Data with Logical and Statistical constraints

Michel de Rougemont[0000−0001−6518−8874]

Abstract Descriptive Complexity and Algorithmic Complexity theory both use an analysis in the worst-case. However, hard problems such as SAT become much easier when we relax the worst-case condition. We introduce the notion of statistical queries which take finite structures as inputs and return distributions on finite domains. A statistical constraint is a relation between statistical queries. We use the notion of a stochastic approximation [15] for structures which satisfy a statistical constraint and can be generated with a distribution μ. A hard problem is approximable with an algorithm A if A is correct on YES instances with high probability, and on NO instances generated by μ with high probability. We explain how a generalization of Maxclique is easy on graphs which follow a power law degree distribution, even if the graph is given as a stream of edges.

1 Introduction

Logical constraints are sentences in some logic $\mathcal{L}$, mostly First and Second-order Logic. The *Consistency* problem decides if there is a model which satisfies these constraints. The *Entailment* problem decides if a constraint is a logical consequence of a set $\phi_1, ...\phi_k$ of constraints. These problems are central both for Logic and Computer Science, more specifically for *Finite Model theory* and *Data Base theory* which share many central concepts. Janos Makowsky made central contributions at the intersection of these areas, which have led to new fundamental notions, published for example in [6, 7, 13].

The importance of finite structures and the link to Complexity theory are central for the understanding of what *efficiently computable* is. In the classical Complexity theory, the SAT problem has been a fundamental NP-complete problem with natural extensions to optimization problems such as Maxsat, extended versions such as QBF

Michel de Rougemont
University Paris II e-mail: mdr@irif.fr

(Quantified Boolean Formula) and counting version such as #SAT. The Descriptive Complexity characterizes all these problems with Logic-based languages. They are considered hard because we do not know polynomial time algorithms for their exact solutions. Logic-based approaches such as [7] identified subclasses for which there exist solutions in polynomial time.

The SAT competition gives a different point of view however. Very efficient heuristics have solved these problems for larger and larger instances over the years. In [3], a recent history of these heuristics is presented. In general, a SAT instance is first analyzed with statistical methods and the space of inputs is then partitioned into several areas. Different heuristics are used for each area and provide efficient solutions in practice.

The notion of *effective computability* therefore needs to be better understood. An $O(n)$-time algorithm must be linear for all inputs of size n, for both Computational and Descriptive Complexity which both use the *worst-case complexity*. It is however possible that an algorithm A would require $O(2^n)$ time on a few inputs, while all other inputs require only $O(n)$-time. How easy are these inputs to generate? For SAT, there are hard instances, but they are either randomly generated or hard to generate with a deterministic algorithm. There have been at least three possible approaches for a better understanding of efficient algorithms when problems are hard in the worst-case:

- Find classes of inputs for which hard problems become easy. Bounded Clique-width for graphs is an example which introduces a new parameter k, as shown by Janos Makowsky and his co-authors in [7],
- Use Approximations, on inputs and outputs,
- Relax the Worst-case Complexity.

This paper presents an approach based on a *statistical property* which relaxes the Worst-case Complexity. We show how some NP-hard problems can become easy on finite structures which satisfy such a property, using the notion of a *1-sided stochastic randomized* approximation algorithm A of a decision problem, defined in [15]. We consider random inputs of size n which follow some statistical property, generated by a distribution μ. On Yes instances, the algorithm accepts with high probability, and on NO instances generated by μ, the algorithm rejects with high probability (typically $2/3$ or $1-\delta$). We only guarantee a correct answer on NO instances when the input follows the statistics and the probabilistic space is the product of $\mu \times \Omega$ where Ω is the probabilistic space of the algorithm.

As an example, we take graphs whose degree distribution follows a power law (a specific statistical constraint) and we explain how a generalization of Maxclique can become easy when the graph is given as a stream of edges. We survey this approach, based on statistical constraints, which restricts the class of inputs and take Words, Graphs and relational databases as a source of structures for which there are natural statistical constraints.

In the second section, we review some classical results concerning logical constraints, classical approximations for search and decision problems and average complexity. In the third section, we present the statistical queries and constraints,

for classes of Words, Graphs and Datawarehouses, i.e. relational structures used for data analysis. In the fourth section, the notion of a stochastic approximation for structures which satisfy some statistical constraint is presented and we discuss the case of graphs which follow a power law degree distribution.

2 Classical approaches

We review three different approaches to understand the complexity of search and decision problems and the existence of efficient algorithms.

