Complexity and Information
Traub Werschulz
(1998)

Part One: Fundamentals

1. Introduction

Computational complexity is a measure of the intrinsic computational resources required to solve a mathematically posed problem. It is an ***invariant*** since the computational complexity depend only on the problem, and is independent of the particular algorithm used to solve it.

This text will consider ***information*** which is ***partial*** and ***contaminated*** which introduces ***intrinsic uncertainty***. ***Global information*** is that which is fixed or assumed for the domain. ***Local information*** is evaluated within a particular domain. Information also has a price.

Information-based complexity (IBC) is the branch of computational complexity that studies problems for which the information is partial, contaminated and priced.

The ***radius*** of information measures the intrinsic uncertainty in the solution due to the available information. We can consider this uncertainty as a worst case setting (which guarantees an answer with error at most ε) or an average case setting (with the weaker guarantee that the expected error is at most ε)

2. Information-Based Complexity

2.1 An Example: Integration. Most functions f(x) arising in practice do not have antiderivatives hence we must seek a numerical solution.

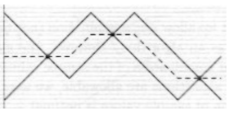
To calculate an approximation to the integral we much choose some sample points t x for which to calculate f(x). This can be done ***adaptively***, where each x is chosen based on the previous values of f(x), or ***nonadaptively***, where the number n and values of x are determined before commencing to calculate f(x). Adaptive information is sequential while non-adaptive information can in principle be evaluated in parallel.

The set of the results at the sample points N(f) is considered local information. It is insufficient to calculate the integral because it only considers the values of f(x) and the sample points t, and anything could happen between these points. We also need global information which restricts the class of f(x).

Suitable global information might be that f(x) has bounded smoothness. That is that f(x) is Lipshitz continuous and that there is an upper bound on the Lipshitz constant.

$$|f(\xi)-f(\eta)|\leq L|\xi-\eta| \quad \forall \xi,\eta\in[0,1]$$

$$Lip(f)= \max_{\xi,\eta\in[0,1]} \frac{|f(\xi)-f(\eta)|}{L|\xi-\eta|}$$



This global information can be used to construct worst case bounds on a linear piecewise construction to estimate the integral of f(x). These functions all share the same information. Furthermore, this provides the best possible estimate of the integral of f(x).

This situation will exist when any continuous function is reduced to discrete numerical values suitable for computation. Hence, the information operator mapping the mathematical input to the computational output is many-one.

2.2 A General Formulation of IBC. Let S:F→G be a solution operator. We wish to compute an approximation to S(f) for an arbitrary problem element fϵF.

The global information is specified by F, which captures our a priori knowledge of the problem elements and often remains fixed. N is an information operator which contains the information that we have about a problem element fϵF. It is obtained by computing N(f) = [L_1(f),..., L_n(f)] where L_1,..., L_n may be chosen adaptively or non-adaptively. n=n(f) may also be chosen adaptively.

An (idealized) algorithm is any mapping Φ: N(F)→G so that we obtain an approximation to S(f) represented by U(f)=Φ(N(f)). The approximation operator U(f) permits analysis at the information level.

The ***model of computation*** state what we are permitted to do and how much it will cost. The cost of computing L(f) is c (independent of f and L). Each of the permissible combinatory or arithmetic operations Ω is performed with unit cost.

The error is e(U,f) - ||S(f) – U(f)|| and the cost is cost(U,f) = cost(N,f) + cost(Φ,N(f))

An ***ε-approximation*** of U requires e(U) ≤ ε. The ***computational complexity*** is the cost(U) of computing an ε-approximation. When these are achieved, the corresponding algorithm Φ and information N are said to be ***optimal***.

There are two goals of IBC:

1. Determine the complexity of a given problem in various settings,
2. Find an optimal (or near optimal) algorithm and information for computing an ε-approximation.

We use an indirect approach based on finding the uncertainty inherent in using a given amount of information.

$N^{-1}(y)$ is the set of all indistinguishable problem elements, having the same information as f. Similarly $SN^{-1}(y)$ is the set of indistinguishable solution elements. The ***radius of information*** is defined as:

$$r(N) \equiv r^{wor}(N) = \sup_{y \in N(f)} radius(SN^{-1}(y))$$

The radius of information measures the intrinsic uncertainty of solving the problem with given information (independent of algorithm).

