Querying with Vague Quantifiers Using Probabilistic Semantics

Christian G. Fermüller, Matthias Hofer, and Magdalena Ortiz

TU Vienna, Austria

Abstract. Many realistic scenarios call for answers to questions involving vague expressions like *almost all*, *about half*, or *at least about a third*. We present a modular extension of classical first-order queries over relational databases, with *binary, proportional, semi-fuzzy quantifiers* modeling such expressions via random sampling. The extended query language has an intuitive semantics, and allows one to pose natural queries whose answers. This is also demonstrated by experiments with an implementation involving the (geographical) MONDIAL data set.

1 Introduction

In verbal communication one frequently uses expressions like *about half*, *at least about a third*, *at most about a quarter*, *almost all*, etc. Obviously the meaning of these expressions is vague, context dependent and potentially also involves intensional aspects, like considerations about specific expectations of speakers and listeners. Nevertheless such quantifier expressions may be understood as mainly referring to the *proportion* of those domain elements satisfying the range predicate that also satisfy the scope predicate. For example, *Almost all capitals have airports* refers to the proportion of capitals in the domain of discourse (range predicate) that satisfy the scope predicate *'has an airport'*, and claims that this proportion is close to 1. The intrinsic vagueness of quantifier expressions is often a virtue, rather than a problem: it allows us to communicate concisely and effectively without having to compute or even to know the precise proportion in question. This observation motivates the design of formal models of proportionality quantifiers that support efficient and intuitive answers to queries that involve corresponding formal counterparts of relevant natural language expressions.

In this paper we propose to augment a standard first order query language over relational and RDF data with quantifiers that allow one to formulate queries like *Which countries are larger than almost all other countries?* or (Boolean queries) like *Do at least about two thirds of all capitals have more than a million inhabitants?* The vagueness of the corresponding quantifier expressions is modeled by a probabilistic, sampling based semantics, connecting our approach to game based semantics for so-called semi-fuzzy proportionality quantifiers [5, 6] as explained below.[1]

[1] Although this is not a main concern in this work, we point out that a sampling based semantics may also be useful to compute quick, approximative answers in face

Related work. Flexible and approximate query answering has received significant attention in the database community for several decades, and the literature on the subject is vast. Our work has some relation to fuzzy query answering, see [14] and references therein. However, our approach is different in at least two respects: (1) we want to keep the user interface free of references to degrees of truth or of set membership, and (2) we base the semantics of semi-fuzzy quantifiers on sampling, rather than on *ad hoc* truth functions, as usual in fuzzy set based approaches. We are not aware of similar proposals that modularly extend existing query languages with proportional quantifiers only, while keeping crisp both the database and underlying query formalism. Sampling plays an important role on approximate query answering (see, e.g., [9, 2]), but in this area, the focus is usually on keeping suitable samples of the data to support the efficient computation of aggregate queries over large data sets, rather than supporting vague quantifiers formalizing natural language expressions. In this light, we think that our specific combination of ideas is original and may be of interest.

The paper is organized as follows. We start by providing basic terminology on (semi-)fuzzy quantifiers in Section 2, and we briefly recall the game and sampling based approach to fix their semantics in Section 3. Section 4 reviews principles of statistical sampling that are used in the central Section 5, which presents our proposal for extending queries with vague proportional quantifiers. In Section 6 we illustrate our approach on a a concrete use case. Section 7 briefly summarizes our work and presents a few topics for related future research.

2 (Semi-)fuzzy proportionality quantifiers

The standard model for vague quantifier expressions in a computational context uses fuzzy logic (see, e.g., [20]). A fuzzy set over a *domain* D is a function $\widetilde{X} : D \to [0, 1]$, assigning to each element of D a *membership degree* in the real unit interval. Fuzzy sets are intended to model vague or imprecise predicates, where any given element does not necessarily either determinately satisfy or not satisfy the predicate, but where $\widetilde{X}(d)$ is understood as the *degree of truth* of the assertion that d satisfies the predicate modeled by $\widetilde{X}$. If $\widetilde{X}(d) \in \{0, 1\}$ for all $d \in D$, one speaks of a *crisp set* or predicate. Identifying 0 with 'definitely false' and 1 with 'definitely true', crisp predicates clearly correspond to classical (bivalent) predicates and crisp sets turns into (membership functions of) ordinary sets.

As already indicated, commonly used (logical) natural language quantifiers involve a range as well as a scope predicate, i.e. they are *binary* in the terminology of, e.g., [4]. This also holds for standard universal and existential quantification:

of massive volumes of data. For example it might be useful to quickly obtain a highly, but not perfectly reliable answer to a query like *Have at least about a quarter of all citizens lived abroad at some time?* without having to visit each relevant entry in a huge database containing such data for all citizens.

one routinely asserts sentences of the form *All A are B* or *Some A are B*.[2] For crisp predicates, we may express the first sentence in classical logic as $\forall x(A(x) \to B(x))$ and the second one as $\exists x(A(x) \wedge B(x))$. But note that such a reduction of binary quantification to unary quantification is in general not possible for other binary quantifiers.

