Uniform generation in spatial constraint databases and applications

David Gross-Amblard a,*, Michel de Rougemont b

a Équipe Vertigo, Laboratoire CEDRIC, Spécialité Informatique – CC 432, Conservatoire National des Arts et Métiers – Paris, 292, rue St. Martin, 75141 Paris cedex 03, France
b Université Paris II and LRI, Université Paris XI, Bât. 490, 91405 Orsay cedex, France

Received 3 March 2002; received in revised form 31 March 2005
Available online 11 November 2005

Abstract

We study the efficient approximation of queries in linear constraint databases using sampling techniques. We define the notion of an almost uniform generator for a generalized relation and extend the classical generator of Dyer, Frieze and Kannan for convex sets to the union and the projection of relations. For the intersection and the difference, we give sufficient conditions for the existence of such generators. We show how such generators give relative estimations of the volume and approximations of generalized relations as the composition of convex hulls obtained from the samples.

Keywords: Constraint databases; Approximation; Estimator; Generator; Sampling; Volume; Dimension

1. Introduction

The constraint database model, introduced by Kanellakis, Kuper and Revesz [20], offers a uniform way to handle spatial information. This model allows the manipulation of arbitrary high-dimensional geometric sets in a unified framework, using extensions of natural query languages (namely FO + LIN and FO + POLY). When the dimension of geometric
sets is considered fixed, $\text{FO} + \text{LIN}$ and $\text{FO} + \text{POLY}$ queries can be evaluated in polynomial time [20]. But, as noticed in [13] and many other papers, the complexity of constraint query languages behaves badly with the dimension of the geometric sets (basically exponentially in the dimension). In this paper we present a general technique to approximate queries, based on the uniform generation, i.e. on the generation of random points in a definable set with a uniform distribution. We show the relationships between uniform generation, approximate computation of the volume and approximation of the set defined by a first-order formula. We use randomized algorithms, i.e. procedures that succeed with a high probability, and many papers show that only randomized algorithms can produce such approximations (see [9]).

Uniform random sampling has many applications in databases: statistical analysis, decision support, and estimation of aggregate queries where an approximate result is sufficient [18,26]. These methods are of primary interest for Geographical Information Systems (GIS), because many applications are of a statistical nature [8,11,27]. Since constraint databases are well-suited for GIS applications, it is natural to consider sampling operations in this setting.

Algorithms that generate random points with an almost uniform distribution in a given set are called uniform generators and, in case of convex sets, can be related to the computation of approximate volumes in polynomial time in the dimension with a relative error. Relative approximation necessarily requires randomized algorithms.

We use two basic tools: the almost uniform polynomial time generator of Dyer, Frieze and Kannan [9] for a convex set, and the approximation of a polytope by convex hull built from uniform samples [1]. We show how to obtain almost uniform generators, approximate volumes and approximate sets for generalized relations. Our main results are:

1. An almost uniform generator and volume estimator for generalized relations in DNF form and for the projection of a convex relation. For generalized relations not in DNF, we give a sufficient condition for the existence of a uniform generator and for the relative approximation of the volume.
2. A set reconstruction method for positive existential queries. This method allows for approximating the result set of a query, i.e. approximating the shape of the resulting geometrical set, not only its volume.

Related work. The use of sampling in classical databases has been widely studied for the approximation of aggregate COUNT queries [18,25]. From the practical point of view, Olken and Rotem [27] study the uniform generation from a collection of spatial objects stored in an R-tree. They point out that their sampling algorithm scales up to any dimension, but they consider sampling from one spatial object as a black box. The volume approximation (with an additive error) of sets defined by linear or polynomial constraints has been studied by Koiran, Karpinski and Macintyre [23,24] who considered logical formulas that derandomize a Monte-Carlo integration method. The problem of designing good query languages with a volume operator and an approximate volume operator (for an additive error) is studied by Benedikt and Libkin [6].

On classical discrete domains, uniform generation of points in an NP relation has been studied by Jerrum, Valiant and Vazirani [19]: they prove that for self-reducible NP rela-
tions the problem of approximate counting is equivalent to the problem of almost uniform generation. It is then natural to define the notion of a uniform generator for a generalized relation and relate this notion to the approximation of the volume. In the continuous setting, the volume of a convex polytope is \#P-hard to compute in the dimension. If we consider relative approximations, Elekes, Bárány and Füredi [2,10] show that any approximation algorithm must be randomized. The first fully polynomial approximation scheme for the volume of a convex body was given by Dyer, Frieze and Kannan [9]. This procedure is non trivial and cannot be achieved with the uniform sampling in the unit cube: for example, an exponential number of trials are necessary to obtain a single sample from a \(d\)-dimensional sphere (the ratio of the volume of a square and a \(d\)-dimensional sphere is \(\Omega(1/d^d)\)).

**Organization.** In the next section we recall the basic definitions of constraint databases and define the notion of an \((\gamma, \varepsilon, \delta)\)-uniform generator. Section 3 considers approximation when the dimension is assumed fixed. In Section 4 we study the relationship between uniform generator, volume approximation and shape approximation when dimension is not fixed. Section 5 concludes by a discussion on the generalization to polynomial constraints.