The ***information complexity*** and the ***combinatory complexity*** are the minimal information and combinatory costs of calculating an ε-approximation, respectively.

A lower bound on complexity can be set by the cardinality of N because there must be at least n evaluations of f(x), each costing c. An upper bound recognizes any additional combinatory cost to arrive at the approximation.

2.3 An Example: Integration (Concluded). If the information N is non-adaptive then zero data provides the worst case. Since the integral is a linear functional, symmetric about the origin, and convex we will always be able to find nonadaptive information of the same cardinality or less, with the same radius or less than an adaptive alternative. Hence we need only consider the non-adaptive case. We can calculate extremely tight bounds:

$$c\left[\frac{L}{4\varepsilon}\right] \le comp^{wor}(U_\varepsilon) \le cost^{wor}(U_\varepsilon) \le (c+1)\left[\frac{L}{4\varepsilon}\right]$$

The size of the error is:

$$|S(f)-U_n(f)| \le Lip\frac{(f)}{4n}$$

3. Breaking the Curse of Dimensionality.

Most mathematical models are multivariate and many of those suffer the curse of dimensionality if one insists on a guaranteed maximum error for every input. There are only two ways to break this curse (which work on some problems):

1. We can weaken the worst case guarantee, accepting instead a stochastic assurance.
2. We can change the class of inputs.

Clever approximations cannot break the curse.

We will focus on a multivariate numerical integration problem, INT, because it is common, general and instructive, and it is the best researched in IBC.

Bakhvalov (1959) showed that the ε-complexity of d-dimensional integration problem for integrands of total smoothness r is:

$$comp^{wor}(\varepsilon, d, \mathrm{INT}) = \Theta\left(cost(d)\left(\frac{1}{\varepsilon}\right)^{\frac{d}{r}}\right)$$

Smoothness r is defined by bounding the derivatives of the functions. The class of functions F_r control the size of all derivatives of order at most r, while F_r* controls only the nth-order derivative. F_r is often easier to work with and $F_r \subset F_r^*$ so any negative result, such as an exponential lower bound on complexity, on the class F_r also applies to F_r*.

If r=0 then the ε-complexity is infinite and the problem is unsolvable. If r>0 then the complexity depends exponentially on dimension d – the curse of dimensionality. The problem is tractable in 1/ε if there exist a function K and a number p (independent of d and ε) such that comp(ε,d) ≤ (c(d) +2)K(d)(1/ε)^p. Similarly for d. The best value of p is called the exponent of the problem.

The complexity of many other problems is of similar form including approximation, partial differential equations, integral equations, Markovian decision processes and non linear optimization.

The intractability of many problems has been proven in IBC with arguments at the information level while they remain only conjectures in discrete complexity.

We can relax the requirement for the ***worst case setting*** to using randomised algorithms working on randomised number of samples n and randomised sample points t. We can say that comp^{ran}(ε) ≤ comp^{wor}(ε) since any deterministic approximation can be treated as a randomised approximation.

The Monte Carlo algorithm is the classic example of using randomised information for which only 1/(ε)2 function evaluations are necessary and sufficient to compute an *ε-approximation*. Here, comp^{ran}(ε) = c(d)ε^{-2}(1+o(1)), and randomisation breaks intractability for the integration problem at the expense of dropping the absolute assurance of a worst case setting for a stochastic assurance.

Pseudo-random Monte Carlo behaves roughly the same as the Monte Carlo method with random points, provided the functions are relatively smooth and suitable precautions are taken with the pseudo-random number generator.

The cost of Monte Carlo is always Θ(c(d)(1/ε^2)), independent of the smoothness of the integrand class (it does not take advantage of the smoothness). Bakhvalov (1959) presented a method which does take advantage of smoothness and is optimal with $comp^{ran}(\varepsilon, d, \mathrm{INT}) = \Theta\left(c(d)\left(\frac{1}{\varepsilon}\right)^{\frac{2\mathrm{d}}{2\mathrm{r}+d}}\right)$.