2.1 Logical constraints

The importance of finite structures for the *Entailment* problem was stressed in [6]. It was shown that for some class of constraints called *Embedded Implicational Depencies*, the *Entailment* problem on finite structures is co-recursively enumerable complete. It is one of first results on *Finite Model theory*, which has become a mainstream subject.

The study of density functions of graph properties definable in Monadic Second Order Logic started with the work of Blatter and Specker [4]. This rich subject on MSO definable graph properties was extended in [13]. The case of relations of arity 4, studied in[8], shows a very different situation, relevant for database theory where relations may have a large arity.

If MSO-definable properties on graphs can be NP-complete, hence hard, which additional constraint implies a feasible solution? In [7], Bruno Courcelle, Janos Makowsky and Udi Rotics propose the bounded Clique-width property and proved that MSO properties on graphs with bounded Clique-width have a linear time solution. This influential result has started an entire new research area.

Other graph properties such as bounded treewidth have similar properties and lead to study the time complexity of an algorithm as a function of n and other graph parameters, hence *parametrized complexity*. In particular, when the time complexity is polynomial in n and exponential in the graph parameters [17].

2.2 Approximation

A search problem returns a numerical value and defines a function f such that on input x, we have $f(x) = y$. A randomized algorithm A (ε, δ)-approximates the search problem if $\mathbb{P}rob[\mid A(x) - f(x) \mid < \varepsilon] > 1 - \delta$ for an additive error and $\mathbb{P}rob[\mid A(x) - f(x) \mid < \varepsilon.f(x)] > 1 - \delta$ for a multiplicative error.

For a decision problem Q, this definition is inadequate and we need to shift the approximation on the input. Assume a distance dist on the inputs x, such as the Edit distance on Words, Trees and Graphs, and a model where we query the input: given a predicate P of arity k and arguments $a_1, ...a_k$, we ask if $P(a_1, ...a_k)$ is true or not. We extend the distance to a property Q as $\mathsf{dist}(x, Q) = Min_{x' \in Q}\mathsf{dist}(x, x')$ and say that x is ε-far from Q if $\mathsf{dist}(x, Q) \geq \varepsilon$. An (ε, δ)-Tester is a randomized algorithm A such that:

- If $x \in Q$ then A accepts with probability 1,
- If x is ε-far from Q then $I\!\!P rob[A(x)\text{rejects}] > 1 - \delta$.
- The query complexity is independent of n, the size of x, and depends only on ε and δ.

This definition is 1-sided, but can be generalized to a 2-sided version. The query complexity can also be extended to some sublinear function of n. Both definitions assume a worst-case situation.

2.3 Average-case Complexity

The natural approach, originated by Levin [14], considered a distribution $\mathcal{D}$ on the inputs and required that the expected time complexity on $\mathcal{D}$ be bounded by a function of n. Given a reduction between *Distributional problems*, problems with a distribution $\mathcal{D}$, [14] presented a complete problem for polynomial time computable distributions. More advanced results are presented in [5].

On the algorithmic side, the *non worst-case analysis* [19] presents the analysis of algorithms on specific distributions. In section 4, we consider inputs which satisfy statistical constraints and a distribution μ on these inputs. We then consider a probabilistic space which is a product of $\mu \times \Omega$ where Ω is the probabilistic space of the algorithm. The algorithm is correct only on inputs generated by μ.

3 Statistical constraints

A statistical query generalizes the notion of a *query* on a class $\mathcal{K}$ of finite structures, as a function which takes a finite structure $U_n \in \mathcal{K}$, whose domain is of size n, as input and returns a relation on U_n of arity r. A *statistical query* takes a finite structure $U_n \in \mathcal{K}$ as input and returns a multivariate distribution δ on U_n of arity $r \geq 1$. A *statistical constraint* is a relation between distributions, for example the equality or the proximity $\mathsf{dist}(\delta, \delta') \leq \varepsilon$ relation for some dist function between distributions, and is similar to a boolean query. On relational structures, statistical queries are often called *OLAP* (OnLine Analytical Processing) queries and defined in SQL with the *GROUP BY* expression.

In general, we take a class $\mathcal{K}$ of finite structures of size n augmented with the set Q of rationals with the basic arithmetical operations as parameters. We construct terms and formulas to define the statistical queries. The use of specific distances between distributions is a central element of this approach. The difference with the Average-case complexity is that the distribution of inputs concerns inputs of size n, which satisfy some statistical constraints and not arbitrary distributions $\mathcal{D}$ on the inputs. In section 4, we introduce the *1-sided-stochastic approximation*, similar to the approximation in Property Testing.