### 2. Notations and definitions

**Constraint databases.** We are using the standard notations of linear constraint databases [3–6,14–17,29,33]. Let \(U\) be an infinite set. We call \(\mathcal{M} = \langle U, \Omega \rangle\) an infinite structure with domain \(U\). The set \(\Omega\) is the set of interpreted functions, predicates and constants. We restrict our attention to linear constraints over the reals, i.e. constraints associated to the structure \(R_{\text{lin}} = \langle \mathbb{R}, +, -, <, 0, 1 \rangle\).

A \(d\)-ary generalized tuple is a conjunction of atomic formulas in the language of \(\mathcal{M}\). A \(d\)-ary finitely representable relation is a set \(S \subseteq U^d\) such that there exists a first-order formula \(\phi\) over the language of \(\mathcal{M}\) with

\[
\forall \bar{a} \in U, \quad \mathcal{M} \models \phi(\bar{a}) \quad \text{if and only if} \quad \bar{a} \in S.
\]

The set \(S\) is also called a generalized relation. Since the structure \(R_{\text{lin}}\) admits elimination of quantifiers, the formula \(\phi\) is equivalent to a quantifier-free formula. This formula is equivalent to a formula in disjunctive normal form, thus each generalized relation is a finite union of generalized tuples. The size of a relation \(S\) is the number of symbols of the formula defining \(S\).

A relational database schema is a set of relation names \(\{R_1, \ldots, R_l\}\). A finitely representable instance is a collection of generalized relations \(\{S^1, \ldots, S^l\}\), each associated with its corresponding name in the schema. Our query language will be the first-order logic over the structure \(R_{\text{lin}}\) and the database schema, denoted by FO + LIN. It consists of the atomic formulas over the schema and \(\{+, -, <, 0, 1\}\), and the natural composition of boolean connectives and quantifiers.

**Geometry.** A relation \(S\) represented by one generalized tuple over the language of \(R_{\text{lin}}\) is a finite intersection of open halfspaces. This means that \(S\) is convex. If we are given two positive rational numbers \(r_{\text{inf}} < r_{\text{sup}}\) such that \(S\) contains a ball of radius \(r_{\text{inf}}\) and is totally...
contained in a ball of radius $r_{\text{sup}}$, we say that $S$ is a well-bounded convex relation [12]. Since a generalized relation $S$ is represented by a finite union of generalized tuples, this is also a finite union of convex. We call $S$ a well-bounded relation if all these convex sets are well-bounded.

In the sequel, $\mu_S$ will be the $d$-dimensional volume of the relation $S$. This value is well-defined since all bounded finitely representable relations in $\text{FO} + \text{LIN}$ are measurable. Since we focus on complexity issues, we consider sequences of instances: we will denote by $R$ the sequence of relations $(R_d)_{d \in \mathbb{N}}$, where, for each $d$, $R_d$ has dimension $d$. The union (respectively intersection) of a finite set of sequences $S_1, \ldots, S_k$ is the sequence $T$ defined for each $d \in \mathbb{N}$ by $T_d = S_1^d \cup \cdots \cup S_k^d$ (respectively $T_d = S_1^d \cap \cdots \cap S_k^d$). The minimum (relatively to the volume) of a finite set of sequences, denoted by $\min(S_1, \ldots, S_k)$, is the sequence $T$ such that for each $d \in \mathbb{N}$, $T_d = S_{j_0}^d$, where $j_0$ the smallest element of $\{1, \ldots, k\}$ such that $\mu_{S_{j_0}^d} = \min_i(\mu_{S_i^d})$, $i \in \{1, \ldots, k\}$.

**Definition 2.1.** Two sequences of relations $R$ and $S$ are polynomially related (poly-related for short) if there exists a constant $k \in \mathbb{N}$ such that, for every $d \in \mathbb{N}$:

$$\max\left\{ \frac{\mu_{R_d}}{\mu_{S_d}}, \frac{\mu_{S_d}}{\mu_{R_d}} \right\} \leq d^k.$$  

We will sometimes use “relation” instead of sequence of relations when the exact meaning is clear from the context.

**Generators and estimators.** We shall consider the problem of sampling points uniformly from a relation $S$. Since the domain of $\mathcal{R}_{\text{lin}}$ is infinite, we consider a discretization of $S$. We call a grid of step $p$ the set $G_p$ of points in $\mathbb{R}^d$ whose coordinates are multiple of $p$. For a relation $S \subseteq \mathbb{R}^d$, the graph induced by $G_p$ on $S$ is the graph with vertices $V = G_p \cap S$. Edges of this graph are pairs $(\bar{a}, \bar{b}) \in V \times V$ such that $\bar{a}$ and $\bar{b}$ are at distance $p$. We use a small enough grid such that the number of vertices induced on $S$ is closely related to the volume of $S$. All approximations are relative, as in the classical literature on fully polynomial-time randomized approximation schemes (FPRAS, see, e.g., [28]).