This method is optimal for r=0 and not optimal for r>0 but the margin of non-optimality decreases as d increases for fixed r.

Unfortunately there are intractable problems for which randomisation does not help.

Alternatively, we can choose the ***average case setting*** and provide the assurance that the average error is at most ε. In this setting both the information and the algorithm are chosen deterministically an we settle for an expected error with respect to a measure on the problem elements. We can say that comp^{avg}(ε) ≤ comp^{ran}(ε) ≤ comp^{wor}(ε) .

The problem of selecting sample points to achieve good average cost has been open since Sacks & Ylvisaker (1966).

By exploiting L^2 discrepancy theory and shifted Hammersly points, an optimal algorithm will have: $n = \Theta\left(\left(\frac{1}{\varepsilon}\right)\left(\log\frac{1}{\varepsilon}\right)^{\frac{d-1}{2}}\right)$. Unfortunately the proof is not constructive in the shifts required of the Hammersly points.

Thus approximating continuous functions in the average case setting is tractable in 1/ε. However, the average case complexity depends on the measure. For instance the problem is tractable for the Winer sheet measure and intractable for the isotropic Wiener measure. Thus the average case setting is not a panacea for curing intractability.

The second way of trying to break intractability is by choosing a different set of inputs. We will look at a different notion for 'integrands of smooth r'. We consider the unit ball $\dot{F}_r$ in the space of functions having bounded mixed derivatives which contains functions whose smoothness in any particular direction is r.

$$comp^{wor}(\varepsilon, d, \mathrm{INT}) = \Theta\left(c(d)\left(\frac{1}{\varepsilon}\right)^{\frac{1}{r}}\left(\log\frac{1}{\varepsilon}\right)^{\frac{d-1}{2\mathrm{r}}}\right)$$

$$comp^{avg}(\varepsilon, d, \mathrm{INT}) = \Theta\left(c(d)\left(\frac{1}{\varepsilon}\right)^{\frac{1}{r+1}}\left(\log\frac{1}{\varepsilon}\right)^{\frac{d-1}{2(r+1)}}\right)$$

Hyperbolic-cross points are almost optimal in both cases.

Using this same approach of bounded mixed derivatives for th approximation problem we get:

$$comp^{wor}(\varepsilon, d, \mathrm{INT}) = \Theta\left(c(d)\left(\frac{1}{\varepsilon}\right)^{\frac{1}{r-½}}\left(\log\frac{1}{\varepsilon}\right)^{\frac{(d-1)r}{r-½}}\right)$$

$$comp^{avg}(\varepsilon, d, \mathrm{INT}) = \Theta\left(c(d)\left(\frac{1}{\varepsilon}\right)^{\frac{1}{r+½}}\left(\log\frac{1}{\varepsilon}\right)^{\frac{(d-1)(r+1)}{r+½}}\right)$$

Hyperbolic-cross points are almost optimal in both cases.

So far we have been using the Θ notation which shows dependence on 1/ε but hides factors that depend on d or r. However, estimates regarding these factors are unavailable and believed to be very hard.

Wozniakowski introduced the concept of a problem being strongly tractable if there exist a non-negative numbers K and p such that:
comp(ε,d) ≤ (c(d)+2)K(1/ε)^p.

Part Two: Some Interesting Topics.

4. Very High-Dimensional Integration and Mathematical Finance.

The valuation of financial instruments often required th calculation of very high-dimensional integrals (360+). Because of its 1/ε^2 cost, Monte Carlo is widely used in many applications to gain a stochastic assurance of the worst case. We can do better than this if we are willing to shift to the average case setting.

Since the average case setting is deterministic we need a procedure to obtain a “small” set of sample points in d dimensions, which is uniform. Uniform means that the fraction of points lying within any rectangular subregion (parallel to axes) is proportional to the volume of that subregion. The discrepancy of a sequence of points is a measure of its deviation from uniformity.

