Coláiste na hOllscoile Corcaigh, Éire
University College Cork, Ireland

CS4618 Artificial Intelligence I

Today: Assessment of Randomised Search Heuristics: Black-Box Complexity

Thomas Jansen

November 9th

Plans for Today

1. Black Box Complexity
    - Reminder and Lower Bounds
    - Simple Results on Black Box Complexity
2. Black Box Complexity of Unimodal Functions
    - Introduction
    - Main Result and Proof
3. Summary
    - Summary & Take Home Message

Remember

Let $\mathcal{F} \subseteq \{f \colon S \to V\}$ be a class of functions, A a black box algorithm for $\mathcal{F}$, x_t the t-th search point sampled by A.

- optimisation time of A on $f \in \mathcal{F}$
  $T_{A,f} = \min\{t \mid f(x_t) = \max\{f(x) \in S\}\}$
- worst case expected optimisation time of A on $\mathcal{F}$
  $T_{A,\mathcal{F}} = \max\{\mathrm{E}(T_{A,f}) \mid f \in \mathcal{F}\}$
- black box complexity of $\mathcal{F}$
  $B_{\mathcal{F}} = \min\{T_{A,\mathcal{F}} \mid A \text{ is black box algorithm for } \mathcal{F}\}$
- $\forall \mathcal{F} \colon B_{\mathcal{F}} \leq |\mathcal{F}|$
- $f^* := \{f_a \mid a \in \{0,1\}^n\}$ where $f_a(x) := f(a \oplus x)$
- $\forall \mathcal{F} \subseteq \{f \colon \{0,1\}^n \to \mathbb{R}\} : B_{\mathcal{F}} \leq 2^{n-1} + 1/2$

Two-Player Zero-Sum Games

Remember two player zero-sum games

in general $n \times m$-pay-off matrix $M = (M_{i,j})$

row player aims at $V_r := \max_i \min_j M_{i,j}$

column player aims at $V_c := \min_j \max_i M_{i,j}$

Game solved iff $V_r = V_c$: optimal strategy for both players

Deterministic players (pure strategies) are somewhat boring. . .

Randomisation: mixed strategies

row player chooses probability distribution p over rows

column player chooses probability distribution q over columns

Mixed Strategies in Two-Players Zero-Sum Games

$$\mathrm{E}\,(\text{pay-off}) = \sum_{i=1}^{n} \sum_{j=1}^{m} p_i M_{i,j} q_j$$

row player aims at $V_r := \max_p \min_q \mathrm{E}\,(\text{pay-off})$

column player aims at $V_c := \min_q \max_p \mathrm{E}\,(\text{pay-off})$

Minimax Theorem (von Neumann)
$\max_p \min_q \mathrm{E}\,(\text{pay-off}) = \min_q \max_p \mathrm{E}\,(\text{pay-off})$

Loomis' Theorem
$\max_p \min_j \mathrm{E}\,(\text{pay-off}) = \min_q \max_i \mathrm{E}\,(\text{pay-off})$

Two-Player Zero-Sum Games and Algorithm Design

Why should we care?

Consider algorithm design for some
problem with finite set of possible inputs of finite size $\mathcal{I}$
allowing for a finite number of deterministic algorithms $\mathcal{A}$
as a two-player zero-sum game.

One player chooses (i. e., designs) an algorithm $A \in \mathcal{A}$.

The other player chooses the input $I \in \mathcal{I}$.

The run time $T_{A,I}$ is the pay-off.

We can have

- randomised algorithms by probability distributions q over $\mathcal{A}$
- probability distributions p over $\mathcal{I}$.

Yao's Minimax Principle

Where does this perspective lead to?

