Coin Tossing Algorithms for Integral Equations and Tractability

Erich Novak

See our book “Tractability of Multivariate Problems II” with H. Woźniakowski; joint work with S. Heinrich and H. Pfeiffer

University of Jena
The Problem

Compute $u(s)$, integral equation

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

on $[0,1]^d$ with Lipschitz kernel $k$, $\|k\|_\infty < \alpha < 1$ and right hand side.

Deterministic algorithms: Optimal order $e_n \asymp n^{-1/(2d)}$, curse of dimension.

Optimal order with MC (Heinrich & Mathé 1993)

$$e_n \asymp n^{-1/2 - 1/(2d)}.$$ 

How much randomness is needed?
Can we work with “few” random bits?
Compare with recent book of Sugita.
Reduction of Randomness

In order to approximate $I_d(f) = \int_{[0,1]^d} f(x) \, dx$ up to error $\varepsilon > 0$ with the classical MC-method, one needs roughly $\varepsilon^{-2}$ function values and

$$d \cdot \varepsilon^{-2}$$

random numbers from $[0,1]$.

Trick of Bakhvalov (1964): Pairwise independent sample points for MC are ok, take

$$x_k = x + k \cdot y \mod 1$$

with $2d$ random numbers for $x, y \in [0,1]^d$.

Hence $2d$ random numbers are enough.

What about random bits?
Approximation of Means

Approximate, for \( f = (f(0), \ldots, f(N-1)) \in \mathbb{R}^N \),

\[
S_N(f) = \frac{1}{N} \sum_{i=0}^{N-1} f(i).
\]

Assume that \( f \) is from the set \( F = \mathcal{B}(L_p^N) \), the unit ball of the space \( L_p^N = L_p(\mu) \) where \( \mu \) is the equidistribution on \( \{0, 1, \ldots, N-1\} \) and \( 1 \leq p \leq \infty \).

\( N = 2^s \): class. MC-meth. \( A_n^\omega(f) = \frac{1}{n} \sum_{j=1}^{n} f(i_j^\omega) \)

To implement this algorithm, we need \( n \log_2 N \) random Bits.

Can we do better?
Approximation of Means

Bakhvalov's idea & field with $N$ elements obtain algorithm with the same error, $n$ function values and $2\log_2 N$ random bits.

Illustration: choice of the $i^\omega_j$ for $N$ prime:
Take $x, y \in \{0, 1, \ldots, N - 1\}$ independently acc. to the uniform distribution and put

$$i^\omega_j = x + (j - 1) \cdot y \mod N.$$ 

Heinrich, N., Pfeiffer (2004):
Let $1 \leq p \leq \infty$, $N \in \mathbb{N}$ and $\beta > 1$. Then, for $\beta n < N$,

\begin{align*}
e_n(S_N, B(L_p^N)) \asymp n^{-1/2} & \quad \text{for } 2 \leq p \leq \infty \text{ and} \\
e_n(S_N, B(L_p^N)) \asymp n^{-1+1/p} & \quad \text{for } 1 \leq p < 2.
\end{align*}

Roughly $\log_2 N$ random bits are needed.
Approximation of Integrals

Approximate

\[ I_d(f) = \int_{[0,1]^d} f(x) \, dx \]

where \( f \) is from a Sobolev class:

\( r \in \mathbb{N}, \, 1 \leq p \leq \infty, \, D = [0,1]^d \)

\[ W_p^r(D) = \{ f \in L_p(D) : \partial^\alpha f \in L_p(D), |\alpha| \leq r \} \]

\[ \|f\|_{W_p^r(D)} = \left( \sum_{|\alpha| \leq r} \|\partial^\alpha f\|_{L_p(D)}^p \right)^{1/p} \]

\( \partial^\alpha \quad \text{weak partial derivative} \)

\( \mathcal{B}(W_p^r(D)) \quad \text{unit ball of } W_p^r(D) \)

Assume that \( r/d > 1/p \) (Sobolev embedding condition).
Result for Algorithms that use Random Bits

Heinrich, N., Pfeiffer (2004):

Let $1 \leq p \leq \infty$, $r, d \in \mathbb{N}$, $r/d > 1/p$. Then there is a constant $c > 0$ such that the optimal order

$$e_n(I_d, \mathcal{B}(W^r_p(D))) \asymp n^{-r/d - 1/2} \quad (2 \leq p \leq \infty)$$

$$e_n(I_d, \mathcal{B}(W^r_p(D))) \asymp n^{-r/d - 1 + 1/p} \quad (1 \leq p < 2)$$

can be obtained with $c \log n$ random bits.

The upper bound is proven by a discretization technique: splitting the problem into a sequence of discrete summation problems.
The Integral Equation

Compute \( u(s) \), integral equation

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

on \([0,1]^d\) with Lipschitz kernel \( k \), \( \|k\|_\infty < \alpha < 1 \) and right hand side. Use the Neumann series and apply the above results for summation and/or integration and obtain, see N. & Pfeiffer (2004), the upper bound:

\[
    \text{cost} \leq \varepsilon^{-2} + d (\log \varepsilon^{-1})^2.
\]

Need \( \varepsilon^{-2} \) function values and \( d (\log \varepsilon^{-1})^2 \) random bits.

*Problem is tractable for random bit MC methods.*