Calculation of low discrepancy sequence (LDS) in L^2 and L$^\infty$ can be archived in:

$$\inf_{z_1, \ldots, z_n \in [0,1]^d} D^{(2)}_{n,d}(z_1, \ldots, z_n) = \Theta\left(n^{-1}(\log n)^{(d-1)/2}\right)$$

$$\inf_{t^*_1, \ldots, t^*_n \in [0,1]^d} D^{*}_{n,d}(t^*_1, \ldots, t^*_n) = \Theta\left(n^{-1}(\log n)^{d-1}\right)$$

This delivers asymptotically superior performance in n but, mathematical finance problems tend to have modest n and very large d. Hence, we reach the conclusion that LDS should not be used for high dimensional problems (ie d≥12)

In 1992 JFT tested the efficacy of these quasi-Monte Carlo (QMC) algorithms (using LDS) for the valuation of financial derivatives.

- Both QMC algorithms outperformed MC
- The convergence of the QMC algorithms was smoother enabling automatic termination.
- MC was very sensitive to the initial seed.

Further empirical results in 1995 found:

- Both QMC algorithms beat the MC algorithm by a wide margin.
- QMC algorithms outperform the MC algorithm for a small number of sample points.
- The advantage of QMC algorithms over the MC algorithm if further amplified as the accuracy demands grow.

A number of hypothesis have been advanced to explain the observed empirical results. On such by Sloan and Wozniakowski is based on the observation that many problems of mathematical finance are highly non-isotropic. The paper proves that if there are a minimal number of sample points that can reduce the error by a factor ε, then there exists a QMC method that the number of points required is less than Cε^{-p} with p≤2. However it is a nonconstructive proof and provides no method of finding these important points. If p≈1 then the empirical results could be explained.

The performance of QMC is very impressive. There is evidence that this performance extends to isotropic problems. The empirical convergence rate of QMC is 1/n, as if it see that this is really a one dimensional problem.

5. Complexity of Path Integration.

In path integration (or functional integration), we wish to approximate integrals of the form:

$$S(f) = \int_X f(x)\mu(dx) \quad \forall f \in F$$

where μ is a probability measure on an infinite-dimensional space X and F is a class of functions f defined on X.

The usual method for computing a path integral is:

- approximating the original infinite-dimensional integral by one of finite dimension d, and then
- using Monte Carlo to solve the d-dimensional integral.

This provides an ε-approximation but it requires $\Theta(\varepsilon^{-2})$ function evaluations which can be expensive for small ε.

For a problem class having finite regularity, path integration is intractable in the worst case setting. However, the problem is tractable in the randomized setting, and Monte Carlo is essentially optimal.

Wasilkowski and Wozniakowski provide an algorithm which outperforms Monte Carlo for the class of entire functions with worst case cost of Kcε^{-p} where p≤2 (or p=2/3 for the Wiener measure).

6. Are Ill-Posed Problems Solvable?

Pour-El & Richards' (1988) monogram used computability theory to show that there are partial differential equations with computable initial conditions and non-computable solutions. Werschulz (1987) obtained an analogous conclusion regarding unbounded solutions using IBC in a page.

In a seminal paper, Hadamard (1902) developed the notion of a 'correctly set' or well posed problem's being one for which the solution operator is continuous (a small change in f results in a small change in S(f)). Many important ill-posed problems occur in practice.

- A problem is solvable if we can compute an ε-approximation for any ε>0.
- A problem is weakly solvable if there exists an $\varepsilon_0 > 0$ such that we can compute an ε-approximation if $\varepsilon \geq \varepsilon_0$.
- A problem is unsolvable if we cannot compute an ε-approximation no matter how large we choose ε to be.

Linear transformations S of normed linear spaces are continuous iff they are bounded. Thus a linear problem is well-posed if S is bounded, and ill-posed if S is unbounded.

Wershulz proved that if S is unbounded then $e^{wor}(\Phi,N)=\infty$, hence $comp^{wor}(\varepsilon)=\infty$. Thus ill-posed problems are unsolvable in the worst case.

A problem is well-posed on the average if S is bounded on the average, and is ill-posed on the average, if S is not bounded on the average.

When a problem is well-posed on the average, it is solvable in the average case setting. However, when a problem is ill-posed on the average, it can be solvable, weakly solvable or unsolvable on the average case setting depending on the measure μ.

