On average case complexity

Rainer Schuler
Universität Ulm
Abteilung Theoretische Informatik

1 Introduction

The aim of this talk is to give an introduction to the notion of Levin's average case complexity and then show some of the fields were recent research in this area is focused on. The first part is motivated by the struggle to find a precise and generally accepted definition of what is an efficient time on the average algorithm, given some distribution on the input. The definition should be easy (to use) and of course be machine independent and should posses properties like being closed under composition of algorithms. A reduction of a problem A to another problem which is efficiently solvable on average should give an efficient on average procedure to solve A. We then look in more detail at DistNP the class of problems in NP with polynomial time distributions on the inputs. This class has complete and self-reducible problems.

2 Distributional complexity classes

In the following let $\mu' : \Sigma^* \to [0,1]$ be a density function, μ, $\mu(x) = \sum_{y \leq x} \mu'(y)$ be the corresponding distribution, and $\mu'_n(x) = \mu'(x) / \sum_{|y|=n} \mu'(y)$ be the conditional probability of an input x of length n. For simplicity we only require that $\sum_x \mu'(x) = c$. for some constant c.
In this context a problem is always a pair, a decision problem together with a distribution function. For example the above mentioned class DistNP contains all pairs (D, μ), where $D \in$ NP and μ is a polynomial time computable distribution function. One stimulating question is (was) whether all or part of DistNP can be solved efficiently on the average. It is convenient to restrict the domain (the inputs) of a problem to a subset of Σ^* which evloves by coding the instances of the problem. Then it is possible to define the length of a formula and its probability independent of the chosen coding technique [BG90, BG91]. So we can define a distributional version of 3-SAT:

The set of instances (syntactic correct 3-SAT formulas), a formula F_n consists of n Literals from a variable set $\{v_1, v_2, \ldots, v_n\}$. Each clause, except possibly the last, contains 3 Literals, i.e. $l_{3i}, l_{3i+1}, l_{3i+2}$.

The length function $|F_n| = n$.

The probability distribution μ with $\mu'(F_n) = \frac{1}{n^2}(2n)^{-n}$.

Note that there exist $(2n)$ different possibilities to select a Literal. Thus to produce instances according to μ one has to choose a length n with probability $\frac{1}{n^2}$ and then n Literals from $\{v_1, v_2, \ldots, v_n, \overline{v}_1, \overline{v}_2, \ldots, \overline{v}_n\}$, each with probability $\frac{1}{2n}$.
Another problem, which is complete for the class DistNP (as we will see later), is a distributional halting problem [Gur87, Gol88]:

The set of instances, an input $X_{m,n,t}$ is a 3-tuple $\langle M, x, 0^t \rangle$ consisting of a description M of a nondeterministic machine of size m, an input x, $|x| = n$. (for M) and a number t of steps (to simulate M).

The length function $|X_{m,n,t}| = b(m,n,t)$. (Here $b(m,n,t)$ denotes a polynomial bijection from a tuple of natural numbers to natural numbers).

The probability distribution μ with $\mu'(X_{m,n,t}) = \frac{c}{b^2(m,n,t)} 2^{-(m+n)}$.

Again, for the distribution, one has to choose a length proportional to the inverse quadratic and then uniformly an input in the chosen length. Now lets get back to the notion of polynomial on the average. A first intuitive definition could require that the expected value of a function is bounded by a polynomial, i.e.:

$$\exists k \forall n : \sum_{|x|=n} \mu'_n(x) \cdot f(x) \leq O(n^k).$$

This definition has the disadvantage that there exist functions f and distributions μ such that f is average polynomial with respect to μ but f^2 is not. One easily verified example is

$$f(x) = \begin{cases} 2^n, \text{ if } x = 0^n \\ n, \text{ otherwise} \end{cases} \quad \text{and } \mu'_n(x) = 2^{-n}.$$

This anomality (and others) is solved by a defintion from Levin. (One other problem noted in [Gol88] is, that even if a problem is solvable in polynomial time on average with some oracle, which is itself in P, then a procedure combining both algorithms is not in average polnomial time).