Let $\mathcal{L}$ be a logic on a class $\mathcal{K}$ of finite structures with several domains. Consider a First-order Σ_1 formula $\psi(x_1)$ with the free variable $x_1 \in D$ and some existential quantifier. We may have several domains, we write:

$$\psi(x_1 \in D) : \exists y_1 \in D' \ \varphi(x_1, y_1)$$

to specify that x_1 ranges over D and y_1 ranges over $D' = \{1, 2...n\}$. We say that $\psi(x_1)$ is *separating* on a relational structure $U = (D, D', R_1, ...R_k)$ if the sets $W_a = \{b : \varphi(a, b)\}$ are disjoint for each $a \in D$. Observe that if there is a functional dependency $y_1 \rightarrow x_1$, then $\psi(x_1)$ is always *separating*.

The *counting formula* $\psi_c(x_1)$ defines the function which associates to each $a \in D$ the number of distinct values of y_1, i.e. $|W_a|$ and we write:

$$\psi_c(x_1) : \#y_1 \ \varphi(x_1, y_1)$$

We can also write $\psi_c(a) =\mid \{b : \varphi(a, b)\} \mid$.

When the formula $\psi(x_1)$ is separating for the domain D of cardinality m, we can introduce the *distribution formula* $\psi_d(x_1)$ which defines the distribution of values for $x_1 = a_1, .., a_m$:

$$\psi_d(x_1) : \%y_1 \quad \varphi(x_1, y_1)$$

We define $\psi_d(a) = \frac{\psi_c(a)}{\sum_a \psi_c(a)} = \frac{\psi_c(a)}{n}$ where n is the size of D', the domain of y_1, assuming the dependency $y_1 \rightarrow x_1$. In this case, $\sum_a \psi_d(a) = 1$ and $\psi_d(x_1)$ is a 1-dimension distribution. In general we may have $\psi_d(x_1, x_2, ...x_d)$ for a distribution of dimension d. On Words, Graphs and Datawarehouses (specific relational structures), there are functional dependencies which guarantee that the formulas are separating and therefore define statistical queries. On Datawarehouses (see section 3.3), if the free variables $x_1, x_2, ...x_d$ of the query are *analysis variables*, then the distributions are well defined because

$$y_1 \rightarrow x_1, x_2, ...x_d$$

Given a distance dist between 2 distributions, a *Statistical constraint* is a relation between two distributions δ_1, δ_2, either equality $\delta_1 = \delta_2$ or $\mathsf{dist}(\delta_1, \delta_2) < \varepsilon$ for a specific distance dist between distributions.

3.1 Words

A binary word $w_n \in \{0,1\}^n$ is classically represented by the finite structure:

$$W_n = (\{1, 2...n\}, P_0, P_1, <)$$

where the domain is the set $\{1, 2...n\}$ ordered with the binary predicate $<$, and unary predicates P_i for $i = 0, 1$ such that $P_i(j)$ is true if $w_n[j] = i$. On a large alphabet $\Sigma_m = \{a_1, a_2...a_m\}$ where each a_i is a symbol, it is more convenient to consider a word w_n of length n as a finite structure with two domains such as:

$$U_n = (\{1, 2...n\}, \Sigma_m, P, <)$$

where the binary relation $P \subseteq \Sigma_m \times \{1, 2...n\}$ is defined by $P(i, j)$ is true if $w_n[j] = a_i$, i.e. the letter a_i appears in position j.

A statistical query gives, for example, the distribution of the occurrences of each letter, where the occurrence function $\mathsf{occ} : \Sigma_m \to \{1, 2...n\}$ is such that for each letter $a_i \in \Sigma_m$, $\mathsf{occ}(a_i) = \#a_i$ where $\#a_i$ is the number of occurrences of the letter a_i. The frequency function $f : \{1, 2...n\} \to \{0, 1, 2...n\}$ gives the *ordered* occurrences, i.e. $f(i + 1) \leq f(i)$ holds for $i = 1, ..n - 1$. So $f(1) = \#a_i$ is the number of occurrences of the most frequent letter a_i and $f(n) = \#a_j$ is the number of occurrences of the least frequent letter. The relative occurrence occ' is defined as $\mathsf{occ}'(a_i) = \mathsf{occ}(a_i)/n$ and the relative frequency f' is defined as $f'(i) = f(i)/n$. Both functions take values on the rationals $\mathbb{Q}$. Consider the two words $aaabb$ and $bbbaa$: they have the same frequencies f but different occurrence functions occ.