A randomized algorithm has the ability to pick a random bit $b$ at each step, and to adapt its computation according to the value of $b$. Hence a given computation is a path in the tree of all possible random choices along with its corresponding probability. The set of all paths and probabilities form a probability space $\Omega$, hence the probability that a given computation occurs is well-defined. We note for short $j \in R$ $S$ the random choice of an element in $S$ with a prescribed probability. If $\alpha$, $\beta$ and $\varepsilon$ are positive reals with $0 < \varepsilon < 1$, we say that $\alpha$ approximates $\beta$ with ratio $1 + \varepsilon$ if $(1 + \varepsilon)^{-1} \beta \leq \alpha \leq (1 + \varepsilon) \beta$. A randomized algorithm is said to be an $(\varepsilon, \delta)$-volume estimator for a relation $S$ if, given $0 < \varepsilon, \delta < 1$ as parameters, it computes a value $\hat{\mu}_S$ such that:

$$\mathbb{P}_\Omega[\hat{\mu}_S \text{ approximates } \mu_S \text{ with ratio } 1 + \varepsilon] \geq 1 - \delta.$$  

Its running time must be polynomial in the description size of $S$, $1/\varepsilon$ and $\ln(1/\delta)$. The $\ln(1/\delta)$ bound on complexity is a classical assumption.
Let $0 < \gamma < 1$ and $p$ such that the value $|V|p^d$ approximates $\mu_S$ with ratio $1 + \gamma$. In order to avoid too small grids, we suppose that $p$ is polynomial in $\gamma$ and $1/d$. If these two conditions are met, we call $G_p$, a $\gamma$-grid for $S$. Except for the discretization parameters, we use standard notions from [19]. A generator for $S$ is a randomized algorithm which generates a point uniformly in the graph induced on $S$ by a $\gamma$-grid for $S$. For several reasons, it is convenient to consider such algorithms that may succeed with high probability and may fail, i.e. stop and abandon, with a small probability $\delta$. The distribution of the output is also allowed to deviate from the uniform distribution by a little amount (prescribed by $\varepsilon$).

**Definition 2.2.** An $(\gamma, \varepsilon, \delta)$-generator is a randomized algorithm which, given a relation $S$ and real numbers $0 < \varepsilon, \delta, \gamma < 1$, computes a $\gamma$-grid $G_p$ and outputs points from $V = G_p \cap S$ such that:

1. when a computation is successful, for all vertices $v$ of $V$,
   \[
   \frac{1}{1 + \varepsilon} \frac{1}{|V|} \leq P_{G_p}[\text{output is } v] \leq (1 + \varepsilon) \frac{1}{|V|};
   \]
2. the algorithm fails with probability smaller than $\delta$;
3. the algorithm runs in time polynomial in the description size of $S$, $d$, $1/\varepsilon$, $1/\gamma$ and $\ln(1/\delta)$.

Notice that $\gamma$ controls the size of the grid $G_p$ in dimension $d$, i.e. $|V|p^d$ must be an approximation of the volume of $S$ with ratio $(1 + \gamma)$. The parameter $\varepsilon$ controls the quality of the distribution. A relation that possesses both a generator and a volume estimator is said to be observable.

**Uniform sampling from a convex set and volume estimation.** Given a well-bounded convex body $K$ by a membership oracle (i.e. an algorithm that tells if a point belongs to the set) the Dyer–Frieze–Kannan technique [9] first computes a non-singular affine transformation $Q$ that makes the body $K$ “well-rounded.” The transformed body $Q(K)$ contains the unit ball $B$ and is totally contained in a ball of radius $\sqrt{d}(d + 1)$, depending only on the dimension of the space (this is possible only if the convex body is well-bounded). In a second step, they consider a random walk on the graph $G$ induced by a $\gamma$-grid on the set $Q(K)$, starting at the origin vertex. This random walk is rapidly mixing: after a polynomial number of steps, the random walk is almost uniformly distributed on all the vertices of $G$. Finally, they consider a sequence of convex sets $B = K_0 \subseteq K_1 \subseteq \cdots \subseteq K_q = Q(K)$ such that $\mu_{K_{i+1}}/\mu_{K_i}$ is bounded by a constant (taking homothetic $K_i$’s is sufficient). The uniform generator for each convex set is used to estimate each ratio (by a classical Chernoff estimator). The product of each ratio gives the approximate volume of $Q(K)$, thus of $K$.

A membership oracle for a finitely representable relation $S$ is easy to compute in linear time in its description size: it is sufficient to check each constraint for the given assignment of variables. This leads to the following fundamental result:

**Theorem** (Dyer, Frieze, Kannan). If for each $d \in \mathbb{N}$, $S_d$ is convex and well-bounded, then $S$ is observable.
For \( \varepsilon, \gamma, \delta \) and \( S \) as input, the grid size \( p \) will be \( O(\gamma/d^{3/2}) \). The mixing time of the random walk is \( O((d^{19}/\varepsilon \gamma) \ln(1/\delta)) \), so a random point can be generated in polynomial time. The volume estimator, using the generator, has complexity \( O((d^{19}/\varepsilon) \ln(1/\delta)) \). Several improvements reduce this complexity to \( O^*(d^5) \) [21].

3. Fixed dimension

In classic constraint query languages, the dimension is assumed fixed. In this section we show that volume computation and random sampling can be easily obtained under this hypothesis.

