Regularizing Conjunctive Features for Classification

Pablo Barceló
DCC, Univ. of Chile & IMFD Chile

Alexander Baumgartner
DCC, Univ. of Chile & RISC, Johannes Kepler Univ.

Victor Dalmau
Universitat Pompeu Fabra

Benny Kimelfeld
Technion – Israel Institute of Technology

ABSTRACT

We consider the feature-generation task wherein we are given a database with entities labeled as positive and negative examples, and the goal is to find feature queries that allow for a linear separation between the two sets of examples. We focus on conjunctive feature queries, and explore two fundamental problems: (a) deciding whether separating feature queries exist (separability), and (b) generating such queries when they exist. In the approximate versions of these problems, we allow a predefined fraction of the examples to be misclassified. To restrict the complexity of the generated classifiers, we explore various ways of regularizing (i.e., imposing simplicity constraints on) them by limiting their dimension, the number of joins in feature queries, and their generalized hypertree width (ghw). Among other results, we show that the separability problem is tractable in the case of bounded ghw; yet, the generation problem is intractable, simply because the 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 ever explicitly generating the feature queries.

ACM Reference Format:
Pablo Barceló, Alexander Baumgartner, Victor Dalmau, and Benny Kimelfeld. 2019. Regularizing Conjunctive Features for Classification. In *38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS'19), June 30-July 5, 2019, Amsterdam, Netherlands.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3294052.3319680

1 INTRODUCTION

Feature engineering is a critical and resource-consuming task in the development of machine-learning solutions in general, and classifiers in particular [16, 18, 34]. In the framework proposed by Kimelfeld and Ré [22], the general goal is to utilize the database's knowledge of the raw data structure 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 [22]. 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 of 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 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 [22], 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, 25]), 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, 24, 27, 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 restricted queries that are useful as features in the sense that they provide (approximate) separation.

PODS'19, June 30-July 5, 2019, Amsterdam, Netherlands

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 [22]. 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). In machine learning, complexity restriction and reduction for learned models is known as *regularization* [30, 31]. Finally, complex queries are complicated to interpret and manipulate by human engineers.

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. These restrictions 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 [9]. 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, 24, 27, 29]. 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) [11, 26]. We also study the complexity of combining the bound on the dimension with the bounds on the CQ features.

As said, in the absence of any restriction, the separability problem is coNP-complete even for a fixed schema [22]. 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 [10] (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), 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* [9].

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) [3, 32, 33]: 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 $\mathbf{CQ}$), 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).

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

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-c. [22]	PTIME	PTIME
$\mathcal{L}$-Sep$[\ell]$	coNEXPTIME-c.	PTIME	EXPTIME-c.

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, 25]). Moreover, in some cases, such as Lemma 5.8 of Kimelfeld and Ré [23], 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.

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. Due to space constraints some proofs are in the appendix.

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.

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 conventional, 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, known as its *canonical database*. In particular, if q is of the form (1), then $D_q = \{R_1(\bar{x}_1), \ldots, R_n(\bar{x}_n)\}$ is the database that 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, denoted $q(D)$, is the set of all tuples $\bar{a}$ over $\mathsf{dom}(D)$ such that $(D_q, \bar{x}) \to (D, \bar{a})$. 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

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

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 [22], which recall next.

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 $e \in \eta(D)$:

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

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 $\Lambda_{\bar{w}}(\Pi^D(e)) = \lambda(e)$ 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 $\mathcal{L}$-separates (D, λ) if $\Lambda_{\bar{w}}$ is irrelevant.

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 [22], in the problem $\mathcal{L}$-Sep the goal is to determine the existence of a separating statistic Π in $\mathcal{L}$.

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 3.2. [22] *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.*

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* [10], by fixing the number of atoms allowed in feature CQs.

For every fixed $m \ge 1$, 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. *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 is bounded by $|D|^c \cdot 2^{q(k)}$ for a constant $c \ge 1$ and polynomial $q : \mathbb{N} \to \mathbb{N}$, where $k \ge 1$ is the maximal arity of a relation in D.*

Proof. Observe that (D, λ) is **CQ**$[m]$-classifiable iff it is classifiable by the statistic Π that contains all feature queries $q(x)$ in **CQ**$[m]$ that mention only relation symbols that appear in D. Since m is fixed, the statistic Π can be computed

in time $r^m \cdot 2^{p(k)}$, for p a polynomial, where r is the number of relation symbols mentioned in D. (Thus, $r^m \cdot 2^{p(k)}$ is also a bound for the number of different feature CQs in Π). Since $r \leq |D|$, the latter is also $|D|^m \cdot 2^{p(k)}$. 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 $O(|D|^{2m+1} \cdot 2^{p(k)})$, which is $|D|^{2m+1} \cdot 2^{p'(k)}$, for some polynomial p'.

