An Overview of Information-Based Complexity

Arthur G. Werschulz
Department of Computer and Information Sciences
Fordham University, New York, NY 10023
Department of Computer Science
Columbia University, New York, NY 10027

Technical Report CUCS-022-02

October 17, 2002

Abstract

Computational complexity has two goals: finding the inherent cost of some problem, and finding optimal algorithms for solving this problem. *Information-based complexity* (IBC) studies the complexity of problems for which the available information is partial, contaminated by error, and priced. Examples of such problems include integration, approximation, ordinary and partial differential equations, integral equations, and the solution of nonlinear problems such as root-finding and optimization. In this talk, we give a brief overview of IBC. We focus mainly on the integration problem (which is a simple, yet important, problem that can be used to illustrate the main ideas of IBC) and the approximation problem (which will be of most interest to specialists in learning theory). One important issue that we discuss is the "curse of dimension"—the depressing fact that the worst case complexity of many problems depends exponentially on dimension, rendering them intractable. We explore IBC-based techniques for vanquishing the curse of dimension. In particular, we find that randomization beats intractability for the integration problem but not for the approximation problem; on the other hand, both these problems are tractable in the average case setting under a Wiener sheet measure.

1 Introduction

Computational complexity is a measure of the intrinsic computational resources required to solve a problem. In addition, those who study the computational complexity of a problem typically are interested in optimal algorithms, i.e., algorithms consuming the minimal resources necessary to solve the problem. An example is given by the sorting problem in a decision-tree model of computation. For a given $n > 0$, we want to know the complexity (minimal number of decisions) of sorting a set of

n elements chosen from a fixed totally-ordered set. We also want to find an optimal algorithm, i.e., one that sorts any set of size n using a minimal (or nearly-minimal) number of comparisons. It is well-known that if we use a worst case setting to measure the cost of an algorithm, then the complexity is proportional to $n \log n$, and mergesort and heapsort are nearly-optimal.

Note that we have complete information for the sorting problem; that is, we can completely specify any particular instance of the sorting problem using a finite amount of resources (e.g., bits). This is because of the inherently-finite nature of the problem. But (as we shall see), many problems require infinitely-many degrees of freedom for their complete specification. Hence using a finite a finite amount of resources, what we can know about any particular instance of such a problem is necessarily incomplete.

Information-based complexity (IBC) studies the complexity of problems for which we don't have complete information about any problem instance. More precisely, IBC studies the complexity of problems for which the available information is

- *partial*, i.e., the available information doesn't uniquely identify the problem instance,
- *contaminated by error*, such as roundoff or measurement error, and
- *priced*, so that each information measurement has a cost.

Because of space limitations in this article, we omit proofs and bibliographic references. The interested reader should consult the Bibliography of this paper (especially [5]) for citations and fuller discussion.

2 An Example: The Integration Problem

We can illustrate the basic concepts of IBC via a simple example, univariate integration. Suppose we want to compute integrals

$$S(f) = \int_0^1 f(x)\,dx \qquad \forall f \in F.$$

Here, F is a given class of integrands. Unless we are very lucky, the class F will generally include functions that don't have closed-form antiderivatives, so we won't be able to apply the fundamental theorem of the calculus.

What do we know about any $f \in F$ when trying to compute $S(f)$? Typically, we only have the *information*

$$N(f) = [f(t_1), \ldots, f(t_n)].$$

Here, the sample points $t_1, \ldots, t_n \in [0.1]$ may be chosen either *adaptively* (i.e., each sample point can depend on the previously-chosen sample points, so that the sample points depend on f) or *non-adaptively* (the same sample points can be used for all f). Clearly, this information is partial (unless F is finite-dimensional). Moreover, roundoff or measurement error can contaminate the information values. Finally, there is a cost for obtaining each information value; depending on the nature of the class F, this cost might be the cost of running a subroutine or the cost of running an experiment.

For the sake of concreteness, let us suppose that F is the class of all functions $f\colon [0,1] \to \mathbb{R}$ satisfying the Lipschitz condition

$$|f(\xi) - f(\eta)| \le |\xi - \eta| \qquad \forall\, \xi, \eta \in [0,1].$$

(We refer to this knowledge as *global information*, as opposed to the *local information* $N(f)$ about any particular $f \in F$.) We then construct an approximation

$$S(f) \approx U(f) = \phi\big(N(f)\big).$$

The quality of such an approximation is given by its *error*

$$e(U) = \sup_{f \in F} |S(f) - U(f)|.$$

Note that this is a *worst case* error measure; other definitions (such as average, probabilistic, and randomized) can be used as well.