Both occ' and f' are distributions as $\sum_i \mathsf{occ}'(i) = \sum_i f'(i) = 1$. The typical application is when the size m of the alphabet is large and n is very large. Typically, Large language Models read the whole internet, $m \simeq 3.10^4$ is the number of *tokens*, the basic elements defined by the Byte-Pair Encoding algorithm [9], and $n \simeq 10^{12}$. A strict statistical constraint states that occ' is for example the uniform distribution, i.e.

$$\forall i \;\; \mathsf{occ}(a_i) = n/m$$

when m divides n. In general, we accept rounding errors and a statistical constraint is the property:

$$\forall i \;\; \mathsf{occ}'(i) \simeq 1/m$$

where $\simeq$ is the classical approximation on $\mathbb{Q}$. Hence the class of structures is:

$$V_n = (\{1, 2...n\}, \Sigma_m, P, <; \mathbb{Q}, +, *, /)$$

as the set of rational numbers $\mathbb{Q}$ and the arithmetical functions $+, *, /$ can be used to define basic terms. Another statistical constraint would apply on f', for example f' is a Zipf distribution:

$$\forall i \;\; f'(i) \simeq c/i^2$$

where c is a constant which depends on n, the size of the structures. The term c/i^2 takes values in $\mathbb{Q}$.

A k-gram is the generalization of occ' to factors of length k, i.e. consecutive letters. We write $\mathsf{ustat}_k(w_n)$ as the function from $(\Sigma_m)^k$ into $[0,1]$. It gives the probability to observe a k-factor in a word of length n when sampling a uniform position $1 \leq i \leq n-k+1$, hence the name *uniform statistics*.

$$\mathsf{ustat}_k(w_n) = \frac{1}{n-k+1}. \begin{pmatrix} \#w_1 \\ \#w_2 \\ ... \\ \#w_p \end{pmatrix}$$

where w_i is the i-th k-factor ordered lexicographically. As an example:

$$\mathsf{ustat}_2(aaabb) = \frac{1}{4}. \begin{pmatrix} 2 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \frac{1}{4}. \begin{pmatrix} \#aa = 2 \\ \#ab = 1 \\ \#ba = 0 \\ \#bb = 1 \end{pmatrix}.$$

A k-gram defines the next distribution: given a $(k-1)$-factor what is the distribution of the next letter? For example, for $w_5 = aaabb$, $k = 2$ and the factor a, the distribution of the next letter is:

$$\mathsf{next}_a(aaabb) = \frac{1}{3}. \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

There are 3 factors of length 2 which start with an a: two of them have an a as the next letter letter and one of them has a b as the next letter. Similarly, if the factor starts with a b, the distribution is:

$$\mathsf{next}_b(aaabb) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

We can then consider $\mathsf{ustat}_2(w_n)[x_1, x_2]$ where $x_1, x_2 \in \Sigma$ as the probability that a uniform random factor u of size 2 is such that $u_1 = x_1 \wedge u_2 = x_2$. We write:

$$\mathsf{ustat}_2[x_1, x_2] = \mathbb{P}rob_u[u_1 = x_1 \wedge u_2 = x_2].$$

We then use the notation:

$$\mathsf{ustat}_2[x_2 \mid x_1 = a] = \mathbb{P}rob_u[u_2 = x_2 \mid u_1 = x_1 = a] = \mathsf{next}_a.$$

We use conditional probabilities as projections. Similarly, for a distribution of arity r with variables $x_1 ... x_r$. Consider the formula:

$$\psi_2(x_1, x_2 \in \Sigma_m) : \exists y_1, y_2 \in \{1, 2...n\} \;\; P(x_1, y_1) \wedge y_2 = y_1 + 1 \wedge P(x_2, y_2)$$

The formula $\psi(x_1)$ is separating for the domain Σ_m, because of the functional dependency $y_1, y_2 \to x_1, x_2$: there is only one letter on each position. We can replace

the existential quantifier $\exists y_1, y_2$ by the counting $\#y_1$ and the distribution $\%y_1$ quantifiers and obtain the formulas $\psi_c(x_1, x_2)$ and $\psi_d(x_1, x_2)$. It defines precisely the distribution $\mathsf{ustat}_2[x_1, x_2]$.

If the size m of the alphabet is large, the number of potential k-factors is m^k, i.e. exponential in k. The distribution next is sparse in general but its space representation is too large, although it can be compressed. A neural network or a transformer constructs a compressed version [21] of a distribution next', related to next, of size $(p \cdot k)^2 \cdot d$ which approximates this distribution, where p is the size of the embedding of the tokens and d the depth of the circuit. We have d layers where each layer is a matrix of size $p \cdot k$. A node at layer i applies a non linear function to a linear combination of the values of the nodes at the previous layer $i-1$, for $i > 1$. This distribution is central for Large Language Models and generative A.I. in general.

