline{b_i}, y_i)_{i=1}^m\rangle$ be a collection of Boolean examples, where $\overline{b_i} = (b_i^1, \ldots, b_i^n)$. Construct a training database (D, λ) as follows. The schema of D contains, besides entity relation η, a binary relation E and n unary relations $\kappa_1, \ldots, \kappa_n$. Assume that initially D is empty. Let K be 1 if $p = 1$ or $m = 1$, and $m - 1$ otherwise. For each $i \in \{1, \ldots, m\}$, add K new elements $d_i^1, \ldots, d_i^K$ to

$\mathsf{dom}(D)$. Also include facts $E(d_i^r, d_i^{r+1})_{r=1}^{K-1}$ and $\kappa_j(d_i^K)$ for every $j \in \{1, \ldots, n\}$ with $b_i^j = 1$. Finally, set $\eta(D) = \{d_1^1, \ldots, d_m^1\}$ and let $\lambda(d_i^1) = y_i$, for every $1 \leq i \leq m$.

For every $j = 1, \ldots, n$, let $q^j(x_1) \in \mathbf{CQ}[m, p]$ be the query

$$\exists x_2, \ldots, x_K (E(x_1, x_2) \wedge \cdots \wedge E(x_{K-1}, x_K) \wedge \kappa_j(x_K))$$

It is easy to see that if some query $q \in \mathbf{CQ}[m, p]$ is not equivalent (under renaming of variables) to any query of the form q_j, then $q(D)$ is either $\emptyset$ or $\eta(D)$. Consequently, if Π is any statistic that separates (D, λ) we can assume that all its feature queries are from $\{q^1, \ldots, q^n\}$. Also, note that by construction for every $j \in \{1, \ldots, n\}$ and $i \in \{1, \ldots, m\}$, $d_i^1 \in q^j(D)$ iff $b_i^j = 1$. It follows that for every linear classifier $\overline{w}$, we have that $\Lambda_{\bar{w}}(\overline{b_i}) = \Lambda_{\bar{w}}(\Pi^D(d_i^1))$. Correctness of the reduction follows directly. □

7. Approximate separability

We now discuss a generalization of the separability problem, allowing some examples to be misclassified. Hence, we handle the case where a training database is inseparable due to a small amount of noise in the data. This notion of approximation captures the common goal of minimizing the number of misclassified examples [10,37,26], and corresponds to one of the studied notions of separation errors [5,47]. We revise the previously obtained complexity results for the case that a relative error ϵ, for $0 \leq \epsilon < 1$, is allowed in the classification of the training examples.

Formally, a training database (D, λ) is $\mathcal{L}$-separable *with error* ϵ if there is a statistic Π, with feature queries from $\mathcal{L}$, and a linear classifier $\Lambda_{\bar{w}}$, such that

$$\{e \in \eta(D) \mid \Lambda_{\bar{w}}(\Pi^D(e)) \neq \lambda(e)\} \leq \epsilon \cdot |\eta(D)|\,.$$

We then say that $(\Pi, \Lambda_{\bar{w}})$ $\mathcal{L}$*-separates* (D, λ) with error ϵ. We study the following problem.

Problem:	$\mathcal{L}$-ApxSep
Input:	A training database (D, λ) and an $\epsilon \in [0, 1)$
Question:	Is (D, λ) $\mathcal{L}$-separable with error ϵ?

As before, we study two variants of this problem in which the dimension is given as input or bounded by a constant $\ell \geq 1$. These are denoted by $\mathcal{L}$-*ApxSep*$[*]$ and $\mathcal{L}$-*ApxSep*$[\ell]$, respectively.

7.1. Intractable cases

$\mathcal{L}$-ApxSep is at least as difficult as $\mathcal{L}$-Sep, since $\mathcal{L}$-Sep is precisely $\mathcal{L}$-ApxSep when $\epsilon = 0$. Thus all lower bounds obtained for the latter along the paper continue to hold for the former. The same holds for $\mathcal{L}$-ApxSep$[*]$ and $\mathcal{L}$-ApxSep$[\ell]$ w.r.t. $\mathcal{L}$-Sep$[*]$ and $\mathcal{L}$-Sep$[\ell]$, respectively. More interestingly, such lower bounds continue to hold even if ϵ is an arbitrary *fixed* value with $\epsilon \in [0, 1/2)$.[1] This is proved via a polynomial-time reduction from $\mathcal{L}$-Sep (resp., $\mathcal{L}$-Sep$[*]$ and $\mathcal{L}$-Sep$[\ell]$) to $(\mathcal{L}, \epsilon)$-ApxSep (resp., $(\mathcal{L}, \epsilon)$-ApxSep$[*]$ and $(\mathcal{L}, \epsilon)$-ApxSep$[\ell]$), the restriction of $\mathcal{L}$-ApxSep (resp., $\mathcal{L}$-ApxSep$[*]$ and $\mathcal{L}$-ApxSep$[\ell]$) in which ϵ is an arbitrary fixed value in the interval $[0, 1/2)$. These reductions hold for any class $\mathcal{L}$ of CQs.

Proposition 7.1. *Fix an arbitrary $\epsilon \in [0, 1/2)$. For all $\mathcal{L}$ there are polynomial-time reductions*

- *from $\mathcal{L}$-Sep to $(\mathcal{L}, \epsilon)$-ApxSep;*
- *from $\mathcal{L}$-Sep$[*]$ to $(\mathcal{L}, \epsilon)$-ApxSep$[*]$; and*
- *from $\mathcal{L}$-Sep$[\ell]$ to $(\mathcal{L}, \epsilon)$-ApxSep$[\ell]$ for all fixed $\ell \geq 1$.*

Proof. Let (D, λ) be a training instance over some schema σ and ϵ be the allowed classification error. We construct a training instance (D', λ') such that (D', λ') cannot be $\mathcal{L}$-separated with less than k examples being misclassified, where k is chosen according to the cardinality of $\eta(D)$ and ϵ. Moreover, (D, λ) is $\mathcal{L}$-separable iff (D', λ') can be $\mathcal{L}$-separated with k examples being misclassified.

Let D' be the extension of D with $2k$ fresh constants $c_1, \ldots, c_k$ and $c'_1, \ldots, c'_k$ such that

$$\eta(D') = \eta(D) \cup \{c_1, \ldots, c_k\} \cup \{c'_1, \ldots, c'_k\}$$

and all the other relations of D' are equal to those from D, i.e., the fresh constants do not appear in any relation except for $\eta(D')$. We define $\lambda' : \eta(D') \to \{1, -1\}$ as

[1] For $\epsilon \geq 1/2$ the problem is trivial, since then we can always find a classifier that separates with error ϵ.

$$\lambda'(x) := \begin{cases} \lambda(x) & \text{if } x \in \eta(D), \\ +1 & \text{if } x \in \{c_1, \ldots, c_k\}, \\ -1 & \text{if } x \in \{c'_1, \ldots, c'_k\}. \end{cases}$$

By construction, (D', λ') cannot be $\mathcal{L}$-separated with less than k examples being misclassified. In addition, it is clear that if (D, λ) is $\mathcal{L}$-separable, then (D', λ') can be $\mathcal{L}$-separated with k examples being misclassified. On the other hand, assume that (D', λ') can be $\mathcal{L}$-separated with k examples being misclassified. Since either all the entities $c_1, \ldots, c_k$ or all the entities $c'_1, \ldots, c'_k$ are misclassified by construction, no entity from $\eta(D)$ is misclassified. It follows that (D, λ) is separable.