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 [19, 21]). 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 $\mathbf{CQ}[m]$-separable, and, if so, computing a pair $(\Pi, \Lambda_{\bar{w}})$ that $\mathbf{CQ}[m]$-separates (D, λ), can be done in time $|D|^c \cdot 2^{q(k)}$ for a constant $c \geq 1$ and polynomial q. This concludes the proof. □

From Proposition 4.1, the problem $\mathbf{CQ}[m]$-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 $\mathbf{CQ}[m]$-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 3.2 states that $\mathbf{CQ}$-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 ≠ NP).

It remains an open problem whether $\mathbf{CQ}[m]$-Sep is NP-hard for some fixed $m \geq 1$. 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 $\mathbf{CQ}[m, p]$ be the class of CQs with at most m atoms and in which each variable occurs at most p times. Then:

Proposition 4.3. $\mathbf{CQ}[m, p]$-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 $\mathbf{CQ}[m]$-separable.

- *Feature generation:* Construct a pair $(\Pi, \Lambda_{\bar{w}})$ that $\mathbf{CQ}[m]$-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 GEN. HYPERTREE-WIDTH

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

We introduce the classes of CQs of bounded *generalized hypertree-width* [13] (also known as *coverwidth* [9]). We adopt the definition of Chen and Dalmau [9], 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 $\mathbf{GHW}(k)$ the class of CQs of ghw at most k. In contrast to the case of general CQs, the evaluation problem for CQs in $\mathbf{GHW}(k)$ can be solved in polynomial time [12]. Notice that each CQ in $\mathbf{CQ}[k]$ is also in $\mathbf{GHW}(k)$, but not viceversa.

The existential cover game. There is a link between the evaluation of CQs in $\mathbf{GHW}(k)$ and a version of the pebble game, known as *existential cover game* [9], 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:

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

These approximations 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).

Proposition 5.1. [9] *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 $\mathbf{GHW}(k)$.

Proposition 5.2. [9] *$(D, \bar{a}) \to_k (D', \bar{b})$ if and only if for every CQ $q(\bar{x})$ in $\mathbf{GHW}(k)$ 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 $\mathbf{GHW}(k)$, databases D and tuples $\bar{a}$, it holds that $\bar{a} \in q(D)$ if and only if $(q, \bar{x}) \to_k (D, \bar{a})$.

5.1 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 $\mathbf{GHW}(k)$-Sep is solvable in polynomial time.*

The proof is based on the following lemma.

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

(1) (D, λ) is $\mathbf{GHW}(k)$-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 straighforward. We now prove that $2 \to 1$. For each $e \in \eta(D)$, we define a query

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

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 $e \in E_i$: *(a)* $e \in q_{e_i}(D)$, *and (b)* $e \notin q_{e_j}(D)$ for each $j \in \{1, \ldots, m\}$ with $i < j$. It follows from Kimelfeld and Ré [22] 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:

> **$\mathbf{GHW}(k)$-*separability test:***
> Given a training database (D, λ), 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 $\mathbf{GHW}(k)$.

5.2 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 [9] 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 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. *For all $n, m \geq 1$ there is a training database (D, λ) with $|D| = O(n)$ and $|\eta(D)| = O(m)$ 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.*

We are thus faced with an apparently contradictory situation: while we can efficiently check for the existence of a statistic that $\mathbf{GHW}(k)$-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.

5.3 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 5.8. $\mathbf{GHW}(k)$-Cls *can be solved in polynomial time for all fixed $k \geq 1$.*

Proof. Consider an input for $\mathbf{GHW}(k)$-Cls that consists of a $\mathbf{GHW}(k)$-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 $\mathbf{GHW}(k)$-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 $q(x) \in \mathbf{GHW}(k)$ 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 $\mathbf{GHW}(k)$-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, therefore, 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 [22] 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 $\preceq$.