3.2 Graphs

A finite graph is a structure $G_n = (\{1, 2...n\}, E)$, where $E \subseteq \{1, 2...n\} \times \{1, 2...n\}$ is the Edge relation. As in the previous case, we extend the structure with the rationals and obtain:

$$G_n = (\{1, 2...n\}, E, ; \mathbb{Q}, +, *, /)$$

Let $d(x, i)$ the relation expressing that a node x has i distinct neighbors. In this case, we have the functional dependency $x \rightarrow i$ and we can define the natural degree distributions. The degree function degree defines the number of nodes of a degree i, i.e. for $i, j \in \{1, 2...n\}$, $\mathsf{degree}(i) = j$ if exactly j nodes have degree i. The degree distribution is:

$$\mathsf{degree}_d(i) = \mathsf{degree}(i)/n$$

which gives the probability that a random uniform node has degree i.

We can use the Logic FO+ C (Counting). We extend the first-order quantifiers with new quantifiers $\exists^i, \exists^{>i}, \exists^{<i}$, which quantify the existence of exactly i, more than i, less than i witnesses. Define:

$$d(x, i) : \exists^i y \; E(x, y)$$

$$degree(i) : \#x \; d(x, i)$$

$$degree_d(i) : \%x \; d(x, i)$$

This last formula formula defines the degree distribution of the graph because of the functional dependency $x \rightarrow i$. One can show that we precisely need this extension of First order Logic. An important statistical constraint is that the degree' distribution follows a Zipf law of parameter $\alpha > 1$:

$$\mathsf{degree}_d(i) \simeq c/i^\alpha$$

The $\simeq$ symbol refers to some distance, between two distributions, discussed in Section 3.4.

This last formula formula gives the degree distribution of the graph because if the functional dependency $x \rightarrow i$. If we express that this degree distribution is ε-close to a Zipf distribution for the Fréchet distance introduced in section 3.4, we write:

$$\%x \;\; d(x,i) \;\simeq_{F,\varepsilon}\; c/i^2 \;\text{ for }\; \mathsf{dist}_F(\%x \;\; d(x,i), c/i^2) < \varepsilon$$

In Section 4, we consider the constraint $\%x \;\; d(x,i) = \; c/i^2$, i.e the graphs follow a power law degree distribution of parameter $\alpha = 2$.

3.3 Relational Databases and Datawarehouses

Consider the two tables of Figures 1, 2. The *Product* relation lists different products with a key PID, and the *Buy* relation lists the Sales of the products with a date and a price. The *Buy* relation is often called a *Datawarehouse*, as it may grow to a very large table and there are functional dependencies between the attributes of the *Buy* relation and some attributes of other tables, called the *analysis attributes*. An OLAP schema defines all the functional dependencies between the attributes of the Datawarehouse relation and the analysis attributes. In this example: $PID, DATE, PRICE \rightarrow TYPE, AGE$ and the two attributes $TYPE, AGE$ are analysis variables.

Product

PID	TYPE	AGE
P1	A	O
P2	A	N
P3	B	O
P4	C	N

Fig. 1 Table Product

Buy

PID	DATE	PRICE
P1	Jan 1.	15
P2	Jan 2.	30
P1	Jan 2.	20
P3	Jan 3.	45
P4	Jan 5.	30
P1	Jan 6.	10

Fig. 2 Table Buy

It is natural to analyze the number of Sales by TYPE, as a unary distribution Q_1^d, or the number of Sales by TYPE and AGE as a binary distribution Q_2^d. In both cases, we count the number of tuples of the relation *Buy* for different values of the analysis variables, called *dimensions* in the OLAP terminology. It is also natural to analyze the global Sales, i.e the Sum of the PRICE attribute with the same dimensions. They correspond to the Count or Sum Aggregation operators in SQL, followed by the GROUP BY construction. We can define these statistical queries with simple First Order Formulas in the relational language $Product(x,t,a), Buy(x,y,z)$.