Definition 2.1 *[Lev84, Lev86] A function $f : \Sigma^* \to R^+$ is polynomial on average with respect to a distribution μ (polynomial on μ-average) if $\exists \delta > 0$ such that*

$$\sum_{x \in \Sigma^*} \mu'(x) \cdot \frac{f(x)^\delta}{|x|} \leq \infty.$$

We suggest here to use another definition which is from Schapire [Sch90] (see also [SY92]). For a distribution μ, let $\text{Prob}_\mu[T(x)]$ denote the probability for event T, where x is chosen randomly according to μ.

Definition 2.2 *A function f is p on μ-average if and only if :*

$$\forall m > 0 : \ \mathrm{Prob}_\mu[f(x) > p(|x| \cdot m)] \ < \ 1/m.$$

If p has to be a polynomial we get Levin's definition of polynomial on average. We give the proof from Schapire.
Assume there exists a $\delta > 0$ and a number N such that

$$\sum_{x \in \Sigma^*} \mu'(x) \cdot \frac{f(x)^\delta}{|x|} \leq N.$$

Then by the Markov inequality

$$\mathrm{Prob}_\mu \left[\frac{f(x)^\delta}{|x|} > N \cdot m \right] < 1/m, \text{ and then}$$

$$\mathrm{Prob}_\mu[f(x) > (N \cdot m \cdot |x|)^{1/\delta}] < 1/m.$$

Thus f is bounded by $p(|x| \cdot m) = (N \cdot |x| \cdot m)^{1/\delta}$.
On the other hand, let, for some constant $k > 0$, f be bounded on the average by $p(|x| \cdot m) = (k \cdot m \cdot |x|)^{k/2}$. Then we get for all $m > 0$:

$$\begin{aligned} \frac{1}{m} \ &> \ \mathrm{Prob}_\mu \left[f(x) > (k \cdot m \cdot |x|)^{k/2} \right] = \mathrm{Prob}_\mu \left[f(x)^{1/k} > (k \cdot m \cdot |x|)^{1/2} \right] \\ &\geq \ \mathrm{Prob}_\mu \left[f(x)^{1/k} > k \cdot |x| \cdot m^{1/2} \right] = \mathrm{Prob}_\mu \left[\frac{f(x)^{1/k}}{|x|} > k \cdot m^{1/2} \right] \end{aligned}$$

Now let $t = k \cdot m^{1/2}$, then $\mathrm{Prob}_\mu[f(x)^{1/k}/|x| > t] \leq 1/m = k^2/t^2$, and therefore:

$$\begin{aligned} \sum_{x \in \Sigma^*} \mu'(x) \cdot \frac{f(x)^{1/k}}{|x|} \ &\leq \ \sum_{t \geq 1} \mathrm{Prob}_\mu \left[t - 1 < \frac{f(x)^{1/k}}{|x|} \leq t \right] \cdot t \\ &= \ \sum_{t \geq 0} \mathrm{Prob}_\mu \left[\frac{f(x)^{1/k}}{|x|} > t \right] \\ &\leq \ 1 + k^2 \cdot \sum_{t \geq 1} \frac{1}{t^2} < \infty. \end{aligned}$$

The class of problems solvable in average polynomial time is called AP. Now which problems are solvable in polynomial time? Recall that here a problem is always a decision problem and a distribution on the inputs. Many worst-case NP-complete problems together with uniform distributions on inputs of the same length have been shown to be in AP. Lets look at the above defined 3-SAT problem. Assume we don't have 3-SAT but 3-$\frac{1}{12}$-SAT, i.e. the literals for F_n are chosen from only $n^{\frac{1}{12}}$ many variables. Then a brut force algorithm, which first checks if there are two adjacent clauses, one clause contains a Literal l_i three times and the other contains three times $\overline{l}_i$ (and are hence unsatisfiable), is in AP.