**Theorem 3.1.** Any generalized relation is observable when dimension is assumed fixed.

The proof of this theorem relies on the two following lemmas.

**Lemma 3.1** (direct application of [7]). Computing the exact volume of a generalized relation \( S \) is polynomial time if dimension is assumed fixed.

**Proof.** We first perform quantifier elimination on the formula describing \( S \). This step is polynomial time for fixed dimension. Given the resulting quantifier-free constraint definition of the generalized relation, we apply the plane-sweep algorithm from Bieri and Nef [7]. It complexity is polynomial in the description size of the generalized relation. However, this complexity is exponential in the dimension, so the fixed dimension hypothesis is required.

Of course, the exact volume computation algorithm can be seen as a volume estimator.

In order to sample points from a generalized relation, we simply cut its bounding box into cubes, enumerate these cubes and choose one randomly and uniformly.

**Lemma 3.2.** Sampling from any generalized relation \( S \) is polynomial time if dimension is assumed fixed.

**Proof.** Let \( R \) be the size of the bounding box enclosing \( S \). For the desired precision \( \gamma \), we consider a regular decomposition of the bounding box into cubes of size \( \gamma \). The overall number of these cubes is \( (R/\gamma)^d \), i.e. a polynomial number when dimension is considered fixed. In order to sample cubes uniformly, we first enumerate cubes that belongs to \( S \): this operation needs \( (R/\gamma)^d \) membership tests in \( S \), and each membership test is linear in the description size of \( S \). If \( n \) is the number of cubes in \( S \), for each needed sample, we choose a cube in \( S \) with probability \( 1/n \).

In the sequel, we will obtain similar results for a fragment of generalized relations, without the fixed dimension hypothesis.
4. Uniform generation and volume approximation: general case

The relationship between almost uniform generation and approximate counting is well-studied [19]. For the continuous setting, we will generalize the method of [9] to combinations of observable relations.

4.1. Boolean operations

We will define different algorithms that sample from a relation and approximate its volume. For an observable relation \( S \), we consider \( G_S = (V_S, E_S) \) the graph induced by a \( \gamma \)-grid \( G_S \) for \( S \). We denote by \( \text{ApproxGen}(S, \gamma, \epsilon, \delta) \) the generator for \( S \) and by \( \text{ApproxVol}(S, \epsilon, \delta) \) the volume estimator for \( S \). We now study how to compose these operators with the union, the intersection and the difference of observable relations.

4.1.1. Union

To sample the union of observable relations, we use an argument similar to the one given in [22] for the approximation of \#DNF, but in the geometrical setting. Remark that a direct random walk on the union is not likely to succeed. Indeed, the union may not be connected and the random walk would stay infinitely in one component. Moreover, even if the union is connected, there is no guarantee that the random walk will uniformly cover the whole set, without being attracted by a specific location. Consider for example two large convex sets \( S \) and \( S' \) linked by a thin convex tube \( T \): starting from \( S \), the probability to walk randomly through the bridge \( T \) and to reach \( S' \) is likely to be small (see, e.g., [30]).

**Theorem 4.1.** Given two observable relations \( S^1 \) and \( S^2 \), there exists a \((\gamma, \epsilon, \delta)\)-generator for the relation \( T \) defined by \( S^1 \cup S^2 \).

We will first choose one of the relations with probability proportional to its volume, and choose a random point in this relation with a uniform distribution. In order to deal with overlapping relations and ensure that each point is chosen only once, we output the point only if it is chosen from, say, \( S^1 \). Below is an inductive definition of \( \text{ApproxGen} \) for \( T \) given algorithms \( \text{ApproxGen} \) and \( \text{ApproxVol} \) for base relations \( S^1 \) and \( S^2 \).

**Algorithm 1** \( \text{ApproxGen}(T, \gamma, \epsilon, \delta) \).

Repeat \( k \) times:
1. For each \( i \in \{1, 2\} \), computes \( \hat{\mu}_i = \text{ApproxVol}(S^i, \epsilon/3, \delta') \).
2. Let \( \hat{\mu} = \hat{\mu}_1 + \hat{\mu}_2 \).
3. Choose a \( j \in R \{1, 2\} \) with probability \( \hat{\mu}_j/\hat{\mu} \).
4. Let \( x = \text{ApproxGen}(S^j, \gamma, \epsilon/3, \delta') \).
5. If \( x \in S^1 \) or \( x \in (S^2 - S^1) \), return \( x \), else fail.

We now prove Theorem 4.1. For a point \( x \), we denote by \( j(x) \) the smallest index \( i \) such that \( S^i \) contains \( x \).
Proof. We must verify the construction of the grid and the three conditions of Definition 2.2. One of the difficulties when composing generators is the grid size. Remark that if a relation is exponentially smaller that the others, one can consider it empty without modifying the approximation ratio. We can then suppose without loss of generality that each of the \( S^i \)'s have poly-related volumes.

To construct the greatest common grid \( G \) of all the \( G_{S^i} \), one can extend the uniform generator of each \( S^i \) to this grid. Since the \( S^i \)'s are poly-related, the resulting grid size is not too small. Let \( V_T = V_{S^1} \cup V_{S^2} \) be the vertices of the graph induced by \( G \) on \( T \).