By definition, the error is $\epsilon = \frac{k}{|\eta(D)|+2k}$. Therefore, we get

$$k = |\eta(D)| \frac{\epsilon}{1-2\epsilon}.$$

For $0 \leq \epsilon < \frac{1}{2}$, it holds that $\frac{\epsilon}{1-2\epsilon}$ is some positive constant M and (D', λ') can be constructed in polynomial time because we only add $2M|\eta(D)| = O(n)$ constants, where n is the size of D. □

Now, as all lower bounds for $\mathcal{L}$-Sep, $\mathcal{L}$-Sep[$*$] and $\mathcal{L}$-Sep[ℓ] presented in the paper are for complexity classes that are closed under polynomial-time reductions, Proposition 7.1 implies that they continue to hold for their approximate versions, even if ϵ is an arbitrary *fixed* value with $0 \leq \epsilon < 1/2$. Therefore, our hardness results do not arise from the aim of finding a "strict" classifier, but are due to the inherent complexity of the problem.

7.2. Feasible cases

In view of the previous discussion, we can only hope to obtain a feasible complexity for approximate separability in the cases where (perfect) separability is also feasible. As we have seen, there are two such cases: statistics formed by CQs with a bounded number of atoms, where separability is FPT (Corollary 4.2 and Proposition 6.8), and statistics of unbounded dimension formed by CQs of bounded ghw, where separability is solvable in PTIME (Theorem 5.3). We study both cases below.

7.2.1. Approximate **CQ**[m]-separability

We first study the approximate separability problem **CQ**[m]-ApxSep for statistics formed by CQs with a fixed number of atoms. It is not hard to see that this problem is FPT, if we assume the parameter to be the size of the schema. Notice again the difference with Corollary 4.2, which establishes that the exact separability problem **CQ**[m]-Sep is FPT with the parameter being the *arity* of the schema only. As in Proposition 6.9, the extra requirement on the parameter is necessary (under conventional complexity assumptions).

Proposition 7.2. *The following holds for each $m \geq 1$.*

1. *The problem* **CQ**[m]-ApxSep *is* FPT *with the parameter being the size of the schema.*
2. *The problem* **CQ**[m]-ApxSep *is* NP*-complete. This holds even for fixed arity schemas.*

Proof. We first prove (1). Let (D, λ) and ϵ an instance of **CQ**[m]-ApxSep. Suppose that Π is the statistic formed by all feature queries in **CQ**[m]. As noted in the proof of Proposition 4.1, it takes time $f(s)$ to construct Π, for f a computable function and s the size of the schema. We then have to check whether there is a linear classifier $\Lambda_{\bar{w}}$ such that $(\Pi, \Lambda_{\bar{w}})$ classifies correctly at least an ϵ-fraction of the elements in $\eta(D)$. To check this we first construct the set V of all vectors of the form $\Pi^D(e)$, for $e \in \eta(D)$. As observed in the proof of Proposition 4.1, it takes time $D^{2m+1} \cdot 2^{p'(s)}$ to compute V, for $p' : \mathbb{N} \to \mathbb{N}$ a polynomial. Moreover, by definition the size of V is bounded by $2^{\dim(\Pi)}$, where $\dim(\Pi)$ denotes the dimension of Π. We then proceed as follows. For each $V' \subseteq V$ we check two things:

- It is the case that V' contains a "large enough" fraction of the vectors corresponding to entities in η:

$$|\{e \in \eta(D) \mid \Pi^D(e) \in V'\}| \geq (1-\epsilon)|\eta(D)|.$$

- There is a linear classifier that correctly classifies all entities in $\{e \in \eta(D) \mid \Pi^D(e) \in V'\}$ with respect to λ.

If such a set V' exists, then the algorithm accepts. It is easy to see that the running time of this algorithm is

$$2^{2^{O(\dim(\Pi))}} \cdot |D|^{O(1)}.$$

But $\dim(\Pi)$ is also bounded by $f(s)$, which shows that the whole procedure can be implemented in FPT when the parameter is assumed to be the size of the schema.

We now prove (2). We use the following problem, which was shown to be NP-hard in [12]: given a collection $\langle(\overline{b_i}, y_i)_{i=1}^m\rangle$ of boolean examples and an $\epsilon \in [0,1]$, decide whether there is a linear classifier that classifies correctly at least a ϵ-fraction of the examples. This problem reduces to $\mathbf{CQ}[m]$-ApxSep as in the proof of item (2) of Proposition 6.12 (setting p to ∞). □

From (2), if for any $m \geq 1$ the problem $\mathbf{CQ}[m]$-ApxSep is FPT with the parameter being the arity of the schema, then P = NP. The difference in complexity between $\mathbf{CQ}[m]$-Sep and $\mathbf{CQ}[m]$-ApxSep stems from the nature of their underlying classification task: $\mathbf{CQ}[m]$-Sep calls for *exact* linear separability, which is in PTIME [27,24], while $\mathbf{CQ}[m]$-ApxSep calls for *approximate* linear separability, which is NP-complete [22]. This yields item (2) in Proposition 7.2.

A similar situation holds for $\mathbf{CQ}[m]$-ApxSep[$*$], the restriction of $\mathbf{CQ}[m]$-ApxSep to statistics with at most ℓ features, where ℓ is given as part of the input. On the other hand, if ℓ is fixed, then we can again ensure fixed-parameter tractability by using only the arity of the schema as the parameter.