If the measure μ is a Gaussian measure then the problem is solvable in the average case setting iff it is bounded on the average. Every linear ill-posed problem is solvable on the average, for all Gaussian measures.

7. Complexity of Nonlinear Problems.

Nonlinearity is not a property – it is the lack of a property. Hence, nonlinear problems must be dealt with on a case by case basis.

It is only recently that the complexity of nonlinear problems has been studied in anything other than the worst case setting.

The Fredholm Problem of the Second Kind.

$$u(x) - \int_{[0,1]^d} k(x,y)u(y)dy \; = \; f(x) \;\; \forall x \in [0,1]^d$$

This type of problem considers the solution u at a fixed but arbitrary point $x_0 \in [0,1]^d$. If we have partial information on both the kernel k and f, then the solution operator mapping (k,f) onto $u(x_0)$ is nonlinear.

Heinrich and Mathe (1993) showed that in the randomized setting, the n^th^ minimal randomized error is proportional to $n^{-(r/(2d)+ \frac{1}{2})}$, and that the ε-complexity is proportional to $\varepsilon^{12d/(r+d)}$.

Note that the nth randomized minimal radius is the product of two factors. The first, $\Theta(n^{-r/(2d)})$, is the worst case minimal radius for the Fredholm problem, and the second , $\Theta(n^{-\frac{1}{2}})$, is the randomized minimal radius for the integration problem with continuous integrals. Accordingly their solution has both a deterministic and a stochastic part. The deterministic part makes a finite element approximation of k and the second uses a Monte Carlo scheme to find approximate solutions to the Fredholm problem.

Nonlinear Equations.

Sikorski surveyed worst case complexity for nonlinear equations. A typical result is that the bisection method is an optimal algorithm for many such classes, so that $comp^{wor}(\varepsilon) = \Theta(\log(1/\varepsilon))$.

In the average case setting:

- adaptive stopping $comp^{avg}(\varepsilon) = \Theta(\log(1/\varepsilon))$ is exponentially better than nonadaptive stopping $comp^{avg}(\varepsilon) = \Theta(\log\log(1/\varepsilon))$, and
- if adaptive stopping is allowed, bisection is exponentially non-optimal.

8. What Model of Computation Should Be Used by Scientists?

Cons of the Turing machine model:

- Not natural to use a discrete model of computation in conjunction with the continuous model of science.
- Not predictive of running time of scientific computation on a digital computer.
- No all 'reasonable' machines are equivalent to Turing machines.

Pros of the real-number model.

- 'Natural' for continuous mathematical models.
- Predictive of computer performance on scientific problems
- Utilizes the power of continuous mathematics.

Cons of the real-number model.

- It is impossible to construct a physical device that implement the real-number model.
- It is preferable to use a finite state abstraction of a finite state machine.

9. Do Impossibility Theorems from Formal Models Limit Scientific Knowledge?

Gödel's **undecidability** results cannot be used to infer a limit to scientific knowledge because:

1. There might be something special about mathematical models describing nature that makes then decidable.
2. The mathematical models are not unique as the scientist is free to choose the model that suits the purpose.

If it could be proven that all viable mathematical models were undecidable then this would be a limit. But is not clear that this can be done.

Intractability results cannot be used to infer a limit on scientific knowledge for similar reasons.

Uncomputable results involve almost all real numbers and a number of scientists have expressed unease in physics.

Computability is an asymptotic concept so it is the costs of the algorithms rather than non-computability that may pose a fundamental limitation.

There are arguments that certain observables in quantum gravity may be uncomputable. There are simple partial differential equations with computable initial conditions but non computable solutions (ill-posed). However, the non-solvability of ill-solved problems is a worst-case phenomenon that disappears in the average case for Gaussian measures.

Hence, there has not been convincing evidence that **non-computability** limits scientific knowledge.

10. Complexity of Linear Programming.

Although the complexity of Linear Programming is polynomial in the Turing machine model, the question of its complexity in the real number model remains open.

Khachian studied the ellipsoid algorithm and proved that LP is polynomial in the Turing machine model but Traub and Wozniakowski showed that the cost of this ellipsoid program is not polynomial in the real number model.

The Real-Number model utilizes a real-number system, exact arithmetic (+, -, *, /, comparison and √), and has unit cost for each operation.