Initiated by Zadeh [19], the literature on *fuzzy quantifiers* is large; we refer to the monograph [12] and the more recent survey article [4] for an overview of relevant work. Focusing on binary quantification, a corresponding fuzzy quantifier is given by a *truth function* $v_I(\widetilde{Q}) : \widetilde{A} \times \widetilde{B} \to [0,1]$ mapping any two fuzzy sets over the domain D into a truth value (*degree of truth*) in $[0,1]$. Here I refers to an interpretation, that assigns a fuzzy set (fuzzy predicate) over D to each monadic predicate symbol. If both, the range predicate $\widetilde{A}$ and the scope predicate $\widetilde{B}$, are crisp, then the quantifier $\widetilde{Q}$ is called *semi-fuzzy*. We identify the domain elements (members of D) with corresponding constants and assume throughout this paper that D is finite. Note that both stipulations are well justified in our database oriented context. This context also motivates the focus on semi-fuzzy quantifiers, since the predicates correspond to the relations of the given dataset, which is assumed to be of the usual relational format. (Indeed, we will not distinguish between a crisp predicate and the corresponding relation of the dataset.) If $v_I(\widetilde{Q})(A,B)$ only depends on the proportion

$$Prop(A,B) =_{df} \tfrac{|A \cap B|}{|A|}$$

then $\widetilde{Q}$ is a binary *proportional* semi-fuzzy quantifier. This is the type of quantifier models that we want to incorporate into a classical query language.

3 Semantics via sampling in Giles's game

As we have seen in the last section, every binary semi-fuzzy proportional quantifier is characterized by a truth function that maps $Prop(A,B)$ into a truth value in $[0,1]$. This triggers the question which of the uncountably many candidates for truth functions are adequate for modeling particular natural language quantifiers like *almost all* or *about half*. Moreover, one may want to embed corresponding formal quantifiers into a standard t-norm based fuzzy logic, following the paradigm of contemporary mathematical fuzzy logic, as represented, e.g., in [3]. To address both challenges, Fermüller and Roschger [5, 6] introduced an extension of Giles's game based semantics for Łukasiewicz logic [10, 11] that allows one to derive the truth functions for certain families of proportional semi-fuzzy quantifiers from rules that regulate attacks on and defenses of logically complex formulas. We do not care about the logical connectives of Łukasiewicz logic here, but rather consider queries that feature a single vague quantifier expression applied to crisp range and scope predicates. For monadic quantifiers (where the

[2] Unary quantification, as in *All are B* and *Some are B*, can be considered as a special instance of binary quantification, where the range predicate A is suppressed since it is identified with the one satisfied by all elements of the range of discourse.

domain D is identified with the range A and thus $Prop(A, B) = Prop(B)$) one may formulate game rules like the following for a family of semi-fuzzy quantifiers G_m^k, taken from [6]:

– If the proponent **P** asserts $\mathsf{G}_m^k xF(x)$ then her opponent **O** may attack this statement by betting against k random instances of $F(x)$, while **P** bets for m random instances of $F(x)$.

Here, 'random instance' refers to a uniformly random choice of a constant d. By betting for (against) the corresponding instance $F(d)$ a player of the game risks to have to pay one unit of money to the opposing player if the corresponding formula turns out to be false (true) in the given interpretation.[3] By identifying the inverse of the expected loss of money of the proponent of a formula with its degree of truth, one obtains truth functions for corresponding semi-fuzzy quantifiers. In this manner truth functions for G_{2n}^n, for $n \geq 1$, can be extracted from the above rule that amount to reasonable models of the natural language quantifier *at least about a third*, parameterized by a certain measure of 'tolerance'. For querying we will not directly refer to truth functions, but rather make use of the observation that each concrete run of the semantic game yields a (dispersive) classical truth value. This yes/no-reply corresponds to a statistical test using the randomly chosen constants as sample. In this manner standard principles of statistics, as explained in the next section, will lead us to a semantics for queries featuring semi-fuzzy proportionality quantifiers.

4 Principles of sampling

We briefly review the theoretical basis of our sampling based evaluation methods [13], which we use to specify the semantics of quantifiers in Section 5.2.