Proposition 7.3. *For all fixed $m \geq 1$, the following hold:*

1. *The problem $\mathbf{CQ}[m]$-ApxSep[$*$] is* FPT *with the parameter being the size of the schema.*
2. *The problem $\mathbf{CQ}[m]$-ApxSep[$*$] is* NP*-complete even for fixed arity schemas.*
3. *For every fixed $\ell \geq 1$, the problem $\mathbf{CQ}[m]$-ApxSep[ℓ] is* FPT *with the parameter being the arity of the schema.*

Proof. Let us start with (1). Let (D, λ), $\ell \geq 1$, and $\epsilon \in [0,1)$ be an instance of $\mathbf{CQ}[m]$-ApxSep[$*$]. Consider a statistic Π. We say that a set $S \subseteq \eta(D)$ is Π-*consistent* if $d \in S \Leftrightarrow e \in S$, for every $d, e \in \eta(D)$ with $\Pi^D(d) = \Pi^D(e)$. Also, a set $\mathcal{P}$ of statistics is *complete* for (D, λ), if among all pairs $(\Pi, \Lambda_{\bar{w}})$ with minimum error for (D, λ) there is one where Π belongs to $\mathcal{P}$. A complete family of statistics gives rise to the following algorithm: accept if for some statistic $\Pi \in \mathcal{P}$ and some Π-consistent $S \subseteq \eta(D)$ there is a linear classifier $\Lambda_{\bar{w}}$ such that

$$\Lambda_{\bar{w}}(\Pi^D(e)) = 1 \iff e \in S,$$

and $(\Pi, \Lambda_{\bar{w}})$ classifies correctly at least an ϵ-fraction of the elements in $\eta(D)$.

The correctness of the algorithm is straightforward. Its running time (ignoring the cost of finding $\mathcal{P}$) is

$$O\left(\sum_{\Pi \in \mathcal{P}} 2^{2^{\dim(\Pi)}} |D|^{O(1)}\right)$$

where $\dim(\Pi)$ denotes the dimension of Π.

It remains to find a complete set $\mathcal{P}$ of statistics. As noted in the proof of Proposition 4.1 the set of queries in $\mathbf{CQ}[m]$ that mention only relation symbols that appear in D has cardinality at most $f(s)$, where f is a computable function and s is the size. Hence, we can obtain a complete $\mathcal{P}$ set of cardinality $\sum_{i=1}^{\ell} \binom{f(s)}{i} \leq 2^{f(s)}$ by choosing all statistics containing at most ℓ queries from $\mathbf{CQ}[m]$. Since each statistic in the set has at most $f(s)$ queries, it follows that the algorithm is FPT.

The proof of (3) is analogous, but now we use the fact that the number of queries in $\mathbf{CQ}[m]$ that mention only relation symbols that appear in D has cardinality at most $|D|^m \cdot f(k)$, for f a computable function and k the arity of the schema.

The proof of (2) can easily be obtained by combining ideas of the proofs of items (2) in Proposition 6.12 and 7.2. □

We conclude this part by observing that all our feasibility results are via constructive proofs that result in the proper statistic; hence, in the cases of tractable separability (and variants), both approximate feature generation and approximate classification, namely $\mathbf{CQ}[m]$-ApxCls, $\mathbf{CQ}[m]$-ApxCls[ℓ], and $\mathbf{CQ}[m]$-ApxCls[$*$], are FPT. The problem $\mathcal{L}$-ApxCls takes as input a number $\epsilon \in [0,1)$, a training database (D, λ) that is $\mathcal{L}$-separable with error ϵ, and an evaluation database D'. The goal is to construct a labeling λ' of D' such that there exists $(\Pi, \Lambda_{\bar{w}})$ that $\mathcal{L}$-separates (D', λ'), and at the same time, $\mathcal{L}$-separates (D, λ) with error ϵ. The problems $\mathcal{L}$-ApxCls[ℓ] and $\mathcal{L}$-ApxCls[$*$] are defined analogously.

7.2.2. Approximate $\mathbf{GHW}(k)$*-separability*

Now we look at approximate separability for statistics formed by CQs of bounded generalized hypertreewidth. Our main result is as follows.

Theorem 7.4. *Fix $k \geq 1$. There is a polynomial time algorithm that takes as input a training database (D, λ) and computes a labeling $\lambda' : \eta(D) \to \{1, -1\}$ such that:*

1. *(D, λ') is $\mathbf{GHW}(k)$-separable; and*
2. *for every $\lambda'' : \eta(D) \to \{1, -1\}$ such that (D, λ'') is $\mathbf{GHW}(k)$-separable, we have that $|\{e \in \eta(D) \mid \lambda(e) \neq \lambda'(e)\}| \leq |\{e \in \eta(D) \mid \lambda(e) \neq \lambda''(e)\}|$.*

Proof. Let (D, λ) be a given training database. For each $e \in \eta(D)$, we define $[e]$ to be the set of elements $e' \in \eta(D)$ such that $(D, e') \rightarrow_k (D, e)$ and $(D, e) \rightarrow_k (D, e')$. It is easy to see that the classes of the form $[e]$, for $e \in \eta(D)$, define a partition of $\eta(D)$. Define a new labeling $\lambda' : \eta(D) \rightarrow \{1, -1\}$ as follows:

$$\lambda'(e) := \begin{cases} 1 & \text{if } \sum_{e' \in [e]} \lambda(e') \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$

Due to Theorem 5.3, there is a polynomial-time algorithm that computes every $[e]$; therefore, λ' can be constructed in polynomial time. By its definition, each equivalence class $[e]$ is consistent with λ', that is, λ' maps all elements of $[e]$ to the same value. Hence, due to Lemma 5.4, it is the case that (D, λ') is **GHW**(k)-separable.

We will show that λ' is a best approximation of λ, in terms of the cardinality of the "disagreement," among the labelings λ'' of $\eta(D)$ such that (D, λ'') is **GHW**(k)-separable. Formally, this means that for every $\lambda'' : \eta(D) \rightarrow \{1, -1\}$ such that (D, λ'') is **GHW**(k)-separable, we have that $|\{e \in \eta(D) \mid \lambda(e) \neq \lambda'(e)\}| \leq |\{e \in \eta(D) \mid \lambda(e) \neq \lambda''(e)\}|$, or, equivalently, that $\sum_{e \in \eta(D)} |\lambda'(e) - \lambda(e)| \leq \sum_{e \in \eta(D)} |\lambda''(e) - \lambda(e)|$. We will show that this inequality holds, for all such λ'', already in each equivalence class $[e]$; that is, $\sum_{e' \in [e]} |\lambda'(e') - \lambda(e')| \leq \sum_{e' \in [e]} |\lambda''(e') - \lambda(e')|$.