Let \( \mu_i = \mu_{S^i} \) and \( \mu = \mu_1 + \mu_2 \). The algorithm \( \text{ApproxGen} \) may fail because of probabilistic steps and the test in (5). Let us first suppose that each step succeeds and that a point \( x \) is returned. The only way to return this particular \( x \) is that step (3) produces a \( j \) equal to \( j(x) \). The probability that this particular relation is chosen is \( \hat{\mu}_j(x)/\hat{\mu} \). The point \( x \) is drawn from \( S^{j(x)} \) by \( \text{ApproxGen} \) with probability \( 1/\hat{\mu}_j(x) \). So the output probability of \( x \) is \( (\hat{\mu}_j(x)/\hat{\mu})(1/\hat{\mu}_j(x)) \), which approximates \( 1/\mu \) with ratio \( (1 + \varepsilon/3)^3 \leq (1 + \varepsilon) \). This gives the desired almost uniform distribution.

The algorithm is unreliable at steps (1), (4) and (5). For steps (1) and (4) we choose a failure probability \( \delta' = 1/8 \). For step (5), let \( x_0 \) be a point in \( V_T \). Let \( N \) be the number of different \( V_{S^i} \) containing \( x_0 \). Hence, \( N \leq 2 \). There are at most \( N \) distinct pairs \( (j, x_0) \) that can be produced by the algorithm at steps (3) and (4), each with the same probability. But only one leads to an accepting state (i.e. pair \( (j(x_0), x_0) \)). So for any \( x_0 \), the success probability of step (5) is \( 1/N \), which is bigger than \( 1/2 \).

The total failure probability is smaller than \( 2\delta' + 1 - 1/2 = 1 - 1/4 \). Considering \( k \) successive executions of this algorithm, the overall failure probability is smaller than \( (1 - 1/4)^k \). Since \( (1 - 1/4)^k \leq e^{-k/4} \), repeating the whole algorithm \( k = 4 \ln(1/\delta) \) times gives a general failure probability smaller than parameter \( \delta \).

The total complexity of this algorithm is bounded by \( 4 \ln(1/\delta) \) times the complexity of steps (1) and (4). They are both polynomial in \( d \), and \( 1/\varepsilon \) by hypothesis. The whole algorithm is then polynomial in \( d \), \( 1/\varepsilon \), \( 1/\gamma \) and \( \ln(1/\delta) \). \( \square \)

Theorem 4.2. Given two well-bounded observable relations \( S^1 \) and \( S^2 \), there exists an \( (\varepsilon, \delta) \)-volume estimator for the relation \( T \) defined by \( S^1 \cup S^2 \).

Proof. Let \( B_0 \) be the inner ball of \( S^1 \), and \( B_1 \) the ball enclosing all \( S^i \)'s. We apply the same volume estimation method as in the convex case, but now using the generator from the union. Notice that the description size of relations \( S^1 \), \( S^2 \) and \( B_1 \)'s radius are related by a linear function. \( \square \)

Corollary 4.1. Well-bounded observable relations are closed under binary union.

4.1.2. Intersection

Consider \( T \) as the intersection of two observable relations \( S^1 \) and \( S^2 \). In order to sample in the intersection, we will sample from, say, \( S^1 \), and test membership in \( S^2 \). But if this intersection is very small (exponentially smaller than \( S^1 \)), we may sample a long time before discovering a single point in the intersection. In this case we will neither have an efficient uniform generator nor an approximation of the volume.
Proposition 4.1. The relation $T$ defined as $S_1 \cap S_2$ is observable if $T$ and $\min(S_1, S_2)$ are poly-related.

Proof. Compute the approximate volumes of $S_1$ and $S_2$. Choose $j$ such that $S^j$ has the smallest volume, e.g. $j = 1$. Use the generator for $S^1$ and check whether the points are also in $S^2$. If they are, output them, otherwise iterate the process.

We obtain a $(\gamma, \epsilon, \delta)$-uniform generator for $T$: the $\gamma$-grid of $S^j$ is a $\gamma$-grid for $T$ because $T$ and $S^j$ are poly-related. The previous generator is an $(\gamma, \epsilon, \delta)$-generator. After some polynomial time, we almost surely obtain points in $T$ and their convex hull contains a ball of radius $r_{\min}$. The set $T$ is almost uniformly generated and well-bounded: its volume can be approximated by the technique of [9].

4.1.3. Unbounded unions and intersections

In classical constraint databases, the number of applications of the union and intersection operators is supposed fixed. But the previous algorithms remain polynomial time even for unbounded unions and intersections.

Corollary 4.2. Given observable relations $S^1, \ldots, S^m$, the relation $T$ defined by $\bigcup_{i=1}^m S^i$ is observable.

Proof. The generator is obtained as in Theorem 4.1 by choosing $\delta' = 1/m$. A point $x$ sampled from $S^i$ is returned only if $j(x) = i$. Time complexity is increased by a factor $m$. The volume estimation is straightforward.