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, $(\Pi, \Lambda_{\bar{w}})$ $\mathbf{GHW}(k)$-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 $(D, e_i) \to_k (D', f)$. The pseudo-code of the procedure is shown in Algorithm 1. □

Algorithm 1 Classification algorithm $\mathbf{GHW}(k)$-Cls.

Require: An $\mathbf{GHW}(k)$-separable training database (D, λ) and an evaluation database D'

1: $([e_1], \ldots, [e_m]) :=$ topological sort of the equivalence classes defined by $\to_k$ 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** $(D, e_i) \to_k (D', f)$ **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** $\sum_{1 \le i \le m} w_i \cdot \mathbb{1}_{q_{e_i}(D')}(f) \ge w_0$ **then**
12: $\lambda'(f) = 1$
13: **else**
14: $\lambda'(f) = -1$
15: **end if**
16: **end for**
17: **return** $\lambda' : \eta(D') \to \{1, -1\}$

In summary, in this section we have established that it takes polynomial time to decide whether a given training database (D, λ) is $\mathbf{GHW}(k)$-separable (Theorem 5.3). A 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 5.8).

6 BOUNDING THE DIMENSION

While the separability and classification problems become tractable if we restrict to statistics formed by CQs in $\mathbf{GHW}(k)$, for each fixed $k \ge 1$, 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 entities in $\eta(D)$. This problem is not exclusive to the class $\mathbf{GHW}(k)$; 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 $\ell \ge 1$, 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$[\ell]$.

As we show next, the study of $\mathcal{L}$-Sep$[*]$ and $\mathcal{L}$-Sep$[\ell]$ is directly related to *query-by-example* problem (QBE). This allow us to apply the wide arsenal of results and tools for QBE [6, 32, 33] in order to understand the complexity of $\mathcal{L}$-Sep$[*]$. We first introduce QBE.

6.1 The Query-by-Example Problem

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 $q(D) \cap S^- = \emptyset$. Then QBE for the class $\mathcal{L}$ is 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. [6, 32, 33] *The following statements hold:*

- **CQ**-QBE *is* coNEXPTIME*-complete.*
- $\mathbf{GHW}(k)$-QBE *is* EXPTIME*-complete, for each* $k \ge 1$.

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

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 $\lambda(e) \ne \lambda(e')$, yet e and e' are "indistinguishable" by CQs [22]. As the next example shows, this does not hold under the current restriction on the dimension of the statistic.

Example 6.2. 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 $\lambda(a) = \lambda(b) = 1$ 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.

$(\mathcal{L}, \ell)$*-separability test:*

Given a training database (D, λ),

- Guess a vector $\bar{\kappa}_e \in \{1, -1\}^\ell$, for each $e \in \eta(D)$;
- Check $(\bar{\kappa}_e, \lambda(e))_{e \in \eta(D)}$ for linear separability;
- For all $j \in \{1, \ldots, \ell\}$, test whether an $\mathcal{L}$-explanation for (D, S_j^+, S_j^-) exists, where $S_j^+ = \{e \mid \bar{\kappa}_e[j] = 1\}$ and $S_j^- = \{e \mid \bar{\kappa}_e[j] = -1\}$.

Here, $\bar{\kappa}_e[j]$ 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 6.3. *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).*

It is not hard to see, by applying Theorem 6.1, that for every $\ell \geq 1$ the $(\mathbf{CQ}, \ell)$-separability test can be carried out in coNEXPTIME, while for every fixed $k \geq 1$, the $(\mathbf{GHW}(k), \ell)$-separability test can be carried out in EXPTIME. Then, from Lemma 6.3 we obtain an upper bound for the complexity of $\mathbf{CQ}$-Sep$[*]$ and $\mathbf{GHW}(k)$-Sep$[*]$, for every $k \geq 1$.

Proposition 6.4. $\mathbf{CQ}$-Sep$[*]$ *is in* coNEXPTIME*, while the problem* $\mathbf{GHW}(k)$-Sep$[*]$ *is in* EXPTIME *for every* $k \geq 1$.

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 6.5. *Let $\mathcal{L}$ be a class of CQs such that $\mathcal{L}$ contains all CQs with only one atom (over every schema). Fix $\ell \geq 1$. Then $\mathcal{L}$-*QBE*, when restricted to inputs of the form (D, S^+, S^-) where S^+, S^- are nonempty unary relations such that $S^- = \mathrm{dom}(D) \setminus S^+$, reduces in polynomial time to $\mathcal{L}$-*Sep$[\ell]$.