So, let $\lambda'' : \eta(D) \rightarrow \{1, -1\}$ be such that (D, λ'') is **GHW**(k)-separable, and let $e \in \eta(D)$. Since λ' is consistent on $[e]$, either all $\lambda'(e')$ are $+1$ or all $\lambda'(e')$ are -1. Hence, either all $\lambda'(e') - \lambda(e')$ are nonnegative or all $\lambda'(e') - \lambda(e')$ are nonpositive. It follows that

$$\sum_{e' \in [e]} |\lambda'(e') - \lambda(e')| \; = \; \Big| \sum_{e' \in [e]} (\lambda'(e') - \lambda(e')) \Big| .$$

Analogously,

$$\sum_{e' \in [e]} |\lambda''(e') - \lambda(e')| \; = \; \Big| \sum_{e' \in [e]} (\lambda''(e') - \lambda(e')) \Big| .$$

So, we need to prove that

$$\Big| \sum_{e' \in [e]} (\lambda'(e') - \lambda(e')) \Big| \; \leq \; \Big| \sum_{e' \in [e]} (\lambda''(e') - \lambda(e')) \Big|$$

or, equivalently, that

$$\Big| \sum_{e' \in [e]} \lambda'(e') - \sum_{e' \in [e]} \lambda(e') \Big| \; \leq \; \Big| \sum_{e' \in [e]} \lambda''(e') - \sum_{e' \in [e]} \lambda(e') \Big| .$$

Let us define $x' = \sum_{e' \in [e]} \lambda'(e')$, define $x'' = \sum_{e' \in [e]} \lambda''(e')$, and define $y = \sum_{e' \in [e]} \lambda(e')$. We need to prove that $|x' - y| \leq |x'' - y|$. Since both λ' and λ'' are constant (either always 1 or always -1) on $[e]$, we have that $x' = x''$ or $x' = -x''$. In the first case, we are done. In the second one, we need to show that $|x' - y| \leq |-x' - y|$, i.e., $|x' - y| \leq |x' + y|$. But this is true for λ', as either both x' and y are nonnegative, or both x' and y are nonpositive. The pseudo-code of the procedure is given in Algorithm 3. □

Theorem 7.4 implies that **GHW**(k)-ApxSep and **GHW**(k)-ApxCls are tractable.

Corollary 7.5. *For all fixed $k \geq 1$, the problems* **GHW**(k)-ApxSep *and* **GHW**(k)-ApxCls *can be solved in polynomial time.*

Proof. Given a training database (D, λ), we apply Theorem 7.4 to compute in polynomial time a labeling $\lambda' : \eta(D) \rightarrow \{1, -1\}$ such that (D, λ') is **GHW**(k)-separable and λ' minimizes the disagreement with respect to λ, among those labelings λ'' such that (D, λ'') is **GHW**(k)-separable. Thus, the minimal error δ, for $0 \leq \delta \leq 1$, with which a statistic **GHW**(k)-separates (D, λ) is $(\{e \in \eta(D) \mid \lambda'(e) \neq \lambda(e)\})/|\eta(D)|$. Then in order to determine whether (D, λ) is separable with error ϵ, we simply check whether $\delta \geq \epsilon$.

To solve **GHW**(k)-ApxCls on an evaluation database D', we solve in polynomial time the problem **GHW**(k)-Cls on input given by training database (D, λ') and evaluation database D'. This generates a labeling λ^* of $\eta(D')$ such that there is a pair $(\Pi, \Lambda_{\bar{w}})$ that **GHW**(k)-separates both (D, λ') and (D', λ^*). Therefore, the pair $(\Pi, \Lambda_{\bar{w}})$ also **GHW**(k)-separates (D, λ) with error δ, and thus with error ϵ, and **GHW**(k)-separates (D', λ^*) with no error. □

8. More expressive feature queries

In this section, we embark on a preliminary exploration of the separability problem for more expressive feature languages, in particular First-Order Logic (FO) and some fragments thereof. While the problems have been discussed over CQs, they naturally extend to any query language $\mathcal{L}$, and we can talk about $\mathcal{L}$-separability and about $\mathcal{L}$-Sep for arbitrary fragments $\mathcal{L}$ of FO. We write **FO** when $\mathcal{L}$ is the class of all FO formulas.

Let us observe first that **FO**-separability is a more general notion than **CQ**-separability.

Algorithm 3 Approx-separability algorithm **GHW**(k)-ApxSep.

Require: A training database (D, λ)
1: $([e_1], \ldots, [e_m]) :=$ equivalence classes with respect to $\rightarrow_k$ over $\eta(D)$
2: **for** each $e \in \eta(D)$ **do**
3: **if** $\sum_{e' \in [e]} \lambda(e') \geq 0$ **then**
4: $\lambda'(e) = 1$
5: **else**
6: $\lambda'(e) = -1$
7: **end if**
8: **end for**
9: **return** $\lambda' : \eta(D) \rightarrow \{1, -1\}$

Example 8.1. Consider a schema that consists of a unary relation symbol A and a binary relation symbol E. Let D be a database over such a schema defined as

$$\{A(a), E(a,a), A(b), E(b,b), E(b,c)\}.$$

We define $\eta(D) = \{a, b\}$, $\lambda(a) = 1$, and $\lambda(b) = -1$. Clearly, (D, λ) is separable by the **FO**-formula $\phi(x) \equiv \exists y(y \neq x \wedge E(x, y))$. In turn, $(D, a) \rightarrow (D, b)$ and $(D, b) \rightarrow (D, a)$, and thus a and b are indistinguishable by CQs over D. Hence, (D, λ) is not **CQ**-separable. □

Next we show that **FO**-separability collapses to *single-feature* **FO**-separability.

Proposition 8.1. *A training database is* **FO***-separable iff it is* **FO***-separable by a statistics* Π *with a single* FO *formula.*

Proof. Let (D, λ) be an arbitrary fixed training database which is **FO**-separable by some pair $(\Pi, \Lambda_{\bar{w}})$ with $\Pi = (\phi_1, \ldots, \phi_n)$. Naturally, $\Lambda_{\bar{w}}$ can be seen as a boolean function mapping vectors from $\{-1, 1\}^n$ to $\{-1, 1\}$. Due to the **FO**-separability of (D, λ), we have that

$$\Pi^D(e) \neq \Pi^D(e'), \quad \text{for all } e, e' \in \eta(D) \text{ such that } \lambda(e) \neq \lambda(e').$$

Hence, there is a boolean combination $\Phi(x)$ of the ϕ_is in Π such that

$$\Phi(D) = S^+ = \{e \in \eta(D) \mid \lambda(e) = 1\}.$$

Since FO is closed under boolean combinations, Φ is an FO-formula. Thus, the statistics that consists exclusively of Φ **FO**-separates (D, λ). □

Hence, the complexity of separability for FO is the same regardless of whether the dimension of the statistic is bounded or not. That is, the complexity of the problems **FO**-Sep, **FO**-Sep[$*$], and **FO**-Sep[ℓ], for any $\ell \geq 1$, is the same. It can be proved, on the other hand, that the complexity of **FO**-Sep[1] coincides with that of QBE for FO (**FO**-QBE), as one can reduce in polynomial time from **FO**-Sep[1] to **FO**-QBE and, on the other hand, use **FO**-QBE as a subroutine to solve **FO**-Sep[1] in polynomial time. Arenas and Díaz [4] have shown that **FO**-QBE is GI-complete, where GI is the class of problems with a polynomial-time reduction to the graph-isomorphism problem. Therefore:

Corollary 8.2. *The problems* **FO**-Sep, **FO**-Sep[$*$], *and* **FO**-Sep[ℓ], *for any* $\ell \geq 1$, *are* GI*-complete.*