Corollary 4.3. The relation $T$ defined as $\bigcap_{i=1}^m S^i$ is observable if $T$ and $\min(S^1, \ldots, S^m)$ are poly-related.

Notice that a SAT instance can be encoded in the following geometric way: with each literal $x$ (respectively $\overline{x}$), we associate the constraint $3/4 < x < 1$ (respectively $0 < x < 1/4$). A disjunction is the finite union of such constraints, defining a finite union of convex sets, which are observable. A SAT instance is finally represented as the intersection of such observable sets.

Since relative volume approximation can be used to decide emptiness of a geometric set, an $(\epsilon, \delta)$-volume estimator for the general intersection would yield a polynomial-time algorithm for the SAT problem. Hence, the restriction on the relative size of $T$ is necessary, unless P = NP.

4.1.4. Difference

Consider the difference of two observable sets $S^1$ and $S^2$. It is neither connected nor convex in general but may still be observable.

Proposition 4.2. The relation $T$ defined as the difference of two observable relations $S^1$ and $S^2$ is observable if the size of $T$ and $S^1$ are poly-related.

Proof. Consider the generator for $S^1$ where we only output points which are not in $S^2$. The generator selects the $\gamma$-grid of $S^1$. Because $T$ is relatively large, we almost surely obtain
points in $T$ after some polynomial time and their convex hull contains a ball of radius $r_{\text{min}}$. The set $T$ is almost uniformly generated: we obtain a $(\gamma, \epsilon, \delta)$-uniform generator and an $(\epsilon, \delta)$-volume estimator. □

4.2. Projection

Suppose we have a generator for a convex relation $S \subseteq \mathbb{R}^d$, and let $x = (x_1, \ldots, x_d)$ be a uniformly generated point in $S$. Then the point $y = (x_2, \ldots, x_d)$ belongs to $T$, the projection of $S$ according to the first coordinate. But $y$ is not uniform in $T$, as shown in Fig. 1.

In this example, $d = 2$, and $C_1$ to $C_8$ denote the cylinders induced by a $\gamma$-grid for $S$. If a point $x = (x_1, x_2)$ is drawn uniformly on a grid for $S$, it is more likely to appear in the cylinder $C_5$ than in the smaller $C_1$. Its projection $y = (x_2)$ on the second coordinate is not uniform in $T$. It is then necessary to compensate this effect: we reject $y$ with probability proportional to the volume of the cylinder containing $x$, which can itself be computed by the previous algorithms.

Theorem 4.3. The relation $T$ defined by projection of a convex relation $S$ is observable.

If $S$ has a very elongated form, it can be “well-rounded” (i.e. mapped into a sphere by an affine transform). Then the volume of $T$ and $S$ are related. Given a subset $I$ of coordinates and a point $y$ in $T$, let $H_S(y)$ be the cylinder of points whose projection on coordinates in $I$ is exactly $y$. We consider the following generator for the projection:

Algorithm 2 (ApproxGen$(T, I, \gamma, \epsilon, \delta)$).

1. Repeat $k$ times:
2. Choose $x = \text{ApproxGen}(S, \gamma, \epsilon/3, \epsilon/(4d^3))$.
3. Let $y$ be the projection of $x$ on the coordinates in $I$.
4. Compute $\hat{h} = \text{ApproxVol}(H_S(y), \epsilon/3, \epsilon/(4d^3))$.
5. Return $y$ with probability $1/\hat{h}$, else fail.

Fig. 1. Projection.
Proof. We can consider without loss of generality that $S$ is well-rounded, i.e. that $S$ contains the unit ball and is contained in a ball of radius $d^{3/2}$. One can use the projection of a $\gamma$-grid for $S$ as a $\gamma$-grid for $T$.

Let $\hat{\mu}$ be the volume of $S$ and suppose that each probabilistic step of the algorithm succeeds. A point $y \in T$ is returned if step (2) generates a point $x \in H_S(y)$, and step (5) accepts. The first event occurs with probability $\hat{h}/\hat{\mu}$, where $\hat{h}$ is the volume of $H_S(y)$. Step (5) accepts with probability $1/\hat{h}$, so the overall probability is $1/\hat{\mu}$. This is almost uniform with ratio $(1 + \varepsilon/3)^3 \leq (1 + \varepsilon)$.

The algorithm is unreliable on steps (2) and (4) with probability $\varepsilon/(4d^3)$. Since the grid size $p$ is equal to $\varepsilon/d^{3/2}$ and that $H_S(y)$ contains at most $d^{3/2}d^{3/2}/\varepsilon$ points of the grid, the last step succeeds with probability at least $\varepsilon/d^3$. The overall failure probability is then smaller than $\varepsilon/(4d^3) + \varepsilon/(4d^3) + (1 - \varepsilon/d^3) \leq 1 - \varepsilon/(2d^3)$. Repeating the whole algorithm $k = O((d^3/\varepsilon) \ln(1/\delta))$ times gives the desired success probability $\delta$.

Since the projection of the inner (respectively enclosing) balls of $S$ is an inner (respectively enclosing) ball for $T$, the volume of $T$ can be approximated by the technique of Dyer, Frieze and Kannan. 