Minimax Theorem (von Neumann)

$\max_p \min_q \mathrm{E}\left(T_{A_q,I_p}\right) = \min_q \max_p \mathrm{E}\left(T_{A_q,I_p}\right)$

Loomis' Theorem

$\max_p \min_A \mathrm{E}\left(T_{A,I_p}\right) = \min_q \max_I \mathrm{E}\left(T_{A_q,I}\right)$

Yao's Minimax Principle

For all distributions p over $\mathcal{I}$ and all distributions q over $\mathcal{A}$:

$\min_A \mathrm{E}\left(T_{A,I_p}\right) \leq \max_I \mathrm{E}\left(T_{A_q,I}\right)$

in words

We get a lower bound for the

worst-case performance of a randomised algorithm by

proving a lower bound on the worst-case performance of an

optimal deterministic algorithm

for an arbitrary probability distribution over the inputs.

$B_{\textsc{Needle}^*}$

Theorem

$B_{\textsc{Needle}^*} = 2^{n-1} + 1/2$

Proof by application of Yao's Minimax Principle

The upper bound coincides with the general upper bound.

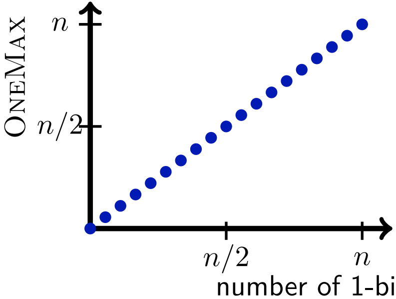
We consider each $\textsc{Needle}_a$ as possible input.
We choose the uniform distribution.
Deterministic algorithms sample the search space in a pre-defined order without re-sampling.
Since the position of the unique global optimum is chosen uniformly at random,
we have $\mathrm{Prob}\,(T = t) = 2^{-n}$ for all $t \in \{1, \ldots, 2^n\}$.
This implies $\mathrm{E}\,(T) = \sum\limits_{i=1}^{2^n} i \cdot 2^n = \frac{2^n(2^n+1)}{2^{n+1}} = 2^{n-1} + \frac{1}{2}$. □

Remark We already knew this from NFL.

Another Example Function: OneMax

Definition $\textsc{OneMax}\colon \{0,1\}^n \to \{0,1,\dots,n\}$

$$\textsc{OneMax}(x) = \sum_{i=1}^{n} x[i]$$



Consider $\textsc{OneMax}^* = \{\textsc{OneMax}_a \mid a \in \{0,1\}^n\}$,

$$\textsc{OneMax}_a(x) = \textsc{OneMax}(a \oplus x)$$

$$\textsc{OneMax}_a(x) = \mathsf{H}\left(x, \overline{a}\right)$$

Hamming distance $\mathsf{H}\left(x, y\right) = \sum_{i=1}^{n} x[i] + y[i] - 2x[i]y[i]$

$B_{\textsc{OneMax}^*}$

Theorem

$B_{\textsc{OneMax}^*} = \Omega(n/\log n)$

Proof by application of Yao's Minimax Principle

We choose the uniform distribution.

A deterministic algorithm is a tree with at least 2^n nodes:
otherwise at least one $f \in \textsc{OneMax}^*$ cannot be optimised.

The degree of the nodes is bounded by $n+1$:
this is the number of different function values.

Therefore, the average depth of the tree is bounded below by

$$(\log_{n+1} 2^n) - 1 = \frac{n}{\log_2(n+1)} = \Omega(n/\log n).$$

☐

Remark $B_{\textsc{OneMax}^*} = O(n)$ is easy to see.

Unimodal Functions

Consider $f\colon \{0,1\}^n \to \mathbb{R}$.

We call $x \in \{0,1\}^n$ a local maximum of f,
iff for all $x' \in \{0,1\}^n$ with $\mathsf{H}(x,x') = 1$
$f(x) \geq f(x')$ holds.

We call f unimodal, iff f has exactly one local optimum.

We call f weakly unimodal, iff all local optima are global optima, too.

Observation (Weakly) Unimodal functions can be optimised by local search

Does this mean unimodal functions are easy to optimise?