Proof. Let D be a database over some schema σ and assume that S^+, S^- are nonempty unary relations over D such that $S^- = \mathrm{dom}(D) \setminus S^+$. 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

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

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

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 $\bar{w} = (1 - l, 1, \ldots, 1)$. In fact, notice that for each $e \in \eta(D') = \mathrm{dom}(D')$ it is the case that $\Lambda_{\bar{w}}(\Pi^{D'}(e)) = 1$ iff e belongs to the evaluation of at least some CQ in Π over D'. Let us first consider the elements e with $\lambda(e) = 1$. If $e = c_i$, for $i \in \{1, \ldots, \ell - 1\}$, then $e \in q_i(D')$. If $e \in S^+$ then $e \in q(D')$. Consider now an arbitrary element e in S^-. Then $e \notin q_i(D)$, for each $i \in \{1, \ldots, \ell - 1\}$, since $q_i(x) = \kappa_i(x)$. Also, $e \notin q(D')$. This is because the restriction of D' to the symbols in σ homomorphically maps to D, and, thus, if $e \in q(D')$ we would have that $e \in q(D)$. This contradicts the fact that $q(D) \cap S^- = \emptyset$. Finally, since $S^- \neq \emptyset$, it follows by property (1) above that $c^- \notin q(D')$. Otherwise, we would have that $b \in q(D')$, for some $b \in S^-$, and thus $b \in q(D)$, contradicting the fact that $q(D) \cap S^- = \emptyset$. In addition, $c^- \notin q_i(D)$, for each $i \in \{1, \ldots, \ell - 1\}$, by definition.

Assume, on the other hand, that Π separates (D', λ), where Π is a statistic with ℓ features from $\mathcal{L}$. For each $i \in \{1, \ldots, \ell - 1\}$ it is the case that $\lambda(c_i) \neq \lambda(c^-)$, and, therefore,

$$\Lambda_{\bar{w}}(\Pi^{D'}(c_i)) \;\neq\; \Lambda_{\bar{w}}(\Pi^{D'}(c^-)).$$

Hence, for each $i \in \{1, \ldots, \ell - 1\}$ there is at least some $q(x)$ in Π such that $q(D') \cap \{c^-, c_i\}$ is either $\{c^-\}$ or $\{c_i\}$. But $q(D') \cap \{c^-, c_i\} = \{c^-\}$ is ruled out by property (1), and hence for each $i \in \{1, \ldots, \ell - 1\}$ there exists a $q(x)$ in Π such that $q(D') \cap \{c^-, c_i\} = \{c_i\}$. By property (2) now, it follows that $q(D') = \{c_i\}$.

Let $c_i, c_i' \in \{c_1, ..., c_{\ell-1}\}$ with $c_i \neq c_i'$. Then there are queries $q(x)$ and $q'(x)$ in Π such that $q(D') = \{c_i\}$ and $q'(D') = \{c_i'\}$. It must be the case then that $q \neq q'$. Thus, 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 $e \notin q_j(D)$.

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 $q'(D') = q(D')$. Since $\eta(D') = \mathrm{dom}(D')$, an atom $\eta(x)$ is

equal to the trivial condition $x \in \mathrm{dom}(D')$ and can be removed. Let us assume then that

$$q(x) := \exists \bar{y} \left(\kappa_{i_1}(z_1) \wedge \cdots \wedge \kappa_{i_m}(z_m) \wedge R_1(\bar{x}_1) \wedge \cdots \wedge R_n(\bar{x}_n) \right),$$

where $1 \leq i_1, \ldots, i_m \leq \ell - 1$ and the R_is come from σ. Since S^+ is a nonempty unary relation, there is some $a \in q(D')$; hence $x \neq z_j$, for each $j \in \{1, \ldots, m\}$. Thus, $q(x) := \exists \bar{y}_0 \exists z_1, \ldots, z_m \left(\kappa_{i_1}(z_1) \wedge \cdots \wedge \kappa_{i_m}(z_m) \wedge R_1(\bar{x}_1) \wedge \cdots \wedge R_n(\bar{x}_n) \right)$, where z is taken from $\bar{y}$ and $\bar{y}_0$ is $\bar{y}$ without $z_1, \ldots, z_m$. By definition, for each $j \in \{1, \ldots, m\}$, the element c_{i_j} only appears twice in D', namely in the facts $\kappa_{i_j}(c_{i_j})$ and $\eta(c_{i_j})$. Therefore, z_{i_j} cannot appear in any of the atoms $R_i(\bar{x}_i)$, for $1 \leq i \leq n$. This means that we can safely remove each atom of the form $\kappa_{i_j}(z_{i_j})$ from q, as it is expressing a trivial condition; i.e., that there is an element in the interpretation of κ_{i_j} over D'. The resulting query $q'(x)$ satisfies that $q'(D') = q(D')$, and, therefore, $q'(x)$ is an $\mathcal{L}$-explanation for (D, S^+, S^-). □