Let $Y_1, \ldots, Y_s$ be independent and identically distributed Bernoulli random variables, i.e. for each $i \in \{1, \ldots, s\}$ we have $Y_i \in \{0, 1\}$. Then, it is easy to see that $\frac{\sum_{i=1}^{s} Y_i}{s} := \frac{X}{s}$ is a random variable with (scaled) binomial distribution. To evaluate binary vague proportional quantifiers, we need to estimate the proportion of range elements that also fulfill the scope formula. To this end we need to relate three parameters, namely the *sample size* s, the *confidence* $1 - \alpha \in (0, 1)$, and the *precision* of the estimate $\epsilon \in [0, 1]$. This can be expressed as follows [13], where $\mathbb{P}_{s,\rho}$ denotes the probability distribution for a binomial distributed random variables with parameters $s \in \mathbb{N}$ and $\rho \in [0, 1]$:

$$\mathbb{P}_{s,\rho}(\mid \frac{X}{s} - \rho \mid \geq \epsilon) \leq \alpha. \tag{1}$$

The most accurate way to proceed would be to construct confidence regions, using binomial and beta quantiles, which should certainly be performed for real

[3] Actually the overall game is more involved than indicated here, since whole *multisets* of formulas have to be considered in general, when decomposing formulas into subformulas in accordance with the rules. For details we refer to [7].

life applications where accuracy matters the most. Another well known and widely used approach, due to its more efficient nature, makes use of the central limit theorem [13], for which we have to assume that the sample size is sufficiently large. Following this way we may calculate:

$$\mathbb{P}_{s,\rho}(\mid \tfrac{X}{s} - \rho \mid < \epsilon) = \mathbb{P}_{s,\rho}(\mid \tfrac{X - s\rho}{\sqrt{s\rho(1-\rho)}} \mid < \epsilon\sqrt{\tfrac{s}{\rho(1-\rho)}}) \approx$$

$$\approx \Phi(\epsilon\sqrt{\tfrac{s}{\rho(1-\rho)}}) - \Phi(-\epsilon\sqrt{\tfrac{s}{\rho(1-\rho)}}) = 2\Phi(\epsilon\sqrt{\tfrac{s}{\rho(1-\rho)}}) - 1.$$

Since Φ is bijective[4] and $\rho(1-\rho) \leq \frac{1}{4}$, we obtain $s \geq (\frac{\Phi^{-1}(\frac{2-\alpha}{2})}{2\epsilon})^2$.

This last inequality tells us which minimum sample size s we have to use to achieve a certain precision ϵ with confidence $1-\alpha$. To refer to this functional relation of the parameters later, we define $f : [0,1] \times (0,1) \to \mathbb{N}$ as follows:

$$f(\epsilon,\alpha) = \lceil (\frac{\Phi^{-1}(\frac{2-\alpha}{2})}{2\epsilon})^2 \rceil. \tag{2}$$

5 Querying with semi-fuzzy quantifiers

In this section, we present our concrete proposal for querying datasets, using a standard query language extended with semi-fuzzy proportional quantifiers.

For ease of presentation, we take a declarative, logic based view of relational databases and queries over them [1]. Databases are defined as finite relational structures over a given signature or *schema*, and we consider first-order logic formulas over the same signature as basic query language. As data values we use constants and integer numbers, and we allow (in)equalities between values (both constants and integers), and comparisons ($<, >, \geq, \neq$) over integers. This basic setting captures relational algebra expressions (and thus, basic SQL) over relational databases. Moreover, significant fragments of other datamodels and their corresponding query languages can be viewed as special case of FO-queries over relational data as considered here. This applies in particular to the fragment of the SPARQL query language for RDF data [17] considered in Section 6

5.1 Relational databases and FO-queries

As usual, we denote by $\mathbb{Z}$ the integer numbers, and by $\mathbb{N}$ the positive integers. We define a *(relational) schema* as comprising a set $\mathcal{R}$ of *relation names*, together with an *arity function* $ar : \mathcal{R} \to \mathbb{N}$, and a function *npos* that maps each R to a (possibly empty) subset of $\{1, \ldots, ar(R)\}$ of *numeric positions* of R.[5]

Let a relational schema $(\mathcal{R}, ar, npos)$ be given. Let $\mathbf{U}$ be a set of constants or *data values*, and $\mathbf{V}$ be a countably infinite set of *variables*. A *term* is an object in $\mathbf{U} \cup \mathbb{Z} \cup \mathbf{V}$. *Atoms* take the following forms:

[4] As it is the distribution function of standard normally distributed random variables.
[5] We use a simple definition, compatible with more complex notions of schema, which may assign, e.g., names and domains to attributes, and integrity constraints.

(i) $R(t_1, \ldots, t_{ar(R)})$ with $R \in \mathcal{R}$, and the t_i are terms such that $t_i \in \mathbb{Z} \cup \mathbf{V}$ if $i \in npos(R)$, and $t_i \in \mathbf{U} \cup \mathbf{V}$ if $i \notin npos(R)$;
(ii) $t = t'$ and $t \neq t'$ with t, t' terms; and
(iii) $t < t'$, $t > t'$, $t \leq t'$ or $t \geq t'$, for t, t' terms in $\mathbb{Z} \cup \mathbf{V}$.

An atom is called *relational* if it is of the form (i), and *ground* if all its terms are from $\mathbf{U} \cup \mathbb{Z}$. A *database instance* (or simply a *database*) is a *finite* set D of ground relational atoms. The *active domain* of a database D, denoted $ADom(D)$ is the set of constants and numbers from $\mathbf{U} \cup \mathbb{Z}$ that occur in the atoms of D.