First we give an estimation of the probability that these clauses exist. For a fixed pair of clauses the probability is $n^{\frac{1}{12}} \cdot (\frac{1}{n^{1/12}})^6 > \frac{1}{\sqrt{n}}$. Then the probability that no two adjacent clauses contain contrary literals three times is smaller than $(1 - \frac{1}{\sqrt{n}})^{n/6} < 2^{-\sqrt{n}/6}$. On this portion of the inputs, of each length, our algorithm has to make a brut force search for a satisfying assignment. This search takes less than $2^{\sqrt{n}}$ time. This gives us the following estimation on the probability that the algorithm needs more time than $p(|F_n|m) = (k \cdot |F_n| \cdot m)^k$, for some fixed k. For all $m > 0$ and all n:

$$\text{Prob}_{\mu_n}[f(F_n) > (k \cdot n \cdot m)^k] \leq \begin{cases} 0, \text{ if } n \leq k^2 \cdot \log^2 m (\Rightarrow 2^{\sqrt{n}} \leq m^k) \\ 2^{-\sqrt{n}/6}, \text{ otherwise.} \end{cases}$$

Thus the overall probability to exeed the allowed time is:

$$\begin{aligned} \text{Prob}_{\mu}[f(x) > (k \cdot |F_n| \cdot m)^k] &\leq \sum_{n > k^2 \cdot \log^2 m} \frac{1}{n^2} 2^{-\sqrt{n}/6} \\ &= \sum_{n>0} \frac{1}{(n + k^2 \log^2 m)^2} 2^{-\sqrt{(n+k^2 \log^2 m)}/6} \\ &\leq \sum_{n>0} \frac{1}{n^2} 2^{-k \log m/6} \\ &= \frac{1}{m^{k/6}} \sum_{n>0} \frac{1}{n^2} \\ &< \frac{1}{m} \end{aligned}$$

3 Reducibility

Lets look at Levin's many one reducibility first. A reduction function should not be allowed to reduce elements which are likely to occur in one distributional problem to elements which are rare in the other. Otherwise an efficient algorithm for the second problem might not give a efficient solution to the first.

Definition 3.1 *Let (D_1, μ_1) and (D_2, μ_2) be distributional problems. Then, (D_1, μ_1) is* polynomial time many one reducible *to (D_2, μ_2), denoted by $(D_1, \mu_1) \leq_m^p (D_2, \mu_2)$. if there exists a reduction function f on Σ^* which satisfies the following three conditions:*

1. *f is computable in time polynomial(on μ_1-average).*
2. *For all $x \in \Sigma^*$: $x \in D_1 \Leftrightarrow f(x) \in D_2$.*
3. *There exists a constant c such that for all $y \in \Sigma^*$:*

$$\mu_2(y) \geq \sum_{x:f(x)=y} \frac{\mu_1(x)}{|x|^c + c}.$$

Condition (3) ensures that the probability $\mu_2(y)$ of a string y dominates the sum of the probabilities $\mu_1(x)$ of the strings x which are reduced by f to y. We say μ_2 (polynomially) dominates $\mu_1(f^{-1})$.
The class AP is closed under polynomial time many–one reductions (a proof can be found in Goldreich [Gol88]) and the above definition of reducibility is transitive. Note that, to get transitivity, substituting (3) by (3') defined below is not sufficient. (Hint: To find a counter example consider two reductions where the second one is not honest.

(3') There exists a constant c such that for all $y \in \Sigma^*$:

$$\mu_2(y) \geq \frac{1}{|y|^c + c} \cdot \sum_{x:f(x)=y} \mu_1(x).$$

Theorem 3.2 *The bounded halting problem is complete for* DistNP.

Proof:

■

We will now give two examples of reductions. Since selfreducibility is one important notion in complexity theory, we will show that variations of the above defined bounded halting problem and 3-SAT are in fact selfreducible.
Let prefix-BHP, the bounded halting problem, were some initial steps of the computation are already fixed, be defined as follows:

The set of instances, an input $X_{m,n,t,l}$ is a 4-tuple $\langle M, x, 0^t, r\rangle$ consisting of a description M of a nondeterministic machine of size m, an input x, $|x| = n$ (for M), a number t of steps (to simulate M), and a string $r \in \{0,1\}^l$ fixing the first l nondeterministic choices of M.

The length function $|X_{m,n,t,l}| = b(m,n,t,t-l)$. (Here we assume that $0 < l < t$).

The probability distribution μ with $\mu'(X_{m,n,t,l}) = \frac{c}{b^2(m,n,t,l)} 2^{-(m+n+l)}$.