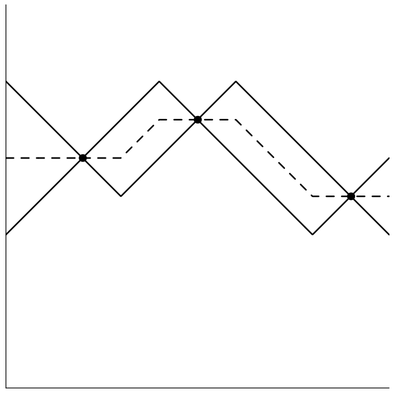

Figure 1: Functions sharing the same information

Note that the available information $N(f)$ does not uniquely characterize any $f \in F$, see Figure 1. This leads to a powerful *adversary principle*, namely, that $U(f)$ must approximate $S(\tilde{f})$ for any $\tilde{f} \in F$ satisfying $N(\tilde{f}) = N(f)$. More precisely, let us define the *radius of information* as

$$r(N) = \sup_{y \in N(F)} \operatorname{rad} S\big(N^{-1}(y)\big),$$

with rad denoting the radius of a set. We then have

$$r(N) = \inf_{\phi \text{ using } N} e(\phi, N).$$

That is, the radius of information is a sharp lower bound on the errors of all algorithms using said information.

Let $f_{\rm lower}$ and $f_{\rm upper}$ be the lower and upper functions of the envelope of functions sharing the same information as f, and let $f_{\rm mid} = \frac{1}{2}(f_{\rm lower} + f_{\rm upper})$ be the center of this envelope. Defining $\phi^*\big(N(f)\big) = S(f_{\rm mid})$, we have

$$e(\phi^*, N) = r(N) = \sup_{\substack{h \in F \\ h(t_1)=\cdots=h(t_n)=0}} S(h).$$

That is, this central algorithm is an optimal error *algorithm.*

Using the second equality in the previous equation, we find that

$$r(N) = \tfrac{1}{2}t_1^2 + \tfrac{1}{4}\sum_{i=1}^{n-1}(t_{i+1} - t_i)^2 + \tfrac{1}{2}(1 - t_n)^2.$$

Moreover, if $N(f) = [y_1, \ldots, y_n]$, our optimal error algorithm takes the form

$$\phi^*\big(N(f)\big) = y_1 t_1 + \sum_{i=1}^{n-1} \tfrac{1}{2}(y_i + y_{i+1})(t_{i+1} - t_i) + y_n(1 - t_n).$$

That is, the optimal error algorithm is a linear *algorithm.*

We previously mentioned that information can be adaptive or non-adaptive. Clearly, non-adaptive information is simpler than adaptive information. Moreover, non-adaptive information can be evaluated in parallel. One might ask whether adaptive information has any advantage over non-adaptive information. The answer is "no." *For any adaptive information $N^{\rm a}$, there exists non-adaptive information $N^{\rm non}$ such that*

$$\operatorname{card} N^{\rm non} \le \operatorname{card} N^{\rm a} \qquad \textit{and} \qquad r(N^{\rm non}) \le r(N^{\rm a}).$$

Here, $\operatorname{card} N$ is the maximal number of information evaluations used by $N(f)$ over all $f \in F$.

We now know how to choose an algorithm that best makes use of given information. What is the best way of choosing the information? That is, we want to find

$$r(n) = \inf_{\operatorname{card} N \le n} r(N),$$

the nth minimal radius of information. Then

$$r(n) = \frac{1}{4n},$$

which is achieved by the nth *optimal information*

$$N_n(f) = [f(t_1^*), \ldots, f(t_n^*)]$$

where

$$t_i^* = \frac{2i-1}{2n} \qquad (1 \le i \le n).$$

Hence

$$U_n(f) \equiv \phi^*\big(N_n(f)\big) = \frac{1}{n}\sum_{i=1}^{n} f(t_i^*)$$

is an nth *minimal error algorithm.* Note that this is a simple midpoint quadrature rule.