4.3. Reconstruction of queries

We consider the problem of reconstructing a definable relation from samples. In the classical approach to constraint spatial databases, one manipulates relations in a symbolic way: starting with an arbitrary first-order formula, one first needs to eliminate quantifiers and this is known to be hard. In order to obtain an asymptotic speed-up for queries over high-dimensional relations, we would like to avoid symbolic computations. If we have an almost uniform generator, we may be able to approximate a query which defines a relation $S$, i.e. write formulas that define a relation $\hat{S}$ such that the volume of the symmetric difference will be small.

Definition 4.1. A $(\varepsilon, \delta)$-estimator for a relation $S$ is a randomized algorithm such that, given $0 < \varepsilon, \delta < 1$ as parameters:

1. The algorithm produces the description of a relation $\hat{S}$ that approximates $S$ with failure probability smaller than $\delta$, i.e.:

$$\mathbb{P}_D[\mu(S \Delta \hat{S}) \geq (1 + \varepsilon)\mu(S)] \leq \delta,$$

where $S \Delta \hat{S} = (S - \hat{S}) \cup (\hat{S} - S)$ denotes the symmetric difference between $S$ and $\hat{S}$.

2. The algorithm uses only point membership queries in $S$.

Notice that the previous generators require only point membership queries. We distinguish the case of convex sets from the general case.
4.3.1. Convex sets

The basic tool is a result from [1]: if we have a uniform generator for a convex polytope $S$, then the convex hull of $N$ uniformly generated points approximates the set $S$ with $r$ vertices, with ratio

$$1 + \frac{rd}{d^{d-2}} \frac{\ln^{d-1} N}{N}.$$

Lemma 4.1. If $N$ is in

$$O\left(\frac{4r^2 d^2}{\varepsilon^4 d^{2d-2}} \ln \frac{1}{\delta}\right),$$

the convex hull of $N$ samples uniformly generated in a convex polytope $S$ with $r$ vertices is an $\varepsilon$-approximation of $S$ with failure probability smaller than $\delta$.

Proof. Let $N$ be the number of random generated points in a convex relation $S$ with volume $\mu_S$ and $r$ vertices. By the result from [1], we know that the expected volume $\mathbb{E}[V]$ of the convex hull of these points is such that

$$\mu_S \left(1 - \frac{rd}{d^{d-2}} \frac{\ln^{d-1} N}{N}\right) \leq \mathbb{E}[V] \leq \mu_S.$$

For big enough $N$, $(\ln^{d-1} N)/N \leq 1/\sqrt{N}$. Taking

$$N = \frac{4r^2 d^2}{\varepsilon^4 d^{2d-2} \varepsilon^2}$$

ensures that $|\mu_S - \mathbb{E}[V]| \leq \varepsilon \mu_S / 2$.

We now repeat this random process $t$ times, and consider $V_1, \ldots, V_t$ the volume of the corresponding convex hulls. The convex hull $C$ of all these points is larger than each of the $V_i$, particularly larger than the mean $S_t = (\sum_{i=1}^t V_i)/t$. Applying the Chernoff bound, we know that:

$$\mathbb{P}_{S_t} \left[ \left| \frac{S_t}{\mu_S} - \frac{\mathbb{E}[V]}{\mu_S} \right| \leq a \right] \geq 1 - 2e^{-2a^2t}.$$

Furthermore, $S_t \leq \mu_C \leq \mu_S$. In order to obtain $|\mu_C - \mathbb{E}[V]| \leq \varepsilon \mu_S / 2$, we take $a = \varepsilon / 2$. Then $|\mu_C - \mu_S| \leq \varepsilon \mu_S$, and the desired success probability is realized with $t = (1/\varepsilon^2) \ln(1/\delta)$. \hfill \Box

Notice however that one has to effectively compute the convex hull of these $N$ random points, which is known to be an exponential process in the dimension (roughly $O(N^{d/2})$) [32]. Consider now the relation $T$ defined by the query $\phi$ on an observable convex relation $S$ in dimension $e + d$:

$$\phi(x_1, \ldots, x_e) \equiv \exists x_{e+1} \exists x_{e+2} \ldots \exists x_{e+d} R(x_1, \ldots, x_{e+d}).$$

The query $\phi$ expresses a projection on an $e$-dimensional subspace of $\mathbb{R}^{d+e}$. Its standard implementation in constraint databases is the Fourier–Motzkin algorithm [31] whose complexity is $O(2^k)$, where $k$ is the number of projected variables. In our example, this is $2^{2d}$. In order to get an asymptotic speed-up, consider the following algorithm:
Algorithm 3.

1. Generate $N$ random points uniformly in the projection of $S$ with the projection generator.
2. Form the convex hull of these points.

**Proposition 4.3.** An $(\epsilon, \delta)$-estimation of the relation $T$ defined by $\phi$ can be obtained in $O(2^{e/2} \cdot \text{poly}(d + e))$ computation steps.