Proof. We have that **FO**-QBE and **FO**-Sep[1] are mutually reducible. In fact, let (D, λ) be some training instance. Recall that $S^+ = \{e \in \eta(D) \mid \lambda(e) = 1\}$ and $S^- = \{e \in \eta(D) \mid \lambda(e) = -1\}$. If there is a **FO**-explanation $q(x)$ for (D, S^+, S^-), then one can clearly use $q(x)$ as a single feature to separate S^+ and S^- in D. In turn, if (D, λ) is **FO**-separable by a statistic with a single feature query $q'(x)$, then either $S^+ \subseteq q'(D)$ and $S^- \cap q'(D) = \emptyset$, or $S^- \subseteq q'(D)$ and $S^+ \cap q'(D) = \emptyset$. In the first case $q'(x)$ is a **FO**-explanation for (D, S^+, S^-), while in the second case its negation $\neg q'(x)$ is a **FO**-explanation for (D, S^+, S^-). Together with Proposition 8.1 we get the stated complexity results. □

What about separability for fragments of FO? As we state next, **FO**-separability collapses to separability for statistics based on a simple class of formulas, namely, *existential* FO formulas, denoted $\exists$**FO**. Recall that these are the FO formulas of the form $\exists \bar{x} \psi$, where ψ is quantifier-free (but allows negation). On the other hand, for the restriction on $\exists$**FO** that disallows negation on ψ (the so-called class of *existential positive* FO formulas, written $\exists$**FO**$^+$), we have that separability collapses to **CQ**-separability. In summary:

Proposition 8.3. *The following statements hold for all training databases* (D, λ)*:*

1. (D, λ) *is* **FO***-separable iff it is* $\exists$**FO***-separable.*

2. (D, λ) *is* **CQ**-*separable iff it is* $\exists$**FO**$^+$-*separable.*

Proof. We start with (1). The right-to-left direction holds trivially. Assume now that (D, λ) is **FO**-separable. It is easy to see that this implies that there are no entities $e, e' \in \eta(D)$ with $\lambda(e) \neq \lambda(e')$ such that $(D, e) \cong (D, e')$. (Here $(D, e) \cong (D, e')$ denotes the existence of an *isomorphism* f from D to itself such that $f(e) = e'$). Assume otherwise. Then for every FO formula $\psi(x)$ it is the case that $D \models \psi(e) \Leftrightarrow D \models \psi(e')$. This implies that $\Pi^D(e) = \Pi^D(e')$ for every statistics formed exclusively by FO formulas, which contradicts the fact that (D, λ) is FO separable.

Let us define now a notion $(D, e) \Rightarrow (D', e')$, for databases D, D' and entities $e \in \eta(D)$ and $e' \in \eta(D')$, stating that there is a one-to-one homomorphism from D to D' that maps e to e'. It is well-known that the satisfaction of formulas in $\exists$**FO** is closed under the notion defined by $\Rightarrow$; that is, if $(D, e) \Rightarrow (D', e')$ then $e \in q(D)$ implies $e' \in q(D')$, for each FO formula $q(x)$ in $\exists$**FO**. In addition, when $D = D'$ the notion $(D, e) \Rightarrow (D, e')$ collapses to $(D, e) \cong (D, e')$.

Thus, from this observation and the condition expressed in the first paragraph, we obtain that there are no entities $e, e' \in \eta(D)$ with $\lambda(e) \neq \lambda(e')$ such that $(D, e) \Rightarrow (D, e')$. This implies that (D, λ) is $\exists$**FO**-separable. In fact, for each entity $f \in \eta(D)$ there is a formula $\psi_{D,f}(x)$ in $\exists$**FO** such that $D \models \psi_{D,f}(f')$ iff $(D, f) \Rightarrow (D, f')$. It is easy to see then that the statistic Π formed by all formulas of the form $\psi_{D,e}(x)$, for $e \in \eta(D)$ with $\lambda(e) = 1$, $\exists$**FO**-separates (D, λ). Indeed, we only need a linear classifier over Π that labels as positives those entities e such that $e \in q(D)$, for at least some query q in Π. This is because each $e \in \eta(D)$ with $\lambda(e) = 1$ satisfies at least one query in Π by definition, while each $e \in \eta(D)$ with $\lambda(e) = -1$ satisfies no query in Π. Assume otherwise, i.e., there is an $e \in \eta(D)$ with $\lambda(e) = -1$ such that e satisfies a query $q(x)$ in Π. By definition, q is of the form $\psi_{D,e'}(x)$, for $e' \in \eta(D)$ with $\lambda(e') = 1$. But then $(D, e) \Rightarrow (D, e')$, which is a contradiction since $\lambda(e) \neq \lambda(e')$.

Let us now prove (2). The direction from left-to-right holds trivially. Assume now that (D, λ) is not **CQ**-separable. It follows from [28] then that there are entities $e, e' \in \eta(D)$ such that $\lambda(e) \neq \lambda(e')$, but e and e' are "indistinguishable" by CQs. The latter implies that both $(D, e) \to (D, e')$ and $(D, e') \to (D, e)$ hold. It is well-known on the other hand that existential positive FO formulas are preserved by homomorphisms; in particular, if $(D, e) \to (D, e')$ then $e \in q(D)$ implies $e' \in q(D)$, for each FO formula $q(x)$ in $\exists$**FO**$^+$. But then the entities e and e' are also "indistinguishable" by formulas in $\exists$**FO**$^+$, which implies that (D, λ) is not $\exists$**FO**$^+$-separable. □

Therefore, from Proposition 8.3 and Corollary 8.2 we obtain that $\mathcal{L}$-Sep is GI-complete for any fragment $\mathcal{L}$ of FO that contains $\exists$**FO**, and from Theorem 3.1 that $\exists$**FO**$^+$-Sep is coNP-complete.

As we have seen, there is an important difference between **FO**-separability and **CQ**-separability: The former collapses to single-feature **FO**-separability from Proposition 8.3, while for the latter there is no bound on the number of features which are required for separating training databases (recall that the same holds for **GHW**(k), for $k \geq 1$, from Theorem 5.7). This motivates the two questions we study next about feature languages $\mathcal{L}$.

1. When does $\mathcal{L}$ have the *dimension-collapse property*, i.e., every training database (D, λ) that is $\mathcal{L}$-separable is also separable by a single-feature statistic in $\mathcal{L}$?
2. In turn, when does $\mathcal{L}$ have the *unbounded-dimension property*, that is, for all $n \geq 1$ there is a training database (D, λ) that is $\mathcal{L}$-separable only by statistics with at least n features?

8.1. The dimension-collapse property

We have seen in Proposition 8.1 that **FO** has the dimension-collapse property. In contrast, we can show that none of **CQ**, **GHW**(k) and $\exists$**FO**$^+$ have this property. Next, we present a general explanation of this fact by providing a characterization of when a query language $\mathcal{L}$ has the dimension-collapse property in terms of a certain definability condition.