So far, we have only discussed optimality at the error level. If we want to discuss complexity, we need to introduce a model of computation. The classical model assumes that each information evaluation has cost $\mathbf{c}$, and each arithmetic operation has cost 1. For an approximation $U = (\phi, N)$, we let

$$\mathrm{cost}(U) = \sup_{f \in F} \mathrm{cost}\,\big(U(f)\big),$$

where cost $\big(U(f)\big)$ is the cost of sampling the information $y = N(f)$ plus the cost of combining this information to produce $\phi(y)$. That is, we measur cost in a worst case setting. Then for any $\varepsilon > 0$, we define the *ε-complexity* as

$$\mathrm{comp}(\varepsilon) = \inf_{e(U) \le \varepsilon} \mathrm{cost}(U).$$

We get an easy lower bound of the form

$$\mathrm{comp}(\varepsilon) \ge \mathbf{c} \cdot m(\varepsilon),$$

where

$$m(\varepsilon) = \inf\{\, n \ge 0 : r(n) \le \varepsilon \,\}$$

is the *ε-cardinality number*, i.e., the minimal number of evaluations needed to insure that the radius of information is less than ε.

From the formula for the nth minimal radius, we find that

$$m(\varepsilon) = \left\lceil \frac{1}{4\varepsilon} \right\rceil.$$

Since the midpoint rule using the information N_n can be computed using n function evaluations and n arithmetic operations, it follows that

$$\mathbf{c} \left\lceil \frac{1}{4\varepsilon} \right\rceil \le \mathrm{comp}(\varepsilon) \le \mathrm{cost}(U_{m(\varepsilon)} \le (\mathbf{c} + 1) \left\lceil \frac{1}{4\varepsilon} \right\rceil.$$

Since we generally expect $\mathbf{c} \gg 1$, these bounds are very tight.

3 A General Formulation of IBC

Although the analysis in the preceding section dealt with a specific model problem (univariate integration over a Lipschitz class), the basic concepts readily extend to general problems. Such a problem is given by a class F of problem elements, a normed linear space G, and a solution operator $S\colon F \to G$. What we know about any $f \in F$ is given by information operators $N\colon F \to H$, generally of the form

$$N(f) = [\lambda_1(f), \ldots, \lambda_n(f)] \qquad \text{for } \lambda_1, \ldots, \lambda_n \in \Lambda.$$

Here, Λ is a class of permissible information operations. One such choice is Λ^{all}, the class of all continuous linear functionals. If F is a class of functions defined over some domain, we can also choose *standard information* Λ^{std} consisting of function or derivative evaluations. This information can be either *exact* or *noisy*, as well as either *non-adaptive* or *adaptive*.

For such a problem, the goals of IBC are the same as for the integration problem of §2:

- For given information N, determine $r(N)$ and an optimal error algorithm using N.
- For a given nonnegative integer n, determine $r(n)$ and nth minimal error algorithm.
- For given $\varepsilon > 0$, determine $\mathrm{comp}(\varepsilon)$ and optimal complexity algorithm.

Many of the concepts discussed in §2 can be extended in the obvious way for more general problems. Hence we can talk about the radius of information, optimal error algorithms and the optimality of central algorithms, optimal information, minimal error algorithms, cardinality numbers, and complexity, for general problems. Moreover, many of the results of §2 hold for more general *linear* problems (i.e., where F is a balanced convex subset of a linear space and S is linear). For example, nonadaptive and adaptive information are (roughly) equally powerful for all linear problems. Moreover, linear algorithms have optimal error for many linear problems.

4 Problems Studied by IBC

As far as particular applications are concerned, IBC has successfully studied problems in integration, approximation, ordinary and partial differential equations, integral equations, ill-posed problems, roots of nonlinear equations, and nonlinear optimization.

Of all the problems studied by IBC, perhaps the approximation problem is of greatest interest to researchers in learning theory. It fits within the IBC framework as follows. Let $F \subseteq G$, and define the solution operator $S\colon F \to G$ as

$$S(f) = f \qquad \forall f \in F.$$

As a rule, F is a subset of a function space G. This problem has been studied for both Λ^{all} and Λ^{std}. Lack of space prevents us from going into detail here, but we shall mention a few results in the next section.