- Q_1^d: number of sales per TYPE.

$$Q_1^d(t) : \%(x,y,z)\exists a \;\; Product(x,t,a) \wedge Buy(x,y,z)$$

- Q_2^d number of sales per TYPE and AGE.

$$Q_1^d(t,a) : \%(x,y,z)\ \ Product(x,t,a) \wedge Buy(x,y,z)$$

Both distributions are well defined because $x, y, z \rightarrow t, a$. For the sum of Sales, we sum on the PRICE attribute and write for the first query:

$$Q_1^d(t) : \%Sum(x,y,z).z\ \exists a\ \ Product(x,t,a) \wedge Buy(x,y,z)$$

A possible Statistical constraint is to fix one of these distributions. For example, the number of Sales by AGE is close to: δ_0: (O: $2/3$, N: $1/3$). It is true for the instance of Figures 1, 2. If the distributions are close for the L_1 distance, we write:

$$\%(x,y,z)\exists t\ \ Products(x,t,a) \wedge Buy(x,y,z) \simeq_{1,\varepsilon} (O : 2/3, N : 1/3)$$

Statistical constraints are similar to the distributions introduced in the Probablistic Relational models of [12]. We now introduce in Section 3.4 a central notion: the distance between distributions.

3.4 Distances between distributions

Given two distributions δ_1, δ_2 on the same domain, there are many possible distances. Classical distances include L_p distances, EMD (Earth-Moving Distance), Fréchet and other pseudo distances such as KL (Kullback-Liebler).

The L_1 distance, also called the *variational distance* between two distributions δ_1, δ_2 on a domain with n elements is defined as:

$$\mathsf{dist}_1(\delta_1, \delta_2) = \frac{1}{2} \cdot \sum_i \mid \delta_1(i) - \delta_2(i) \mid .$$

If we relabel the domain with a permutation π we may have a smaller variation $\sum_i \mid \delta_1(i) - \delta_2(\pi(i)) \mid$ and [10] introduces the *Variation distance up to relabeling* $\mathsf{VDR}(\delta_1, \delta_2)$[1] as the minimum over π of $\frac{1}{2} \cdot \sum_i \mid \delta_1(i) - \delta_2(\pi(i)) \mid$. It is also the L_1 distance between the frequencies of the distributions, i.e. ordered by decreasing values. Notice, that the distribution of the frequencies is invariant by relabeling, hence used for the definition of statistical constraints.

The *Fréchet* distance considers the distributions as points x, y in two dimensions and defines the *Fréchet distance* as the minimum d such that for each point (x, y) of δ_i there is a point of δ_j at an Euclidian distance less than d. For the *relative Fréchet* distance, consider an extended Box $(x \cdot (1 \pm \varepsilon_1), y \cdot (1 \pm \varepsilon_2))$ associated with each point x, y, as in the Figure 3. The *relative Fréchet distance* distance dist_F is the

[1] For a distribution δ, let the histogram h_δ of δ be the function $[0,1] \rightarrow N$ such that $h_\delta(x) = |\{i : \delta(i) = x\}|$, the number ef elements with probability x. Then [10] shows the connection betwen VDR and the Earth-Moving Distance EMD of the histograms: $\mathsf{VDR}(\delta_1, \delta_2) = \mathsf{EMD}(h_{\delta_1}, h_{\delta_2})/2$.

minimum $\varepsilon_1, \varepsilon_2$ such that for each point (x, y) of δ_i there is a point of δ_j in the extended Box $(x \cdot (1 \pm \varepsilon_1), y \cdot (1 \pm \varepsilon_2))$.

We concentrate on two distances L_1 and relative Fréchet. Consider a stream of Graph edges and two inputs determined by the stream of edges up to times t_1 for G_1, and between time t_1 and $t_2 > t_1$ for G_2 (assume $t_2 \simeq 2.t_1$) with approximately the same number of nodes. The two degree distributions δ_1 and δ_2 represented in Figure 3 seem *close* to a power law.

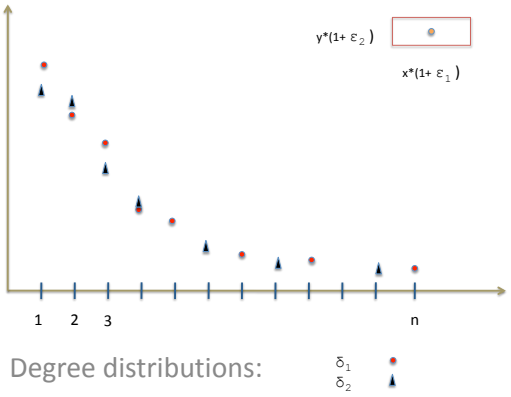

Fig. 3 Degree distributions of G_1 and G_2 on the same stream of Graph edges

The L_1 distance between the degree distributions δ_1 and δ_2 of Figure 3 is however not small because for large degrees i, one of the two distributions has a value 0 depending on i. In this example, the two distributions are close for the relative Fréchet distance because the points x, y of each distributions are relatively close for both x and y, and are also close to a power law. In practice, a statistical law assumes the data is only close to a predefined distribution, as in this example.