Theorem 8.4. *$\mathcal{L}$ has the dimension-collapse property if and only if for every database D, the set $\bigcup_{q \in \mathcal{L}} \{q(D), \eta(D) \setminus q(D)\}$ of entity sets is closed under intersection.*

Proof. Assume that $\mathcal{L}$ has the dimension-collapse property, i.e., each training database (D, λ) that is $\mathcal{L}$-separable is also separable by a statistics with a single feature in $\mathcal{L}$. We prove by contradiction.

Let D be a database such that $X = \bigcup_{q \in \mathcal{L}} \{q(D), \eta(D) \setminus q(D)\}$ is not closed under intersection. It follows that there are queries $q_1 \in X$, $q_2 \in X$ such that neither $q_1(D) \cap q_2(D) \in X$ nor $\eta(D) \setminus (q_1(D) \cap q_2(D)) \in X$. We define $\lambda \colon \eta(D) \to \{-1, 1\}$ in the following way:

$$\lambda(e) := \begin{cases} +1 & \text{if } e \in q_1(D) \text{ and } e \in q_2(D), \\ -1 & \text{otherwise.} \end{cases}$$

The training database (D, λ) is separable by $\Pi = (q_1, q_2)$ where $w_1 = w_2 = 1$ are the weights. Since neither $q_1(D) \cap q_2(D)$ nor $\eta(D) \setminus (q_1(D) \cap q_2(D))$ can be expressed, there is no statistic of length 1 that separates (D, λ).

On the other hand, assume that for each database D the set $X = \bigcup_{q \in \mathcal{L}} \{q(D), \eta(D) \setminus q(D)\}$ is closed under intersection. (Note that since X is closed under complementation (wrt. to $\eta(D)$) and intersection, it is also closed under union.) Let

(D, λ) be an arbitrary fixed training database which is $\mathcal{L}$-separable by some statistic $\Pi = (q_1, \ldots, q_n)$. As before, $\Lambda_{\bar{w}}$ can be seen as a boolean function mapping vectors from $\{-1, 1\}^n$ to $\{-1, 1\}$. By $\mathcal{L}$-separability, for all $e, e' \in \eta(D)$ such that $\lambda(e) \neq \lambda(e')$ holds that $\Pi^D(e) \neq \Pi^D(e')$. Since set intersection, set complement, and set union correspond to conjunction, negation, and disjunction in the boolean algebra, it follows that the set, S, containing all entities labeled positively belongs to X. Hence, there is some $q \in \mathcal{L}$ such that $q(D) = S$ or $\eta(D) \setminus S$. In consequence, a statistic containing only q separates (D, λ). □

Applying this characterization, one can readily see that not only **FO**, but also $\mathbf{FO}_k$, the fragment of formulas with at most k variables, has the dimension-collapse property. It is possible to prove, on the other hand, that the dimension-collapse property also holds for every class Σ_k, for $k \geq 1$, that consists of all FO queries of the form $\exists \bar{x}_1 \forall \bar{x}_2 \ldots \mathcal{Q} x_n \psi$, where ψ is quantifier-free and $\mathcal{Q} = \exists$ if n is odd and $\mathcal{Q} = \forall$ otherwise. Notice that Σ_1 is precisely $\exists$**FO**.

Corollary 8.5. *The languages* **FO**, $\mathbf{FO}_k$, *and* Σ_k, *for any* $k \geq 1$, *have the dimension-collapse property.*

Proof. It is clear that **FO** and $\mathbf{FO}_k$ satisfy the condition expressed in Theorem 8.4. For Σ_k we first observe the following: If there is an **FO**-explanation for a tuple (D, S^+, S^-), for unary relations S^+ and S^- over D, then there is also a Σ_1-explanation. In fact, it is easy to see that if there is a **FO**-explanation then there is one of the form $\bigvee_{a \in S^+} \psi_{D,a}(x)$, where $\psi_{D,a}(x)$ is the FO formula that describes the "isomorphism type" of a over D. This means that $D \models \psi_{D,a}(b) \Leftrightarrow (D, a) \cong (D, b)$ for every $b \in \mathsf{dom}(D)$. On the other hand, it is clear that each $\psi_{D,a}(x)$ can be expressed as a formula in Σ_1. For instance, if $D = \{R(a, b), S(b)\}$ then

$$\psi_{D,a}(x) = \exists y \left(x \neq y \wedge R(x, y) \wedge S(y) \wedge \neg S(x) \right).$$

Let us assume then that (D, λ) is Σ_k-separable. Then there is a statistic with at most one FO feature query that **FO**-separates (D, λ). That is, either there is an **FO**-explanation for (D, S^+, S^-), where $S^+ = \{e \in \eta(D) \mid \lambda(e) = 1\}$ and $S^- = \eta(D) \setminus S^+$, or there is an **FO**-explanation for (D, T^+, T^-), where $T^+ = \{e \in \eta(D) \mid \lambda(e) = -1\}$ and $T^- = \eta(D) \setminus T^+$. From our previous remarks, in any case there is also a statistic with at most one Σ_k feature query, for $k \geq 1$, that Σ_k-separates (D, λ). □

In contrast, neither **CQ** nor **GHW**(k), for any $k \geq 1$, satisfy the condition of Theorem 8.4. This is also the case for Σ_k^+, the restriction of Σ_k where no negation is allowed in the quantifier-free formula ψ. We actually prove a stronger statement below: All of these languages have the unbounded-dimension property.

8.2. The unbounded-dimension property

We provide a simple condition that ensures the unbounded-dimension property for a language $\mathcal{L}$. A family $\mathcal{S}$ of sets is *linear* if $A \subseteq B$ or $B \subseteq A$, for every $A, B \in \mathcal{S}$.

Proposition 8.6. *Assume that for each* $n \geq 1$ *there is a database* D *such that* $\{q(D) \mid q \in \mathcal{L}\}$ *is linear and has cardinality at least* n. *Then* $\mathcal{L}$ *has the unbounded-dimension property.*

Proof. Let $n \geq 1$ and assume that D satisfies the hypothesis. We can assume, by choosing appropriately $\eta(D)$, that $\eta(D) = \{a_1, \ldots, a_n\}$ and that $\{q(D) \mid q \in \mathcal{L}\}$ contains precisely all sets of the form $\{a_1, \ldots, a_i\}$, for $i = 1, \ldots, n$. For every $i = 1, \ldots, n$, let q_i be a feature query in $\mathcal{L}$ satisfying $q(D) = \{a_1, \ldots, a_i\}$ and let $\Pi = (q_1, \ldots, q_{n-1})$.