In view of Theorem 6.1 and Lemma 6.5, we obtain the following:

Theorem 6.6. *It is the case that:*

- $\mathbf{CQ}$-Sep[$*$] *is* coNEXPTIME*-complete.*
- $\mathbf{GHW}(k)$-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* $\mathbf{CQ}$ *or* $\mathbf{GHW}(k)$.

The lower bounds for $\mathbf{CQ}$-Sep[ℓ] and $\mathbf{GHW}(k)$-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 6.5 enlarges the schema of the input database for QBE with only ℓ extra unary symbols (for fixed $\ell \geq 1$).

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 6.7. *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* $\mathbf{CQ}$*-separable,*
(3) for every statistics $\Pi = (q_1, \ldots, q_\ell)$ *that* $\mathbf{CQ}$*-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 $\mathbf{GHW}(k)$*, but then, at least one* q_i *has* $\Omega(2^{2^n})$ *atoms.*

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).

6.3 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 $\mathbf{CQ}[m]$—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 $\mathbf{CQ}[m]$ 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 $\mathbf{CQ}[m]$-Sep[$*$] and $\mathbf{CQ}[m]$-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 $\mathbf{CQ}[m]$-Sep[$*$] is FPT, with the parameter being the size of the schema. The proof of this fact is constructive in the sense that it yields a pair $(\Pi, \Lambda_{\bar{w}})$ that $\mathbf{CQ}[m]$-separates the input training database (D, λ) where Π has at most ℓ features. Therefore, the classification problem $\mathbf{CQ}[m]$-Cls[$*$] is also FPT.

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

Notice the difference with Corollary 4.2, which establishes that $\mathbf{CQ}[m]$-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* $\mathbf{CQ}[m]$-Sep[$*$] *is* NP*-complete even for fixed arity schemas.*

Therefore, if for any $m \geq 1$ the problem $\mathbf{CQ}[m]$-Sep[$*$] is FPT with the parameter being the arity of the schema, then P = NP. The reason why $\mathbf{CQ}[m]$-Sep[$*$] is NP-hard is because it involves *choosing* a set of at most ℓ feature CQs in $\mathbf{CQ}[m]$, for a given $\ell \geq 1$, that separates the input (D, λ). An easy reduction from Vertex Cover shows this problem to be NP-hard even for fixed arity schemas. Notice that this establishes an interesting difference with the problem $\mathbf{CQ}[m]$-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* $\mathbf{CQ}[m]$-Sep$[\ell]$ *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 6.5 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 $\mathbf{CQ}$ and $\mathbf{GHW}(k)$, for $k \geq 1$, for which the complexity of QBE is well understood, the complexity of QBE for $\mathbf{CQ}[m]$, 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 6.5, which is a surprisingly negative result. In fact, the problem is NP-complete even for the class $\mathbf{CQ}[1]$ of single-atom CQs.

Proposition 6.11. $\mathbf{CQ}[m]$-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* $S^- = \mathsf{dom}(D) \setminus S^+$.

The lower bound in Theorem 6.10 follows directly then from Lemma 6.5 and Proposition 6.11.

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, $\mathbf{CQ}[m,p]$-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 $\mathbf{CQ}[m,p]$-Sep$[\ell]$ *and* $\mathbf{CQ}[m,p]$-Cls$[\ell]$ *are in* PTIME*, for each fixed* $\ell \geq 1$.
(2) The problem $\mathbf{CQ}[m,p]$-Sep$[*]$ *is* NP*-complete even for fixed arity schemas.*

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 [8, 20, 28], and corresponds to one of the studied notions of separation errors [5, 35]. 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), 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* $\epsilon \in [0, 1/2)$. *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* $\ell \geq 1$.

Now, as all lower bounds for $\mathcal{L}$-Sep, $\mathcal{L}$-Sep$[*]$ and $\mathcal{L}$-Sep$[\ell]$ 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

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

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.