**Proof.** Step 1 is polynomial in $d + e$ (and $\epsilon, \ln(1/\delta)$) if it uses our projection generator. Step 2 takes exponential time, by any classical convex hull computation, but in the resulting dimension $e$. $\square$

4.3.2. General sets

The set defined by a first-order formula may not be convex. We show how to generalize the previous approach: we will only guarantee the approximation of the result when we can compute an approximate volume. Consider a simple example where $R_1$, $R_2$, $R_3$ are given well-bounded convex relations in dimension 2. Let $T$ be the relation defined by the formula $\exists z[(R_1(x, z) \land R_2(z, y)) \lor R_4(x, z)]$.

An approximation of the result could be obtained by taking the convex hull $C_1$ of $(R_1(x, z) \land R_2(z, y))$ using the uniform generator for the intersection and projecting it on $z$ into $C_1'$. Similarly we could compute the convex hull $C_2$ of $R_4(x, z)$ using the Dyer–Frieze–Kannan uniform generator for $R_4$ and project it on $z$ into $C_2'$. The result would be the union of $C_1'$ and $C_2'$ but we would not guarantee the approximation because the simple projection of a uniform generator is not necessarily uniform. We would also compute convex hulls in dimension 3 which is not necessary. We can modify our procedure as follows:

**Algorithm 4 (Guaranteed approximation of $T$).**

1. Generate uniform points in $\exists z(R_1(x, z) \land R_2(z, y))$ (combining the uniform generation for the intersection and the projection), and take their convex hull $D_1$.
2. Generate uniform points in $\exists z R_4(x, z)$ and take their convex hull $D_2$.
3. The result is the union of $D_1$ and $D_2$.

In order to generate points uniformly in $\exists z(R_1(x, z) \land R_2(z, y))$ we still need the condition: $R_1$ and $R_2$ are poly-related. Notice that the approximation of $D_1$ and of $D_2$ is now guaranteed because we select $N$ uniform points in a polytope. The implicit approach for the general reconstruction of an existential positive formula $\Psi$ is the following algorithm:

**Algorithm 5 (Guaranteed approximation of an existential positive formula $\Psi$).**

1. Write the formula as the disjunction of conjunctions and projections: $\bigvee_i \varphi_i$ where each $\varphi_i$ is built from atomic formulas by conjunction and existential quantification.
2. Generate uniform points in the sets defined by each $\varphi_i$ (with the techniques of the previous section) and take their convex hull $D_i$.
3. The result is the union of the $D_i$. 
Theorem 4.4. Let $\Psi$ be an existential positive formula equivalent to $\bigvee_i \varphi_i$ where $\varphi_i$ is obtained by conjunction and projection. For each constraint database, if there is a uniform generator for the set defined by each formula $\varphi_i$, then the set defined by Algorithm 4 is an $(\varepsilon, \delta)$-estimator for the set defined $\Psi$.

Proof. Apply the projection technique to the intersection of convex sets in order to obtain uniform generators for the sets defined by each $\varphi_i$. Applying the previous method for convex sets, we obtain an $(\varepsilon, \delta)$-estimator for the set defined by $\varphi_i$. The union of the estimates is also an $(\varepsilon, \delta)$-estimator for the set defined by $\Psi$. □

In a case of negation (or difference), we may still get a uniform generator but the reconstruction is more difficult, since convexity is lost.

5. Conclusion and extensions

We studied how to approximate queries in constraint databases with the use of random sampling. We showed how to combine the basic uniform generator of Dyer–Frieze–Kannan for convex sets with the classical logical operators. We obtain uniform generators for the union and the projection of convex sets and gave a sufficient condition for the intersection and the difference.

A uniform generator also yields a method to approximate the volume in polynomial time and gives a general method to approximate queries as a combination of convex hulls. The algorithms for the uniform generation and for the volume are randomized and polynomial in the size of the parameters.

As a conclusion, we mention the extension of our results to polynomial constraints. The Dyer–Frieze–Kannan generator supposes only the existence of a membership oracle for a convex set. This oracle can be easily computed for generalized relations given by polynomial constraints if they are convex (the conjunction of polynomial constraints does not necessarily define a convex set).

Our generators and volume estimators for the union, intersection, difference and projection do not rely on the linearity and will generalize to polynomial observable sets.

Reconstruction of convex sets defined by polynomials involves complicated techniques like interpolation. But if one accepts a good approximation by a simple polytope (i.e. a set defined by linear-only inequalities), our previous reconstruction method can be considered.

Let $G_p$ be a $\gamma$-grid for a convex set $S$. The Dyer–Frieze–Kannan estimator approximates the size of a set $V = G_p \cap S$, by generating almost uniform points in $V$. Consider now the polytope hull($V$), defined as the convex hull of points in $V$. Clearly, the estimator does not distinguish between $S$ and hull($V$), and their volume are closely related.

Hence, if hull($V$) has a given number of vertices $r$, one can use the algorithm of Lemma 4.1 to approximate the convex set $S$ by a convex polytope.

Lemma 5.1. If $r = \text{poly}(d, 1/\varepsilon)$, the relation estimator for hull($V$) is a relation estimator for $S$. 


We suspect that for smooth convex bodies defined by polynomial constraints of a fixed degree, $r$ always satisfies the previous conditions.

**Acknowledgments**

We thank David Applegate, Luc Segoufin and Emmanuel Waller for many helpful discussions.

**References**