Let $\lambda : \eta(D) \to \{-1, +1\}$ be the labeling defined as

$$\lambda(a_i) = (-1)^i, \quad \text{for } i = 1, \ldots, n.$$

Observe that $(\Pi, \Lambda_{\bar{w}})$ separates (D, λ) if $\bar{w} = (0, -1, 1, -1, +1, \ldots, (-1)^{n-1})$. Also, note that the assumption on $\{q(D) \mid q \in \mathcal{L}\}$ and the fact that q_n is redundant (i.e., $q_n(D) = \eta(D)$) implies that we can assume that every statistic $\mathcal{L}$-separating (D, λ) contains only feature queries from Π. We shall complete the proof by showing that the statistic Π' obtained by removing any query q_i from Π does not separate (D, λ). This follows from the fact that $a_i \in q_j(D) \Leftrightarrow a_{i+1} \in q_j(D)$, for every $j \in \{1, \ldots, i-1, i+1, \ldots, n\}$ (and hence, Π' could not 'distinguish' between a_i and a_{i+1}), but $\lambda(a_i) \neq \lambda(a_{i+1})$. □

We can show that all of the above languages satisfy the condition expressed in Proposition 8.6. Correspondingly, they all have the unbounded-dimension property.

Theorem 8.7. *The languages* **CQ**, **GHW**(k) *and* Σ_k^+, *for any* $k \geq 1$, *have the unbounded-dimension property.*

Problem	*Class* $\mathcal{L}$	*Result*	*Reference*
$\mathcal{L}$-Sep	**CQ**	coNP-complete	[28]
	CQ$[m]$	?	–
		FPT w.r.t. the *arity* of the schema	Corollary 4.2
	CQ$[m, p]$	PTIME	Proposition 4.3
	GHW(k)	PTIME	Theorem 5.3
	FO	GI-complete	Corollary 8.2
	$\exists$**FO**	GI-complete	Proposition 8.3
	$\exists$**FO**$^+$	coNP-complete	Proposition 8.3
$\mathcal{L}$-Sep$[*]$	**CQ**	coNEXPTIME-complete	Theorem 6.5
	CQ$[m]$	NP-complete for fixed *arity* schemas	Proposition 6.9
		FPT w.r.t. the *size* of the schema	Proposition 6.8
	CQ$[m, p]$	NP-complete	Proposition 6.12
	GHW(k)	EXPTIME-complete	Theorem 6.5
	FO	GI-complete	Corollary 8.2
$\mathcal{L}$-Sep$[\ell]$	**CQ**	coNEXPTIME-complete	Lemma 6.4
	CQ$[m]$	NP-complete	Theorem 6.10
		FPT w.r.t. the *arity* of the schema	Proposition 7.3
	CQ$[m, p]$	PTIME	Proposition 6.12
	GHW(k)	EXPTIME-complete	Lemma 6.4
	FO	GI-complete	Corollary 8.2

Fig. 3. Overview of complexity results for the separability problems studied in the paper.

Proof. In order to show that **CQ** has the unbounded feature property we only need to define, for every $n \geq 1$, a structure D satisfying the hypothesis of Proposition 8.6. This is done as follows: $\mathsf{dom}(D) = \eta(D) = \{a_1, \ldots, a_n\}$. The schema of D has n unary symbols $\kappa_1, \ldots, \kappa_n$ and facts $\kappa_j(a_i)$ for every $1 \leq i \leq j \leq n$. It is immediate to see that $\{q(D) \mid q \in \mathbf{CQ}\}$ contains precisely the sets of the form $\{a_1, \ldots, a_i\}$, for $i = 0, \ldots, n$. The same holds if we replace **CQ** by **GHW**(k) or Σ_k^+ for any $k \geq 1$. □

9. Final remarks

We studied the separability problem for CQ features under various regularizations by posing upper bounds on the number of atoms per CQ, the ghw of CQs, and the dimension of (i.e., number of features in) the statistic. When the tractability proofs are *constructive*, tractability extends to the problems of feature generation and classification of an evaluation database. This is not the case for the class of CQs of a bounded ghw where the feature CQs might be overly large to materialize; yet, we showed that classification is then tractable even without materializing the feature CQs. We also proved that our complexity results extend to approximate separability, though some of our proofs require nontrivial adjustments. Finally, we gave preliminary results on separability with more expressive languages of feature queries, such as FO, and particularly, about when separability collapses to restricted fragments and a bounded number of feature queries (and even a single one).

An overview of the complexity of the different versions of the separability problem studied in the paper is shown in Fig. 3. For the reader convenience we recall some of the abbreviations involved:

- $\mathcal{L}$-Sep: Separability problem of unbounded dimension for some class $\mathcal{L}$ of queries.
- $\mathcal{L}$-Sep$[\ell]$: Separability problem of fixed dimension ℓ for some class $\mathcal{L}$ of queries.
- $\mathcal{L}$-Sep$[*]$: Separability problem where the dimension is given as input for some class $\mathcal{L}$ of queries.
- **CQ**$[m]$: Class of CQs with at most m atoms, where m is fixed.
- **CQ**$[m, p]$: Class of CQs with at most m atoms and in which each variable occurs at most p times, where m and p are fixed.

We would like to remark that for evert version of the separability that can be solved in PTIME or is FPT, then the same holds for the corresponding classification problem.

Future work An immediate open problem is the complexity of separability for a bounded number of CQ atoms, that is, **CQ**$[m]$-Sep for any fixed $m \geq 1$, when the schema is given as part of the input with no restrictions. An important direction for future work is the extension of the results on generalized hypertree width to more general structural restrictions on CQs that continue to ensure tractability of evaluation. These include CQs of bounded *fractional hypertree width* [19] and *submodular width* [32]. Finally, an important direction is the treatment of feature generation over databases through the lens of PAC learning, for instance, by adopting the concepts of Grohe et al. [20,18].

CRediT authorship contribution statement

All authors of this paper contributed in equal terms to it. This is represented in the alphabetical order of the names in the author's list.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Barceló is funded by Fondecyt grant 1200967 and the Millennium Institute for Foundational Research on Data (IMFD Chile). Baumgartner is funded by Fondecyt grant 11191097.