Approximate $\mathbf{CQ}[m]$*-separability.* We first study the approximate separability problem $\mathbf{CQ}[m]$-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 $\mathbf{CQ}[m]$-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 $\mathbf{CQ}[m]$-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.*

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 [19, 21], while $\mathbf{CQ}[m]$-ApxSep calls for *approximate* linear separability, which is NP-complete [17]. 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 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.*

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.

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

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 $\lambda' : \eta(D) \to \{1, -1\}$ 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. □

Algorithm 2 Approx-separability algorithm **GHW**(k)-ApxSep.

Require: A training database (D, λ)

```
([e_1], ..., [e_m]) := equivalence classes with respect to
    →_k over η(D)
for each e ∈ η(D) do
    if Σ_{e'∈[e]} λ(e') ≥ 0 then
        λ'(e) = 1
    else
        λ'(e) = −1
    end if
end for
return λ' : η(D) → {1, −1}
```

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. We start by observing 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.*

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.*

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 statement 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.*

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 3.2 that $\exists$**FO**$^+$-Sep is coNP-complete.

As we have seen, there is an important difference between **FO**-separability and **CQ**-separability: While the former collapses to single-feature **FO**-separability from Proposition 8.3, 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 $\mathbf{GHW}(k)$, 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?

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**, $\mathbf{GHW}(k)$ 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.*

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 $\exists \bar{x}_1 \forall \bar{x}_2 \ldots Q x_n \psi$, where ψ is quantifier-free and $Q = \exists$ if n is odd and $Q = \forall$ 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.*

In contrast, neither **CQ** nor $\mathbf{GHW}(k)$, 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.

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.*

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**, $\mathbf{GHW}(k)$ and Σ_k^+, for any $k \geq 1$, have the unbounded-dimension property.*

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 immediate open problem is the complexity of separability for a bounded number of CQ atoms, that is, $\mathbf{CQ}[m]$-Sep for any fixed $m \geq 1$, when the schema is given as part of the input with no restrictions. 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. [14, 15].

Acknowledgements. We are grateful to M. Romero for useful comments. Barceló is funded by Millennium Institute for Foundational Research on Data and FONDECYT Grant 1170109. Baumgartner is funded by the Austrian Science Fund (FWF) under the project J 3909-N31. Dalmau is funded by MICCIN grant TIN2016-76573-C2-1P and Maria de Maeztu Units of Excellence Programme MDM-2015-0502. The work of Benny Kimelfeld was supported in part by the Israel Science Foundation (ISF), Grant 1295/15.

REFERENCES

[1] Farrukh Ahmed, Michele Samorani, Colin Bellinger, and Osmar R. Zaïane. 2016. Advantage of integration in big data: Feature generation in multi-relational databases for imbalanced learning. In *BigData*. IEEE, 532–539.
[2] Ethem Alpaydin. 2009. *Introduction to machine learning*. MIT press.
[3] Timos Antonopoulos, Frank Neven, and Frédéric Servais. 2013. Definability problems for graph query languages. In *ICDT 2013*. 141–152.
[4] Marcelo Arenas and Gonzalo I. Diaz. 2016. The Exact Complexity of the First-Order Logic Definability Problem. *ACM TODS* 41, 2 (2016), 13:1–13:14.

[5] Boris Aronov, Delia Garijo, Yurai Núñez-Rodríguez, David Rappaport, Carlos Seara, and Jorge Urrutia. 2012. Minimizing the error of linear separators on linearly inseparable data. *Discrete Applied Math.* 160, 10-11 (2012), 1441–1452.

[6] Pablo Barceló and Miguel Romero. 2017. The Complexity of Reverse Engineering Problems for Conjunctive Queries. In *ICDT 2017*. 7:1–7:17.

[7] Angela Bonifati, Wim Martens, and Thomas Timm. 2017. An Analytical Study of Large SPARQL Query Logs. *PVLDB* 11, 2 (2017), 149–161.

[8] Olivier Chapelle, Patrick Haffner, and Vladimir Vapnik. 1999. Support vector machines for histogram-based image classification. *IEEE Trans. Neural Networks* 10, 5 (1999), 1055–1064.

[9] Hubie Chen and Víctor Dalmau. 2005. Beyond Hypertree Width: Decomposition Methods Without Decompositions. In *CP 2005*. 167–181.