Example 1. Consider a schema containing, among others, the following relations:
- unary *country* and *city*, with no numeric positions, that is: $npos(city) = \{\}$ and $npos(country) = \{\}$;
- a binary *city_of* that relates cities and the countries they are located in, also with $npos(city\_of) = \{\}$;
- a binary *cap_of* that relates each capital city with the country it is capital of; again, $npos(cap\_of) = \{\}$;
- a binary *has_pop* with $npos(has\_pop) = \{2\}$, which relates countries and cities, with an integer number denoting its total population;
- a binary *hasGDP_agr* with $npos(hasGDP\_agr) = \{2\}$, which relates countries with an integer number (between 0 and 100) denoting the percentage of its GDP that comes from agriculture.

A database D_1 over this schema may contain, for example, ground atoms:
$country(USA)$, $country(India)$, ... $city(Chicago)$, $city(Beijing)$, ...
$cap\_of(Beijing, China)$, $cap\_of(NewDelhi, India)$, ...
$has\_pop(China, 1385{*}10^6)$, $has\_pop(Beijing, 21.6{*}10^6)$, $has\_pop(Shanghai, 24.3{*}10^6)$ ...

An *FO-query* is a first-order formula $\psi(\boldsymbol{x})$ with free variables $\boldsymbol{x}$, built from atoms in the usual way, using the connectives $\neg$, $\wedge$, $\vee$, and the quantifiers $\exists$ and $\forall$. We refer to these variables as the *answer variables* of ψ. The *arity* of the query is the number of variables in $\boldsymbol{x}$. We call a query *Boolean* if it is 0-ary, that is, it has no free variables.

We note that a database D can be seen as a Herbrand interpretation over the predicates in the schema, and with domain $ADom(D)$. An n-ary FO query over D defines an n-ary relation over $ADom(D)$, which contains precisely the tuples for which the corresponding formula is satisfied, under the usual semantics.

Let D be a database. A *substitution* is a mapping σ from variables in $\mathbf{V}$ to values in $ADom(D)$. We write $\sigma(\boldsymbol{t})$ for the tuple that results from $\boldsymbol{t}$ by substituting each variable x with $\sigma(x)$, and we write $\sigma(\varphi)$ to denote the formula that results from φ by applying the substitution σ to all its atoms. For $x \in \mathbf{V}$, $c \in ADom$, and a substitution σ, we denote by $\sigma\{x \mapsto c\}$ the substitution σ' that has $\sigma'(x) = c$, and $\sigma'(y) = \sigma(y)$ for all remaining variables in the domain of σ. Abusing notation, we may disregard order in tuples and treat them as sets.

The *satisfaction* in D of a formula ψ w.r.t. σ, in symbols $D \models_\sigma \psi$, is defined inductively in the natural way:
- For relational atoms, $D \models_\sigma R(\boldsymbol{t})$ if $R(\sigma(\boldsymbol{t})) \in D$.

– For the other atoms, $D \models_\sigma t \circledcirc t'$ if $\sigma(t) \circledcirc \sigma(t')$, where each binary predicate $\circledcirc \in \{=, \neq, \geq \ldots.\}$ is interpreted as usual.
– $D \models_\sigma \psi_1 \wedge \psi_2$ if $D \models_\sigma \psi_1$ and $D \models_\sigma \psi_2$.
– $D \models_\sigma \psi_1 \vee \psi_2$ if $D \models_\sigma \psi_1$ or $D \models_\sigma \psi_2$.
– $D \models_\sigma \neg\psi$ if $D \not\models_\sigma \psi$.
– $D \models_\sigma \exists x\ \psi$ if for some $c \in ADom(D)$, we have $D \models_{\sigma\{x \mapsto c\}} \psi$.
– $D \models_\sigma \forall x\ \psi$ if for each $c \in ADom(D)$, we have $D \models_{\sigma\{x \mapsto c\}} \psi$.

Let $\psi(\boldsymbol{x})$ be a query with answer variables $\boldsymbol{x} = (x_1, \cdots, x_n)$, and let $\boldsymbol{c} = c_1, \cdots, c_n$ be a tuple of values from $\mathbf{U} \cup \mathbb{Z}$ of the same arity. Then we say that $\boldsymbol{c}$ *is an answer to* ψ *over* D if $D \models_\sigma \psi$ for the substitution σ with $x_i = c_i$ for each $1 \leq i \leq n$. In this case, we may write $D \models \psi(\boldsymbol{c})$.

Note that for ψ a Boolean query, there are two possibilities: if $D \models \psi$, then the empty tuple is an answer to ψ. In this case, we may say that ψ is *true* in D, or that its answer in D is *yes*. In the other case, if $D \not\models \psi$, then the empty tuple is *not* an answer to ψ: we say that ψ is *false* or that its answer is *no*.