References

[1] F. Ahmed, M. Samorani, C. Bellinger, O.R. Zaïane, Advantage of integration in big data: feature generation in multi-relational databases for imbalanced learning, in: BigData, IEEE, 2016, pp. 532–539.
[2] E. Alpaydin, Introduction to Machine Learning, MIT Press, 2009.
[3] T. Antonopoulos, F. Neven, F. Servais, Definability problems for graph query languages, in: ICDT, 2013, pp. 141–152.
[4] M. Arenas, G.I. Diaz, The exact complexity of the first-order logic definability problem, ACM Trans. Database Syst. 41 (2016) 13.
[5] B. Aronov, D. Garijo, Y. Núñez-Rodríguez, D. Rappaport, C. Seara, J. Urrutia, Minimizing the error of linear separators on linearly inseparable data, Discrete Appl. Math. 160 (2012) 1441–1452.
[6] P. Barceló, M. Romero, The complexity of reverse engineering problems for conjunctive queries, in: ICDT, 2017, 7.
[7] A. Bonifati, W. Martens, T. Timm, An analytical study of large SPARQL query logs, Proc. VLDB Endow. 11 (2017) 149–161.
[8] R. Cappuzzo, P. Papotti, S. Thirumuruganathan, Creating embeddings of heterogeneous relational datasets for data integration tasks, in: D. Maier, R. Pottinger, A. Doan, W. Tan, A. Alawini, H.Q. Ngo (Eds.), SIGMOD, 2020, pp. 1335–1349.
[9] B. ten Cate, V. Dalmau, The product homomorphism problem and applications, in: ICDT, 2015, pp. 161–176.
[10] O. Chapelle, P. Haffner, V. Vapnik, Support vector machines for histogram-based image classification, IEEE Trans. Neural Netw. 10 (1999) 1055–1064.
[11] H. Chen, V. Dalmau, Beyond hypertree width: decomposition methods without decompositions, in: CP, 2005, pp. 167–181.
[12] V. Feldman, V. Guruswami, P. Raghavendra, Y. Wu, Agnostic learning of monomials by halfspaces is hard, SIAM J. Comput. 41 (2012) 1558–1590.
[13] J. Flum, M. Grohe, Parameterized Complexity Theory, Springer, 2006.
[14] J. Friedman, T. Hastie, R. Tibshirani, Sparse inverse covariance estimation with the graphical lasso, Biostatistics 9 (2008) 432–441.
[15] G. Gottlob, G. Greco, N. Leone, F. Scarcello, Hypertree decompositions: questions and answers, in: PODS, 2016, pp. 57–74.
[16] G. Gottlob, N. Leone, F. Scarcello, Hypertree decompositions and tractable queries, J. Comput. Syst. Sci. 64 (2002) 579–627.
[17] M. Grohe, word2vec, node2vec, graph2vec, x2vec: towards a theory of vector embeddings of structured data, in: D. Suciu, Y. Tao, Z. Wei (Eds.), SIGMOD, 2020, pp. 1–16.
[18] M. Grohe, C. Löding, M. Ritzert, Learning mso-definable hypotheses on strings, in: ALT, PMLR, 2017, pp. 434–451.
[19] M. Grohe, D. Marx, Constraint solving via fractional edge covers, ACM Trans. Algorithms 11 (2014) 4.
[20] M. Grohe, M. Ritzert, Learning first-order definable concepts over structures of small degree, in: LICS, IEEE Computer Society., 2017, pp. 1–12.
[21] I. Guyon, S. Gunn, M. Nikravesh, L.A. Zadeh, Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing), Springer-Verlag, New York, Inc., Secaucus, NJ, USA, 2006.
[22] K. Höffgen, H.U. Simon, K.S.V. Horn, Robust trainability of single neurons, J. Comput. Syst. Sci. 50 (1995) 114–125.
[23] S. Kandel, A. Paepcke, J.M. Hellerstein, J. Heer, Enterprise data analysis and visualization: an interview study, IEEE Trans. Vis. Comput. Graph. 18 (2012) 2917–2926.
[24] N. Karmarkar, A new polynomial-time algorithm for linear programming, Combinatorica 4 (1984) 373–396.
[25] S.M. Kazemi, B. Fatemi, A. Kim, Z. Peng, M.R. Tora, X. Zeng, M.C. Dirks, D. Poole, Comparing aggregators for relational probabilistic models, CoRR, arXiv:1707.07785 [abs], 2017.
[26] J.M. Keller, M.R. Gray, J.A. Givens, A fuzzy k-nearest neighbor algorithm, IEEE Trans. Syst. Man Cybern. Syst. 15 (1985) 580–585.
[27] L. Khachiyan, A polynomial algorithm in linear programming, Sov. Math. Dokl. 20 (1979) 191–194.
[28] B. Kimelfeld, C. Ré, A relational framework for classifier engineering, in: PODS, 2017, pp. 5–20.
[29] A.J. Knobbe, M. de Haas, A. Siebes, Propositionalisation and aggregates, in: PKDD, 2001, pp. 277–288.
[30] P.G. Kolaitis, J. Panttaja, On the complexity of existential pebble games, in: CSL, 2003, pp. 314–329.
[31] H.T. Lam, T.N. Minh, M. Sinn, B. Buesser, M. Wistuba, Learning features for relational data, CoRR, arXiv:1801.05372 [abs], 2018, URL: http://arxiv.org/abs/1801.05372.
[32] D. Marx, Tractable hypergraph properties for constraint satisfaction and conjunctive queries, J. ACM 60 (2013) 42.
[33] M. Mohri, A. Rostamizadeh, A. Talwalkar, Foundations of Machine Learning, MIT Press, 2012.
[34] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute, V. Raghavendra, Deep learning for entity matching: a design space exploration, in: G. Das, C.M. Jermaine, P.A. Bernstein (Eds.), SIGMOD, 2018, pp. 19–34.
[35] C.A. Murthy, Bridging feature selection and extraction: compound feature generation, IEEE Trans. Knowl. Data Eng. 29 (2017) 757–770.
[36] C. Perlich, F.J. Provost, Distribution-based aggregation for relational learning with identifier attributes, Mach. Learn. 62 (2006) 65–105.
[37] M. Pontil, A. Verri, Support vector machines for 3D object recognition, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 637–646.
[38] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?": explaining the predictions of any classifier, in: KDD, ACM, 2016, pp. 1135–1144.
[39] M.T. Ribeiro, S. Singh, C. Guestrin, Anchors: high-precision model-agnostic explanations, in: AAAI, AAAI Press, 2018, pp. 1527–1535.
[40] M. Samorani, M. Laguna, R.K. DeLisle, D.C. Weaver, A randomized exhaustive propositionalization approach for molecule classification, INFORMS J. Comput. 23 (2011) 331–345.
[41] B. Schölkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, Adaptive Computation and Machine Learning Series, MIT Press, 2002.
[42] S. Shalev-Shwartz, S. Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, New York, NY, USA, 2014.
[43] Y.Y. Weiss, S. Cohen, Reverse engineering spj-queries from examples, in: PODS, 2017, pp. 151–166.
[44] R. Willard, Testing expressibility is hard, in: CP, 2010, pp. 9–23.
[45] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, A comprehensive survey on graph neural networks, CoRR arXiv:1901.00596 [abs], 2019, URL: http://arxiv.org/abs/1901.00596.
[46] C. Zhang, A. Kumar, C. Ré, Materialization optimizations for feature selection workloads, in: SIGMOD Conference, 2014, pp. 265–276.
[47] W. Ziarko, Variable precision rough set model, J. Comput. Syst. Sci. 46 (1993) 39–59.