Unimodal functions

class of unimodal functions
$\mathcal{U} := \{f\colon \{0,1\}^n \to \mathbb{R} \mid f \text{ unimodal}\}$

What is $B_{\mathcal{U}}$?

We want to find a lower bound on $B_{\mathcal{U}}$.

Remember For any point not optimal under a unimodal function, there exists a path to the global optimum

Definition path of length l is
sequence of l points p_1, p_2, . . . , p_l
with $\mathrm{H}(p_i, p_{i+1}) = 1$ for all $1 \leq i < l$

Path Functions

Consider the following functions:

$P := (p_1, p_2, \ldots, p_{l(n)})$ with $p_1 = 1^n$ is a path
not necessarily a simple path

$$f_P(x) := \begin{cases} n+i & \text{if } x = p_i \text{ and } x \neq p_j \text{ for all } j > i, \\ \textsc{OneMax}(x) & \text{if } x \notin P \end{cases}$$

Observation f_P is unimodal.

$\mathcal{P}_{l(n)} := \{f_P \mid P \text{ has length } l(n)\}$

Random Paths

Construct P with length $l(n)$ randomly:

```
1. $p_1 := 1^n$; $i := 2$
2. While $i \leq l(n)$ do
3.     Choose $p_i \in \{x \mid \mathsf{H}(x, p_{i-1}) = 1\}$ uniformly at random.
4.     $i := i + 1$
```

For each path P with length $l(n)$,
we can calculate the probability to construct P randomly this way.

Remark Paths P constructed this way are likely to contain circles.

A lower bound on $B_{\mathcal{U}}$

Theorem $\forall\delta$ with $0 < \delta < 1$ constant: $B_{\mathcal{U}} > 2^{n^{\delta}}$.

For a proof, we want to apply Yao's Minimax Principle.

We define a probability distribution in the following way:

$\delta < \varepsilon < 1$ constant; $l(n) := 2^{n^{\varepsilon}}$

For all $f \in \mathcal{U}$ we define

$$\mathrm{Prob}\,(f) := \begin{cases} p & \text{if } f \in \mathcal{P}_{l(n)} \text{ and } P \text{ is constructed with prob. } p, \\ 0 & \text{otherwise.} \end{cases}$$

Our Proof Strategy

We need to prove that

an optimal deterministic algorithm

needs on average more than 2^{n^δ} steps

to find a global optimum.

We strengthen the position of the deterministic algorithm by

1. letting it know which functions have probability 0.
2. giving away for free the knowledge about any p_i with $f(p_i) \leq f(p_j)$ once p_j is sampled,
3. giving away for free the knowledge about $p_{j+1}, \dots p_{j+n}$ if p_j is the current known best path point and some point not on the path is sampled,
4. giving away for free the knowledge about $p_{l(n)}$ (the global optimum) once p_{j+n} is sampled while p_j is the current known best path point.

Deterministic Algorithm Too Strong?

Omit all circles froms P.
The remaining length $l'(n)$ is called the true length of P.

What lower bound can be proven this way?

at best $(l'(n) - n + 1)/n$

Observation We need a good lower bound on $l'(n)$.

How likely is it to return to old path points?

alternatively What is the probability distribution for the Hamming distance points on the path?

Distance Between Points on the Path

Lemma

$\forall \beta > 0$ *constant*: $\exists \alpha(\beta) > 0$ *constant*: $\forall i \leq l(n) - \beta n$:
$\forall j \geq \beta n$: $\mathit{Prob}\left(H(p_i, p_{i+j}) \leq \alpha(\beta) n\right) = 2^{-\Omega(n)}$

Summary & Take Home Message

Things to remember

- black-box complexity
- Yao's minimax principle
- black-box complexity of unimodal problems

Take Home Message

- Black-box complexity allows for meaningful general lower bounds for RSHs.
- Unimodal problems are not easy to solve.