Example 2. The following are simple examples of FO-queries over our example schema; ψ_1 is a Boolean query, while ψ_2 is unary and ψ_3 is binary.
ψ_1: Is there a country with a population of more than one billion people?
ψ_2: Which countries have a city with higher population than its capital?
ψ_3: Which are the countries, and their capitals, such that no other city in the country has more inhabitants?

$$\begin{aligned}
\psi_1 :=\ & \exists x, y(\mathit{country}(x) \wedge \mathit{has\_pop}(x, y) \wedge (y > 1000{*}10^6))\\
\psi_2(x) :=\ & \exists y_1, y_2, z_1, z_2(\mathit{country}(x) \wedge \mathit{cap\_of}(y_1, x) \wedge \mathit{city\_of}(y_2, x) \wedge\\
& \wedge \mathit{has\_pop}(y_1, z_1) \wedge \mathit{has\_pop}(y_2, z_2) \wedge (z_1 < z_2))\\
\psi_3(x, y) :=\ & \exists z\big(\mathit{country}(x) \wedge \mathit{cap\_of}(y, x) \wedge \mathit{has\_pop}(y, z) \wedge\\
& \forall y_1, z_1((\mathit{city\_of}(y_1, x) \wedge \mathit{has\_pop}(y_1, z_1)) \rightarrow (z_1 \leq z))\big)
\end{aligned}$$

We note that $D_1 \models \psi_1$, that is, its answer is *yes*, since the substitution $\sigma(x) = \mathit{China}$, $\sigma(y) = 1385{*}10^6$ makes the formula true. We can also observe that the answers to ψ_2 contain *China*, and that (*Beijing*, *China*) is *not* an answer to ψ_3.

5.2 Extending FO-queries with proportional quantifiers

Assume $m \in \mathbb{N}$, and let $n, k \in \{0, \ldots, m\}$ with $n \neq 0$. We consider the following *proportional quantifiers*:

$Q^{[\approx \frac{k}{n}]}$ about k/n $\quad$ $Q^{[\gtrapprox \frac{k}{n}]}$ at least about k/n $\quad$ $Q^{[\lessapprox \frac{k}{n}]}$ at most about k/n

If $k = n$, then we may read both $Q^{[\approx \frac{k}{n}]}$ and $Q^{[\gtrapprox \frac{k}{n}]}$ as *almost all*, and simply write $Q^{[\approx 1]}$. If $k = 0$, then we may read $Q^{[\approx \frac{k}{n}]}$ and $Q^{[\lessapprox \frac{k}{n}]}$ as *almost none* and write $Q^{[\approx 0]}$. Note that each value of m determines a family of proportional quantifiers. Throughout the paper we assume $m = 4$, but any number can be used.[6] Now we define our query language, which extends FO-queries with these quantifiers:

[6] We remark that very large values of m do not usually occur in natural language.

Definition 1 (Queries). *A* query *is an expression $q(\boldsymbol{y})$ of the form*

$$\widetilde{Q}x\big(R(x,\boldsymbol{y'}),\psi(x,\boldsymbol{y})\big)$$

where $\boldsymbol{y'} \subseteq \boldsymbol{y}$, and:
- *$\widetilde{Q}$ is any of the proportional quantifiers defined above,*
- *$R(x,\boldsymbol{y'})$ is a relational atom using the variables in $\{x\} \cup \boldsymbol{y'}$, and whose additional terms are from $\mathbf{U} \cup \mathbb{Z}$, and*
- *$\psi(x,\boldsymbol{y})$ is an FO-query with answer variables $\{x\} \cup \boldsymbol{y}$.*

The answer variables *of q are $\boldsymbol{y}$, and its* arity *is the number of variables in $\boldsymbol{y}$.*

Example 3. To illustrate our language, we consider the following queries:
q_1: Do at least about three quarters of all countries make at most 20% of their GDP in agriculture?
q_2: Do about half of all cities have more than 200000 inhabitants?
q_3: Which countries have a capital which has more inhabitants than about half of all other capitals in the world?
q_4: Which countries have a capital that has more inhabitants than almost all other cities of that country?

The first two are Boolean queries, the other two are unary. Queries q_3 and q_4 are very similar, but they differ on the range predicate: it is unary in q_3 and binary in q_4. In our syntax, they look as follows:

$$\begin{aligned}
q_1 &:= Q^{[\gtrapprox \frac{3}{4}]}x\big(\mathit{country}(x), \exists y(\mathit{hasGDP\_agr}(x,y) \wedge (y \leq 20))\big)\\
q_2 &:= Q^{[\approx \frac{1}{2}]}x\big(\mathit{city}(x), \exists y(\mathit{has\_pop}(x,y) \wedge (y > 200000))\big)\\
q_3(y) &:= Q^{[\approx \frac{1}{2}]}x\big(\mathit{capital}(x), \exists z,z',w(\mathit{cap\_of}(y,w) \wedge \mathit{has\_pop}(w,z) \wedge\\
&\qquad\qquad \wedge\, \mathit{has\_pop}(x,z') \rightarrow (z > z'))\big)\\
q_4(y) &:= Q^{[\approx 1]}x\big(\mathit{city\_of}(x,y), \exists z,z',w(\mathit{cap\_of}(y,w) \wedge \mathit{has\_pop}(w,z) \wedge\\
&\qquad\qquad \wedge\, \mathit{has\_pop}(x,z') \rightarrow (z > z'))\big)
\end{aligned}$$