Similarily, one can define a prefix-3-SAT:

The set of instances, a formula $F_{n,l}$ consists of n Literals from a set of variable $\{v_1, v_2, \ldots, v_n\}$ and a string $r \in \{0,1\}^l$ giving a truth assignment to the first l variables.

The length function $|F_{n,l}| = b(n, n-l)$. (Here we assume that $0 < l < t$).

The probability distribution μ with $\mu'(F_{n,l}) = \frac{c}{b(n,n-l)^2}(2n)^{-(n+l)}$.

Theorem 3.3

(i) *BHP* $\leq_m^p$ *prefix-BHP and 3-SAT* $\leq_m^p$ *prefix-3-SAT.*

(ii) *Prefix-BHP and prefix-3-SAT are selfreducible.*

Proof: (i) Let f reduce BHP to prefix-BHP: $f(\langle M, x, 0^t\rangle) = \langle M, x, 0^t, \lambda\rangle$, then f is polynomial time and $\langle M, x, 0^t\rangle \in$ BHP iff $\langle M, x, 0^t, \lambda\rangle \in$ prefix-BHP. Since the reduction is one to one, it remains to verify that for some constant c:

$$\frac{1}{b^2(m,n,t,t)}2^{-(m+n)} \geq \frac{1}{b^{2c}(m,n,t)+c} \cdot \frac{1}{b^2(m,n,t)}2^{-(m+n)}.$$

Similar let g reduce 3-SAT to prefix-3-SAT: $g(F_n) = F_{n,\lambda}$. Again g is polynomial time and $F_n \in$ 3-SAT iff $F_{n,\lambda} \in$ prefix-3-SAT. The reduction g is also one to one and therefor for a constant c:

$$\frac{1}{b(n,n-1)^2}(2n)^{-(n+1)} \geq \frac{1}{(n^{2c}} \cdot \frac{1}{n^2}(2n)^{-n}.$$

(ii) To reduce instances of prefix-BHP to smaller instances note that for $a \in \{0.1\}$:

$$|\langle M, x, 0^t, r\rangle| = b(m,n,t,t-l) > b(m,n,t,t-(l+1)) = |\langle M, x, 0^t.ra\rangle|.$$

Consider a reduction which first checks if $l = t$ and the string r describes an accepting computation of M on input x, otherwise it accepts if $\langle M, x, 0^t, r0\rangle$ or $\langle M, x, 0^t, r1\rangle$ is in prefix-BHP. This reduction, accepts prefix-BHP, and reduces instances of probability $\frac{1}{b^2(m,n,t,t-l)}2^{-(m+n+l)}$ to smaller instances of probability $\frac{1}{b^2(m,n,t,t-(l+1))}2^{-(m+n+l+1)}$. This probability is only polynomially smaller.
The selfreducibility of prefix-3-SAT is proved similar. ∎

From the above two theorems it follows immedeatly that prefix-BHP, a selfreduclible problem, is complete for DistNP.

References

[BG90] A. Blaas and Y. Gurevich. On the reduction theory for average-case complexity. *Proc. 4th Workshop on Computer Science Logic*, pages 17–30. 1990.

[BG91] A. Blaas and Y. Gurevich. Randomizing reductions of search problems. *Proc. 11th Foundations of Software Technology and Theoretical Computer Science*, pages 10–24, 1991.

[Gol88] O. Goldreich. Towards a theory of average case complexity. Technical Report TR-507, Computer Science Dept., Technion, Haifa, Israel, 1988.

[Gur87] Y. Gurevich. Complete and incomplete randomized NP problems. *Proc. 28th IEEE Symposium on Foundations of Computer Science*, pages 111–117, 1987.

[Lev84] L. Levin. Problems, complete in "average" instance. *Proc. 16th ACM Symposium on Theory of Computing*, page 465, 1984.

[Lev86] L. Levin. Average case complete problems. *SIAM Journal on Computing*, 15:285–286, 1986.

[Sch90] R.E. Schapire. The emerging theory of average-case complexity. Technical Report TM-431, Massachusetts Institut of Technology, 1990.

[SY92] R. Schuler and T. Yamakami. Structural average case complexity. *Proc. 12th Foundations of Software Technology and Theoretical Computer Science*, pages 128–139, 1992.