[10] Jörg Flum and Martin Grohe. 2006. *Parameterized Complexity Theory*. Springer.

[11] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2008. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* 9, 3 (2008), 432–441.

[12] Georg Gottlob, Gianluigi Greco, Nicola Leone, and Francesco Scarcello. 2016. Hypertree Decompositions: Questions and Answers. In *PODS 2016*. 57–74.

[13] Georg Gottlob, Nicola Leone, and Francesco Scarcello. 2002. Hypertree Decompositions and Tractable Queries. *J. Comput. Syst. Sci.* 64, 3 (2002), 579–627.

[14] Martin Grohe, Christof Löding, and Martin Ritzert. 2017. Learning MSO-definable hypotheses on strings. In *ALT (Proceedings of Machine Learning Research)*, Vol. 76. PMLR, 434–451.

[15] Martin Grohe and Martin Ritzert. 2017. Learning first-order definable concepts over structures of small degree. In *LICS*. IEEE Computer Society, 1–12.

[16] Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lotfi A. Zadeh. 2006. *Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[17] Klaus-Uwe Höffgen, Hans Ulrich Simon, and Kevin S. Van Horn. 1995. Robust Trainability of Single Neurons. *J. Comput. Syst. Sci.* 50, 1 (1995), 114–125.

[18] Sean Kandel, Andreas Paepcke, Joseph M. Hellerstein, and Jeffrey Heer. 2012. Enterprise Data Analysis and Visualization: An Interview Study. *IEEE Trans. Vis. Comput. Graph.* 18, 12 (2012), 2917–2926.

[19] Narendra Karmarkar. 1984. A new polynomial-time algorithm for linear programming. *Combinatorica* 4, 4 (1984), 373–396.

[20] James M. Keller, Michael R. Gray, and James A. Givens. 1985. A fuzzy K-nearest neighbor algorithm. *IEEE Trans. Systems, Man, and Cybernetics* 15, 4 (1985), 580–585.

[21] Leonid Khachiyan. 1979. A Polynomial Algorithm in Linear Programming. *Soviet Mathematics Dodlaky* 20 (1979), 191–194.

[22] Benny Kimelfeld and Christopher Ré. 2017. A Relational Framework for Classifier Engineering. In *PODS 2017*. 5–20.

[23] Benny Kimelfeld and Christopher Ré. 2017. A Relational Framework for Classifier Engineering. In *PODS*. ACM, 5–20.

[24] Arno J. Knobbe, Marc de Haas, and Arno Siebes. 2001. Propositionalisation and Aggregates. In *Principles of Data Mining and Knowledge Discovery, 5th European Conference, PKDD 2001, Freiburg, Germany, September 3-5, 2001, Proceedings*. 277–288.

[25] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2012. *Foundations of machine learning*. MIT press.

[26] C. A. Murthy. 2017. Bridging Feature Selection and Extraction: Compound Feature Generation. *IEEE Trans. Knowl. Data Eng.* 29, 4 (2017), 757–770.

[27] Claudia Perlich and Foster J. Provost. 2006. Distribution-based aggregation for relational learning with identifier attributes. *Machine Learning* 62, 1-2 (2006), 65–105. https://doi.org/10.1007/s10994-006-6064-1

[28] Massimiliano Pontil and Alessandro Verri. 1998. Support Vector Machines for 3D Object Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 20, 6 (1998), 637–646. https://doi.org/10.1109/34.683777

[29] Michele Samorani, Manuel Laguna, Robert Kirk DeLisle, and Daniel C. Weaver. 2011. A Randomized Exhaustive Propositionalization Approach for Molecule Classification. *INFORMS Journal on Computing* 23, 3 (2011), 331–345.

[30] Bernhard Schölkopf and Alexander Johannes Smola. 2002. *Learning with Kernels: support vector machines, regularization, optimization, and beyond*. MIT Press.

[31] Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA.

[32] Balder ten Cate and Víctor Dalmau. 2015. The Product Homomorphism Problem and Applications. In *ICDT 2015*. 161–176.

[33] Ross Willard. 2010. Testing Expressibility Is Hard. In *CP 2010*. 9–23.

[34] Ce Zhang, Arun Kumar, and Christopher Ré. 2014. Materialization optimizations for feature selection workloads.. In *SIGMOD Conference*. 265–276.

[35] Wojciech Ziarko. 1993. Variable precision rough set model. *Journal of computer and system sciences* 46, 1 (1993), 39–59.