We note that $D_1 \models q_1$, that is, its answer is *yes*, since the substitution $\sigma(x) = \mathit{China}$, $\sigma(y) = 1385{*}10^6$ makes the formula true. We can also observe that the answers to q_2 contain *China*, and that (*Beijing*, *China*) is *not* an answer to q_3.

Now we define the semantics of our query language. As we have anticipated, it is based on *sampling*, according to the principles discussed in Section 4. We assume that values in the interval $[0,1]$ are given for the confidence $1-\alpha \in (0,1)$ and precision ϵ. These values then determine a minimal sample size $s = f(\epsilon,\alpha)$ as in Equation 2. Then, for testing whether a given tuple of variables $\boldsymbol{c} = c_1,\cdots,c_n$ of values from $\mathbf{U} \cup \mathbb{Z}$ is an answer to a query $\widetilde{Q}x\big(R(x,\boldsymbol{y'}),\psi(x,\boldsymbol{y})\big)$, we take a sufficiently large random sample of objects x that satisfy $R(x,\boldsymbol{c'})$ (where $\boldsymbol{c'}$ is the restriction of $\boldsymbol{c}$ to the positions from $\boldsymbol{y}$ that occur in $\boldsymbol{y'}$), and verify whether the proportion of those for which $\psi(x,\boldsymbol{c})$ holds are within the desired range. Note that, since the sample is random, for the same tuple $\boldsymbol{c}$, we may get different proportions, and hence a different value, if we repeat the query evaluation. This is natural, since our semantics of the proportional quantifiers defines a probability function over the possible answer tuples. As we will illustrate in the next section, the answers retrieved in this way are reliable, even for modest sample sizes.

Definition 2 (Semantics). *Let D be a database, let $R(x, \boldsymbol{c'})$ be a relational atom and $\psi(x, \boldsymbol{c})$ be a FO-query, such that x is the only free variable in both. Let $S \subseteq ADom(D)$ with $S \neq \emptyset$. We define*

$$Prop_D(S, \psi(x, \boldsymbol{c})) = \frac{|\{c \in S \mid D \models_{\{x \mapsto c\}} \psi(x, \boldsymbol{c})\}|}{|S|}$$

Now we let σ be a substitution from $\mathbf{V}$ to $ADom(D)$, and let $D_R = \{c \in ADom(D) \mid R(c, \sigma(y')) \in D\}$. We define the semantics of queries as follows:

- $D \models_{\sigma,S,\epsilon} Q^{[\approx \frac{k}{n}]}x\big(R(x, \boldsymbol{y'}), \psi(x, \boldsymbol{y})\big)$ *if* $S \subseteq D_R$ *and* $Prop_D(S, \sigma(\psi(x, \boldsymbol{y}))) \in \big[\frac{k}{n} - \epsilon, \frac{k}{n} + \epsilon\big]$.
- $D \models_{\sigma,S,\epsilon} Q^{[\gtrapprox \frac{k}{n}]}x\big(R(x, \boldsymbol{y'}), \psi(x, \boldsymbol{y})\big)$ *if* $S \subseteq D_R$ *and* $Prop_D(S, \sigma(\psi(x, \boldsymbol{y}))) \in \big[\frac{k}{n} - \epsilon, 1\big]$.
- $D \models_{\sigma,S,\epsilon} Q^{[\lessapprox \frac{k}{n}]}x\big(R(x, \boldsymbol{y'}), \psi(x, \boldsymbol{y})\big)$ *if* $S \subseteq D_R$ *and* $Prop_D(S, \sigma(\psi(x, \boldsymbol{y}))) \in \big[0, \frac{k}{n} + \epsilon\big]$.

Let ϵ and α in the interval $[0, 1]$ be given. A tuple $\boldsymbol{c} = c_1, \cdots, c_n$ of values from $\mathbf{U} \cup \mathbb{Z}$ of the same arity as $\boldsymbol{y}$ is called a sampled answer to ψ over D (with precision ϵ and confidence $1 - \alpha$) *if $D \models_{\sigma,S,\epsilon} Q^{[\gtrapprox \frac{k}{n}]}x\big(R(x, \boldsymbol{y'}), \psi(x, \boldsymbol{y})\big)$, where σ is the substitution with $x_i = c_i$ for each $1 \leq i \leq n$, and $S \subseteq ADom(D)$ is a random sample (with replacement) of size $|S| \geq f(\epsilon, \alpha)$ as described in Section 4. In this case, we may write $D \models_{\epsilon,\alpha} \psi(\boldsymbol{c})$.*

6 Proof-of-concept: Querying the MONDIAL database

For illustrating the proposed approach on real life data, we chose the MONDIAL database[7]. It is a dataset containing geographical data, that relies on open web data, such as the CIA factbook, Wikipedia, and atlases. The last major revision took place in 2015. Like most open web data, the database is not complete, and data may be somewhat imprecise. However, this is not of major concern here.