5 The Curse of Dimension

Define F_r as the class of all r-times differentiable functions $f\colon [0,1]^d \to \mathbb{R}$, subject to the constraint that for any $f \in F_r$, the derivatives of f having total order up through r are bounded by 1. Consider the problems of d-dimensional integration and approximation in the class F_r, with error for the approximation problem being measured in the max-norm. For the integration problem, only Λ^{std} will be permissible information, while for the approximation, either Λ^{std} or Λ^{all} will be permissible. The ε-complexity of both of these problems is proportional to $\mathbf{c}\,(1/\varepsilon)^{d/r}$. (Moreover, Λ^{std} is as powerful as Λ^{all} for the approximation problem.)

Note that complexity of these problems is exponential in d, rather than (say) polynomial. In other words, these problems are *provably* intractable (as opposed to the NP-complete or NP-hard problems of discrete complexity theory, which are only *conjectured* to be intractable). We say that such problems suffer from the "curse of dimension." Note that high-dimensional problems (with d in the hundreds or thousands) occur in many areas (including science, economics, and finance). Many of these high-dimensional problems are afflicted with the curse of dimension.

Moreover, the curse's shadow even casts a pall over problems of moderate dimension. For example, suppose that $\varepsilon = 10^{-6}$. Ignoring proportionality constants, 10^{18} arithmetic operations will be necessary to solve a problem with $d = 3$ and $r = 1$. If we (generously) assume 10^{10} operations per second, solving this problem for this error threshold will take more than three years!

How can we vanquish this curse? Certainly, choosing a better algorithm won't help, since we're talking about the complexity of the problem, i.e., the minimal cost of solving the problem over all algorithms.

One idea is to weaken the worst case assurance. The *randomized setting* allows randomized approximations $U_\tau(f)$ to the solution $S(f)$, where τ is chosen at random from a probability space. The most famous randomized algorithm is Monte Carlo quadrature. It is known that the randomized error of n-point Monte Carlo for integration over F_r is $1/\sqrt{n}$, and so cost of using Monte Carlo to get an ε-approximation of integration over F_r is at most $(\mathbf{c} + 1)\lceil 1/\varepsilon \rceil^2$. Since this bound is independent of d, Monte Carlo beats intractability for integration.

Since randomization beats intractability for integration, one might hope that the same holds for the approximation problem over F_r. Unfortunately, this is not the case; it is known that the randomized and worst case ε-complexities for approximation are about the same, and so randomization does not break intractability for approximation. *It is an open problem to determine for which problems randomization breaks intractability.*

The *average case setting* is another setting we can use in our attempts to break intractability. Here, the error of an approximation is given by its expectation with respect to a probability measure on the class of problem elements. To be specific, let F_{w} be the class of continuous functions on I^d, equipped with the Wiener sheet measure. Then the ε-complexity of integration over F_{w} is proportional to $\mathbf{c}\,(1/\varepsilon)(\log 1/\varepsilon)^{(d-1)/2}$, with sampling at reflections of shifted Hammersley points. Moreover, the ε-complexity of approximation over F_{w} is proportional to $\mathbf{c}\,(1/\varepsilon)^2(\log 1/\varepsilon)^{d-1}$. Hence both integration and approximation are tractable in $1/\varepsilon$ in the average case setting for the Wiener sheet measure. Said tractability of approximation will be good news for people working in learning theory.

6 For Further Information

The monograph [5] is an expository overview of IBC, covering both fundamentals and a selection of interesting topics. It also has an extensive bibliography of over 430 items.

Those interested in complete statements and proofs of theorems see [4], as well as [1], [2], and [3]. These books also contain extensive bibliographies.

In addition, there is an IBC website at `http://www.ibc-research.org`.

Among other features, this website provides a searchable bibliographic database at `http://www.ibc-research.org/search-refs.cgi`.

7 Acknowledgments

This research was supported in part by the National Science Foundation under Grant CCR-99-87858.

References

[1] NOVAK, E., *Deterministic and Stochastic Error Bounds in Numerical Analysis* vol. 1349 of *Lecture Notes in Mathematics*, Springer-Verlag New York (1988).

[2] PLASKOTA, L., *Noisy Information and Computational Complexity*, Cambridge University Press Cambridge (1996).

[3] RITTER, K., *Average-Case Analysis of Numerical Problems* vol. 1733 of *Lecture Notes in Mathematics*, Springer-Verlag (2000).

[4] TRAUB, J. F., G. W. WASILKOWSKI, and H. WOŹNIAKOWSKI, *Information-Based Complexity*, Academic Press New York (1988).

[5] TRAUB, J. F., and A. G. WERSCHULZ, *Complexity and Information*, Cambridge University Press Cambridge (1998).