The L_1 distance is useful for distributions with small supports and the *relative Fréchet* distance is adapted for large supports of size $O(n)$. These distances are completely different from the Edit distance on the data. Indeed structures of very different sizes are far for the Edit distance but could be close for L_1 and *relative Fréchet*.

4 Stochastic Approximation

In the classical setting, discussed in section 2.2, we approximate an optimization problem such as Maxclique with a possible (ϵ, δ) randomized algorithm A which returns on an input x of size n an integer value in $\{1, 2...n\}$ such that:

$$\forall x \mid x \mid > c \rightarrow \; I\!\!P rob_{\Omega}[\quad \mid A(x) - \mathsf{Maxclique}(x) \mid \leq \varepsilon] \geq 1 - \delta$$

In this definition, the randomized algorithm A has a probabilistic space Ω and guarantees a good answer in the worst-case for large enough inputs $x, i.e. \mid x \mid > c$. We want to relax this last condition and only consider random inputs which satisfy some statistical property. Assume a distribution μ over structures which satisfy some statistical property and a decision problem where the answer is Yes or No.

1-sided-stochastic approximation. For Yes instances we consider the worst-case, but for No instances we only consider random inputs for μ. A *δ-1-sided stochastic randomized* algorithm A for a language L satisfies the following two conditions:

- For all YES instances x, $Prob_{\Omega}[A(x)\ accepts] \geq 1 - \delta$
- For NO instances x drawn from μ, $Prob_{\mu \times \Omega}[A(x)\ rejects] \geq 1 - \delta$

where Ω is the set of possible choices of the algorithm.

2-sided-stochastic approximation. A *δ-2-sided stochastic randomized* algorithm A for a language L satisfies the following two conditions:

- For YES instances x drawn from μ, $Prob_{\mu \times \Omega}[A(x)\ accepts] \geq 1 - \delta$
- For NO instances x drawn from μ, $Prob_{\mu \times \Omega}[A(x)\ rejects] \geq 1 - \delta$

Both definitions depend on μ and we assume that μ is uniform unless it is explicitely specified.

4.1 Large dense subgraphs of graphs which follow a power law degree distribution

Consider a stream of Graph edges (v_i, v_j) and we want to decide if there is a large dense subgraph in the underlying graph G. The approximation of dense subgraphs is well studied in [2] and an $\Omega(n)$ space lower bound is known [1]. A classical density is the ratio $\rho = |E[S]|/|S|$ but we want a much higher density $\gamma = 2.|E[S]|/|S|(|S| - 1)$, hence the expression *very dense*. If $\gamma = 1$, we have a clique, and in practice we look for clusters where $\gamma < 1$.

Definition 1 The (γ, δ)-*large very dense subgraph* problem, where the parameters $\gamma, \delta \leq 1$, takes as input a graph $G = (V, E)$ and decides whether there exists an induced subgraph $S \subseteq V$ such that $|S| > \delta\sqrt{n}$ and $|E[S]| > \gamma|S|(|S| - 1)/2$.

A *very dense subgraph* is also called a γ-clique, as the density is greater than γ. The parameter δ concerns the size of the cluster. The (γ, δ)-*large very dense subgraph* problem is NP-hard and hard to approximate [11], as it contains the maximum clique problem as the special case when $\gamma = 1$. This leads us to use a new notion of approximation, adapted to a specific distribution of inputs. Social graphs define a specific regime where graphs approximately follow a power law degree distribution, precisely the statistical constraint considered in section 3.2.

We proposed in [15] a streaming algorithm which uses $O(\sqrt{n}.\log n)$ space, reads one edge each time and approximates this hard problem on graphs which follow a power law degree distribution. The distribution μ is the uniform distribution on graphs which satisfy this statistical constraint, for each size n.

Theorem 1 *There is a δ-1-sided stochastic randomized streaming algorithm A which uses $O(\sqrt{n}.\log n)$ space for the* (γ, δ)-large very dense subgraph *problem, on inputs which follow a power law degree distribution where the parameters $\gamma, \delta \leq 1$, takes as input a graph $G = (V_n, E)$ and decides whether there exists an induced subgraph $S \subseteq V$ such that $|S| > \delta\sqrt{n}$ and $|E[S]| > \gamma|S|(|S| - 1)/2$.*

The algorithm uses a Reservoir sampling [20] and the analysis relies on the existence of giant components for random graphs generated in a 2-stage process: we first take the configuration model of random graphs which follow a power law degree distribution and then consider an Erdös-Renyi model where edges are uniformly sampled. We then use the Molloy-Reed [18] analysis for the existence of giant components in the Reservoir. If there is a large Connected component in the Reservoir of size $O(\sqrt{n}.\log n)$, the algorithm A accepts, else it rejects.