We evaluated the queries in Example 3. (In fact, the schema and queries of our running example are based on MONDIAL). We used the RDF version of MONDIAL locally and posed standard SPARQL queries, using the Java extension Apache Jena. This, together with random sampling on the list of query results, suffices to simulate the evaluation of queries in our language efficiently. In contrast to other fuzzy querying approaches, like the ones of Bosc and Pivert [18] we here rely on strictly classical data, and focus on a probabilistic evaluation of them. Our goal was to test how the sampling based evaluation performs for particular sample queries. Obviously, if the amount of sampled data increases, then the difference between the evaluation times for full and partial answers,

[7] MONDIAL database. (Last accessed January 30th, 2017). Retrieved from: https://www.dbis.informatik.uni-goettingen.de/Mondial/

respectively, increases as well. However, for the present example, the MONDIAL database (16.4MB), they are still in a similar range. Some of our observations are captured in Figures 1 and 2, which show how the sample size correlates with the calculated proportions, using only a single iteration per size. Figure 1 shows the results for the queries q_1–q_3, and Figure 2 for particular instances of the answer variable y in query q_4. From those results one can straightforwardly evaluate the answers to the natural language queries. Taking the first one, which asks whether the proportion in question is at least about 75 percent, the results show that, for almost all samples sizes, this is the case with (high) confidence $1 - \alpha = 0.95$. Similar results hold for the other queries. (Note that in Figure 2 just a small range of proportions is displayed). Finally, we emphasize that the graphs show the proportions obtained for *one* random sample of each size. But the blue line (sampled results) converges quickly to the red line (correct proportion) if we increase the number of iterations.

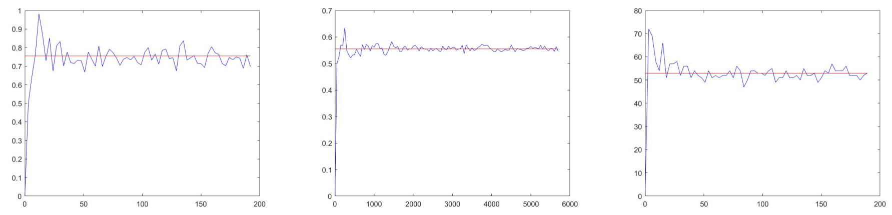

Fig. 1. Left: query q_1; middle: query q_2; right: query q_3. The x-axis always represents possible sample sizes, i.e. the number of domain elements that fulfill the respective range predicate. For the left and the middle picture, the y-axis stands for the proportion of these range objects that also fulfill the scope predicate, while for the right picture it displays the sizes of answer sets. The blue graph shows the achieved results for one sample of the sizes from the x-axis. The red graph displays the correct proportions, or answer set size respectively.

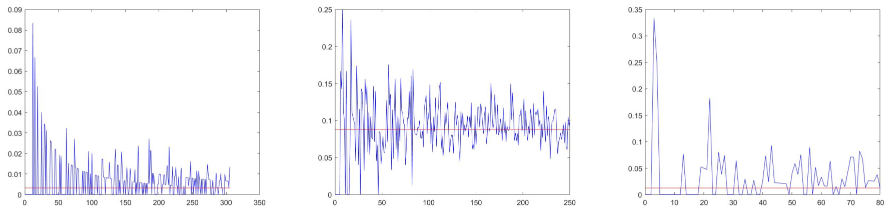

Fig. 2. Query q_4, for: left: y = China; middle: y = USA; right: y = India. The x-axis always represents possible sample sizes, i.e. the number of domain elements that fulfill the respective range predicate. The y-axis stands for the proportion of these range objects that also fulfill the scope predicate. The blue graph shows the achieved results for sample sizes from the x-axis. The red graph displays the correct proportions.

7 Conclusions and Outlook

We have presented an extension of a classical first order query language over relational data with quantifiers that model vague natural language quantifiers like *about half*, *almost all*, *at least about a third*, and *at most about a quarter*.

The proposed semantics of these quantifiers is inspired by an extension of Giles's semantic game [10] for Łukasiewicz logic to semi-fuzzy proportional quantifiers that makes use of random selection of witnessing constants [5, 6]. In the context of Giles game, fuzzy truth functions for these quantifiers are extracted by identifying expected values resulting from sampling based games with degrees of truth. Similarly, our queries trigger a sampling mechanism to return probabilistic answers; i.e. answers that might differ upon repetition, but that conform to the specified meaning of the quantifier expressions with high probability, given particular levels of confidence and precision. We have specified this semantics in a manner that directly extends standard classical FO-queries and thus fits well into the usual frameworks for querying relational and RDF data. As a proof-of-concept we applied the proposed machinery to the RDF version of the MONDIAL database, illustrating that our approach yields promising results that encourage further investigations.