The Random Access Machine (polynomially equivalent to the Turing machine) utilities the integer numbering system, exact or approximate arithmetic, and has cost proportional to the size of th numbers.

The definition of polynomial complexity, assuming $\varepsilon=0$, involved the problem size n in the Real-Number model and the number of digits L required to represent the input in the RAM model.

11. Complexity of Verification.

Chapters 1-10 involved using IBC to understand the optimum computation of ε-approximation.

In verification, we are given a problem and a proposed answer, and are asked to check whether the proposed answer is within ε of the true answer. Verification is a decision problem and surprisingly, checking is more difficult than computing.

The verification operator checks if the proposed answer g is within ε of the Solution S(f) of the problem element f. We only have partial information on f and the Information N(g;f) depends on both the problem element f and the proposed solution element g.

Wozniakowski (1992) proved that verification is unsolvable in the worst case setting, even when the solution operator and information are linear. That is, there exists no approximation that always gives the correct verification regardless of linearity or adaptive information.

Alternatively, in the **probabilistic setting** we are willing to accept an ε-approximation that fails on a set of measure at most δ. Verification can be either trivially or exponentially harder than computation depending on the relationship between ε and δ.

Alternatively, **relaxed verification** introduces an additional relaxation parameter $\alpha>0$ such that there is a region between a clear YES or NO result to the verification. Novak and Wozniakowski (1992) showed that $\text{comp}^{\text{ver}}(\varepsilon, \alpha) \leq \text{comp}^{\text{com}}(\varepsilon\alpha/2)+1$ for any problem linear or nonlinear (and the bound is sharp in the linear case).

12. Complexity of Implementation Testing.

Testing that large complicated systems conform to the specification is difficult and important. Hennie (1964) provided an elegant algorithm for constructing a checking experiment of exponential length. Recently, polynomial time algorithms were obtained by Lee & Yannakakis (1995)

Continuous problems have infinite points so a finite number of test points is insufficient in the worst case.

The testing regime can be **relaxed** so that a maximum number of errors are permitted.
In the **linear, relaxed, worst case** setting:

1. The relaxed testing problem is solvable
2. There exists a universal test consisting of orthogonal elements on the boundary of F
3. The minimal number of tests required is close to the number of tests used by the universal test set (hence it is essentially optimum).

In the **linear, relaxed, average case** setting:

1. A finite number of tests from an arbitrary complete orthogonal basis is conclusive,
2. The eigenvectors of the covariance operator of the probability measure yield an almost optimal test sequence. This sequence is universal.

In the nonlinear case:

- finite testing is inconclusive
- testing nonlinear operators is decidable in the limit (but the number of tests is arbitrarily large).
- A finite test set is conclusive for relaxed testing but the cost may be prohibitive
- Probabilistic testing for failure of at most β is prohibitively expensive in infinite dimensional space
- Weak β-conformance testing only requires samples from finite dimensional subspaces.
- The test sequences for non-linear operators are all universal.

The three legs of IBC are computation, verification and testing.

13. Noisy Information.

Up until this chapter the discussion has ignored the effects of noise, concentrating on exact partial information. Much practical information contains noise.

We consider two kinds of noisy information:

1. **Bounded noise** where we know that $||\xi||\leq\delta$ such that the worst case over problem elements is combined with the worst case over noise information $e^{wor\text{-}wor}(\Phi,N)$
2. **Stochastic noise** where we might know that the noise ξ is give by a probability distribution π (ie with mean zero and variance σ^2). This setting combines the worst case over problem elements with the average case over noisy information for $e^{wor\text{-}avg}(\Phi,N)$

Once we allow inexact information, information-level arguments can be used for problems with complete, yet noisy, information.

Plaskota (1996) shows that in some sense, noisy information is better than exact information if:

1. we are trying to solve the integrations problem for large d,
2. we are willing to live with a weakened assurance,
3. the distribution of the noise is known.