In this approach, it is important to efficiently decide if some data follows a statistical property. In [16], we give sufficient conditions on the frequency distributions of a streams of elements taken from a set $\{e_1, ...e_n\}$, so that the frequency distributon can be tested Online in space $O(poly(\log n))$, in the sense of a property Tester, using the relative Fréchet distance between distributions.

5 Conclusion

Data have a logical structure with many dependencies which are often expressed in First and Second-order Logic. They also have statistical properties, as defined in this paper by simple relations between statistical queries. Both Logical and Statistical constraints are useful to analyze the data and predict their evolution, but the techniques used are quite different.

We gave the example of graphs which follow a power law degree distribution, as a statistical property. This statistical property is definable in FO+ Counting. On these graphs, a generalization of Maxclique, definable in MSO, becomes easy with a δ-1-sided stochastic approximation. In this framework, both logical and statistical constraints can be combined to solve hard problems in the worst case.

Acknowledgment: I thank Richard Lassaigne for fruitful discussions on the notions of statistical constraints.

References

1. Bahman Bahmani, Ravi Kumar, and Sergei Vassilvitskii. Densest subgraph in streaming and mapreduce. *Proc. VLDB Endow.*, 5(5):454–465, January 2012.
2. Sayan Bhattacharya, Monika Henzinger, Danupon Nanongkai, and Charalampos E. Tsourakakis. Space- and time-efficient algorithm for maintaining dense subgraphs on one-pass dynamic streams. *CoRR*, abs/1504.02268, 2015.
3. Armin Biere, Mathias Fleury, Nils Froleyks, and J.H. Marijn Heule. The sat museum. In *Proceedings of the 14th International Workshop on Pragmatics of SAT (SAT 2003)*, volume 3545 of *CEUR Workshop Proceedings*, pages 72–87, 2023.
4. C. Blatter and E. Specker. Modular periodicity of combinatorial sequences. *Abstracts Am. Math. Soc.*, 4, 1983.
5. Andrej Bogdanov and Luca Trevisan. Average-case complexity, 2021.
6. Ashok K. Chandra, Harry R. Lewis, and Johann A. Makowsky. Embedded implicational dependencies and their inference problem. In *Proceedings of the Thirteenth Annual ACM Symposium on Theory of Computing*, STOC, pages 342–354. Association for Computing Machinery, 1981.
7. B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 9(4):125–150, 2000.
8. E. Fischer. The specker–blatter theorem does not hold for quaternary relations. *Journal of Combinatorial Theory, Series A*, 103(1):121–136, 2003.
9. Philip Gage. A new algorithm for data compression. *C Users J.*, 12(2):23–38, 1994.
10. Oded Goldreich and Dana Ron. *On the Relation Between the Relative Earth Mover Distance and the Variation Distance (an Exposition)*, pages 141–151. Springer International Publishing, 2020.
11. J. Hastad. Clique is hard to approximate within $n^{1-\epsilon}$. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, FOCS '96, pages 627–. IEEE Computer Society, 1996.
12. Daphne Koller. Probabilistic relational models. In Sašo Džeroski and Peter Flach, editors, *Inductive Logic Programming*, pages 3–13. Springer Berlin Heidelberg, 1999.
13. Tomer Kotek and Johann A. Makowsky. *Definability of Combinatorial Functions and Their Linear Recurrence Relations*, pages 444–462. Springer Berlin Heidelberg, 2010.
14. Leonid Levin. Average case complete problems. *SIAM Journal on Computing*, 15(1):285–286, 1986.
15. Claire Mathieu and Michel de Rougemont. Large very dense subgraphs in a stream of edges. *Network Science*, 9(4):403–424, 2021.
16. Claire Mathieu and Michel de Rougemont. Testing frequency distributions in a stream. *arXiv, 2309.11175*, 2023.
17. Kitty Meeks and Alexander Scott. The parameterised complexity of list problems on graphs of bounded treewidth. *Information and Computation*, 251:91–103, 2016.
18. Michael Molloy and Bruce Reed. The size of the giant component of a random graph with a given degree sequence. *Comb. Probab. Comput.*, 7(3):295–305, September 1998.
19. Tim Roughgarden. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2021.
20. Jeffrey S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, March 1985.
21. Yibo Yang, Stephan Mandt, and Lucas Theis. An introduction to neural data compression. *Foundations and Trends in Computer Graphics and Vision*, 15(2):113–200, 2023.