We conclude by briefly commenting on some further challenges, possible extensions, and additional use cases.

Other vague quantifier expressions: The quantifier expressions selected in Section 5.2 are only specific examples, illustrating a general principle, but many other interesting natural language quantifiers could be considered. A particular challenge arises for modeling the often used expressions *many* and *few*, since their meaning does not just depend on the proportion of elements satisfying the scope predicate, but rather calls for considerations of context as well as user expectations, as pointed out by linguists (see, e.g., [8, 16, 15]).

Introducing vague predicates: We have only considered classical relational data here, where all relations and predicates are bi-valent. The ideas underlying our approach to vague quantifier semantics can also be applied to derive fuzzy models of predicates like *large*, *small*, etc. Developing such an approach and comparing it with existing fuzzy databases might be useful.

Computational gains: Sampling may not only be viewed as a tool for modeling vagueness, but is also a well known approach to obtain more efficient, approximate answers in face of huge volumes of data [2]. Our current prototype implementation is not intended to illustrate such computational gains, but this is clearly an interesting topic for further research.

Data summarization: While we have focused on querying here, we finally want to point out that our approach to modeling vague quantifiers may be even more useful for data summarization. It seems natural to offer sentences like *About half of the provincial capitals have airports* and *Almost all countries have ethnic minorities* summaries instead of precise statistical data, in particular if the later are marred by spurious precision.

Acknowledgements

This work was supported by the Austrian Science Fund (FWF) projects I1897-N25 and W1255-N23.

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of databases: the logical level.* Addison-Wesley Longman Publishing Co., Inc., 1995.
2. S. Agarwal, B. Mozafari, A. Panda, H. Milner, S. Madden, and I. Stoica. Blinkdb: queries with bounded errors and bounded response times on very large data. In *Proceedings of the 8th ACM European Conference on Computer Systems*, pages 29–42. ACM, 2013.
3. P. Cintula, C. G. Fermüller, P. Hájek, and C. Noguera, editors. *Handbook of Mathematical Fuzzy Logic (in three volumes).* College Publications, 2011 and 2015.
4. M. Delgado, M. Ruiz, D. Sánchez, and M. Vila. Fuzzy quantification: a state of the art. *Fuzzy Sets and Systems*, 242:1–30, 2014.
5. C. Fermüller and C. Roschger. Randomized game semantics for semi-fuzzy quantifiers. In *Advances in Computational Intelligence*, volume 300, pages 632–641. Springer, 2012.
6. C. Fermüller and C. Roschger. Randomized game semantics for semi-fuzzy quantifiers. *Logic Journal of the IGPL*, 22(3):413–439, 2014.
7. C. G. Fermüller. Semantic games for fuzzy logics. In P. Cintula, C. G. Fermüller, and C. Noguera, editors, *Handbook of Mathematical Fuzzy Logic - Volume 3*, pages 969–1028. College Publications, 2015.
8. T. Fernando and H. Kamp. Expecting many. In *Semantics and Linguistic Theory*, volume 6, pages 53–68. 1996.
9. P. B. Gibbons and Y. Matias. New sampling-based summary statistics for improving approximate query answers. In *ACM SIGMOD Record*, volume 27, pages 331–342. ACM, 1998.
10. R. Giles. A non-classical logic for physics. *Studia Logica*, 33(4):397–415, 1974.
11. R. Giles. Semantics for fuzzy reasoning. *International Journal of Man-Machine Studies*, 17(4):401–415, 1982.
12. I. Glöckner. *Fuzzy quantifiers: A computational theory*, volume 193 of *Studies in Fuzziness and Soft Computing.* Springer Verlag, 2006.
13. G. Grimmett and D. Welsh. *Probability: An Introduction.* Oxford UP, 2014.
14. J. Kacprzyk, S. Zadrożny, and G. De Tré. Fuzziness in database management systems: half a century of developments and future prospects. *Fuzzy Sets and Systems*, 281:300–307, 2015.
15. S. Lappin. An intensional parametric semantics for vague quantifiers. *Linguistics & Philosophy*, 23(6):599–620, 2000.
16. B. Partee. Many quantifiers. In *Proceedings of ESCOL 5*, pages 383–402, 1988.
17. J. Pérez, M. Arenas, and C. Gutierrez. Semantics and complexity of sparql. *ACM Transactions on Database Systems (TODS)*, 34(3):16, 2009.
18. O. Pivert and P. Bosc. *Fuzzy preference queries to relational databases.* World Scientific, 2012.
19. L. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computers & Mathematics with Applications*, 9(1):149–184, 1983.
20. L. Zadeh. Fuzzy logic. *IEEE: Computer*, 21(4):83–93, 1988.