For integration with :

- no adaption and σ=0: $f^{non}(n,0) = \Theta(n^{-1/d})$
- no adaption and σ>0: $f^{non}(n,\sigma) = \Theta(n^{-1/d})$
- adaptive and σ>0: $f^{a}(n,\sigma) = \Theta(n^{-1/2})$

Halpern studied clock synchronization in distributed networks with k nodes (processors) and transmission delays between L and H. No algorithm can synchronize the clocks better than $\delta=(H-L)(k-1)/k$.

Wasilkowski refined the result assuming transmission times with a known probability distribution and n messages and a minimal diameter tree T that spans the network.

$$error\left(\left[\frac{n}{k-1}\right],2\right)\sqrt{diam(T)}$$

14. Value of Information in Computation.

Packet (1992) introduced the “value of information”. The value of information agrees with the concept of mutual information in some cases and disagrees in others.

“Fascination with a notion of entropy to represent disorder, uncertainty, or loss of information has been a major theme of scientific thought.” pg 94

The IBC value of information is defined by radius of information r(N), which measures the intrinsic uncertainty due to the information N. It is well defined even for infinite-dimensional problems.

This discussion is restricted to $r^{avg}(N)$, the average case radius of information to facilitate comparison with entropy based concepts from information theory.

The value of information N for a problem is:

$$V(N)=\log_2\frac{r^{avg}(0)}{r^{avg}(N)}$$

Entropy is defined as: $H(X)=-\sum_{x\in X}\mu(x)\log_2\mu(x)$

Mutual Information: I(X;Y) = H(X) – H(X|Y)

When the probability distributions are continuous the entropy calculation involves an integration with the challenges addressed by IBC.

Packet showed that V(N) = I(X;Y) for a continuous binary search (approximate a real number in the unit interval by asking yes/no questions) and integration of continuous functions using the Wiener measure. Note that the second example is a linear approximation to a continuous function. V=I fairly generally for linear functions but there are also linear and non-linear cases where V≠I.

- The radius of information is defined in all IBC settings
- V(N) has a natural interpretation in terms of bits of accuracy.
- V(N) agrees with I(X;Y) for some computational problems and sensibly disagrees in others such as the well know difficulties of defining I(X;Y) for non-atomic probability measures.
- Mutual information and variance do not depend on an error criterion, where-as the value of information dies.
- The radius r(N) is an intrinsic measure of uncertainty, hence so is V(N).

15. Assigning Values to Mathematical Hypothesis.

Computational complexity permits us to quantify the value of mathematical hypothesis. Any change in complexity which results from using hypothesis H_1 rather than hypothesis H_2 is solely due to the change in hypothesis.

The complexity of univariate integration for the unit ball of functions having smoothness r is:

$$comp(\varepsilon)=\Theta\left(\left(\frac{1}{\varepsilon}\right)^{\frac{1}{r}}\right)$$

Thus we know the effect of changing the smoothness hypothesis. If we change from once differentiable to twice differentiable then the complexity decreases by a factor of 1/√ε.

Non-linear constrained optimization problems are intractable for smooth functions but tractable for convex functions that satisfy the Lipschitz condition with a uniform constant.

Smooth: $comp_r(\varepsilon)=\Theta\left(\left(\frac{1}{\varepsilon}\right)^{\frac{d}{r}}\right)$

Convex: $comp_c(\varepsilon)=\Theta\left(\frac{1}{\varepsilon}\right)$

These results quantify the relative value of smoothness and convexity for this optimization problem.

When linear systems become very large (ie sparse 10^4x10^4 matrices) then the time for a direct solution may be prohibitive so it may be better to settle for an approximate solution from partial information. Then, for example, we are able to give a partial answer to how much easier it is to solve a large linear system if the matrix is positive definite.

Further Reading

- ch2 pg20 Haussler,D. (1988) Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. Artificial Intelligence, 36, 177-221
- ch3 pg35 Sacks,J. & Ylvisaker,D. (1966) Designs for regression with correlated errors. Ann Math. Statist. 37, 68-89.
- ch4 pg51 Koksma-Hlawka
- ch4 pg51 Wozniakowski
- ch12 pg 86 Lee,D. & Wozniakowski, H. (1997) Testing Linear Operators: An average case analysis. BIT, (Submitted)
- ch13 pg 88 Plaskota,L. (1996) Noisy Information and Computational Complexity. Cambridge University Press.