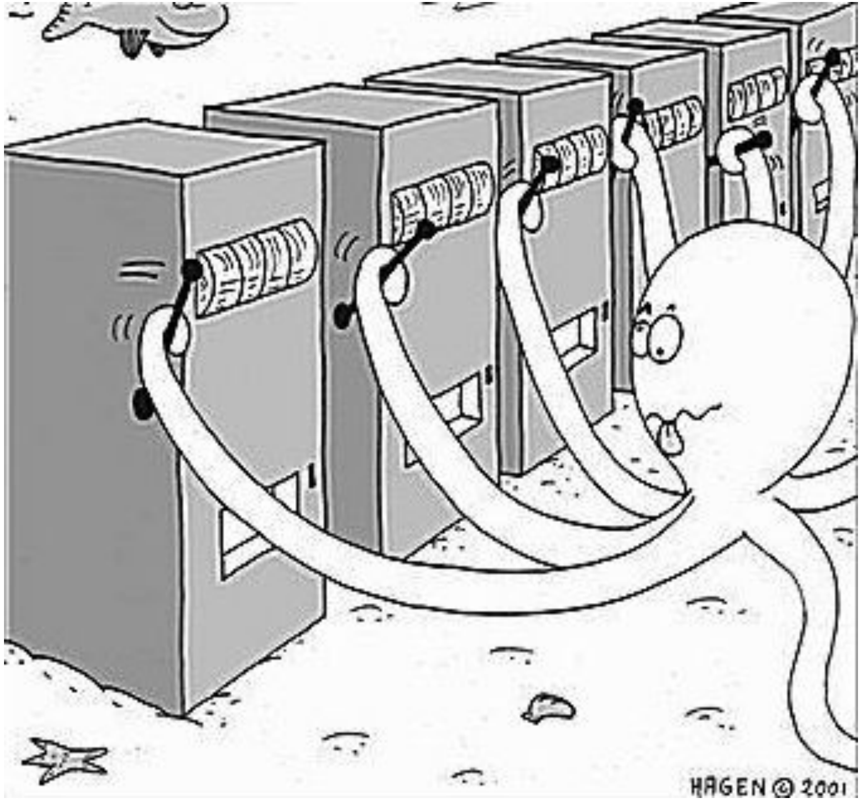
Bandit Optimization

Team Members:
Raavi Gupta (200070064)
Vedang Gupta (200100166)

Course Instructor:
Prof. Ganesh Ramakrishnan

Bandit Optimization

Employing bandits to optimize general black box functions



Source:
https://blog.gofynd.com/multi-armed-bandit-solution-methods-e85e6b19fb2d

Why bandits?

Optimization has been dealt in general in many frameworks previously

In this setting, we focus on the problem of functional optimization with finite number of evaluations

We deal with discrete as well as continuous optimization

Since there is a limited budget the problem shares similarity with many bandit algorithms present today

Not many practical results have been previously shown in the literature (especially for HOO)

Problem statement

Goal

Maximize function $f : \rightarrow R$ given a finite budget n of (noisy or noiseless) evaluations.

Approach

For $t = 1$ to T, select state x_t and observe:

1. Deterministic case: $f(x_t)$
2. Stochastic case: $f(x_t) + \epsilon_t$ (ϵ_t is the noise added)

Discrete Optimization

This set of problems refers to strategies which use a finite number of evaluations to find the best option. This has 3 types of problems:

1) Fixed budget of n evaluations:
    a) Uniform Exploration (UE)
    b) Successive Rejects (SR)
2) Fixed confidence of 1 - δ to choose the optimal arm:
    a) Hoeffding Race
    b) Bernstein Race
    c) Action Elimination
3) Online Optimization methods - Regret is minimized
    a) UCB
    b) UCB-V
    c) MOSS

Methodology

- For the fixed budget and online optimization case, we used a fixed budget of 1000 evaluations
- Sampling was done using one of the sampling strategies mentioned in further slides
- For discrete optimization samples were taken to be noisy
- Each arm was modelled as a Gaussian bandit, where the mean was taken to be the function evaluated at the point and the variance was kept constant
- Each sampled point was considered as an individual arm and then the corresponding algorithm was run to determine the best arm
- At the end of the run, the final accuracy is determined by the sampling strategy and the ability of the algorithm to identify the best arm

Hoeffding and Bernstein Race

Parameter: the confidence level δ.

Let $A = \{1, \ldots, K\}$ and $t = 1$

While $|A| > 1$

(1) sample every option in A for the t-th time.

(2) remove from A all the options having an empirical mean differing from the highest empirical mean by more than $\sqrt{2\log(nK/\delta)/t}$, that is,

$$A \leftarrow A \setminus \left\{ j \in A : \widehat{X}_{j,t} \leq \max_{1 \leq i \leq K} \widehat{X}_{i,t} - \sqrt{\frac{2\log(nK/\delta)}{t}} \right\}.$$

(3) $t \leftarrow t + 1$.

Output the unique element of A.

Hoeffding Race

Parameter: the confidence level δ.

Let $A = \{1, \ldots, K\}$ and $t = 1$

While $|A| > 1$

(1) sample every option in A for the t-th time.

(2) remove suboptimal options from A:

$$A \leftarrow A \setminus \left\{ j \in A : \widehat{X}_{j,t} + \sqrt{\frac{2V_{j,t}\log(nK/\delta)}{t}} + 6\frac{\log(nK/\delta)}{t} \leq \max_{1 \leq i \leq K} \left(\widehat{X}_{i,t} - \sqrt{\frac{2V_{i,t}\log(nK/\delta)}{t}} \right) \right\},$$

where $V_{i,t} = \frac{1}{t}\sum_{s=1}^{t} \left(X_{i,s} - \widehat{X}_{i,t} \right)^2$ is the empirical variance of option i.

(3) $t \leftarrow t + 1$.

Output the unique element of A.

Bernstein race

Sampling methods used

1. Uniform Sampling: All the points are randomly chosen
2. Systematic Sampling: Choose first point randomly and the other ones equally spaced from it
3. Stratified Sampling: The domain of each decision variable is divided into N equally- spaced and non-overlapping intervals and then one value is selected at random from each such interval
4. Halton Sampling: Generates quasi-random numbers of high-dimensionality with a high level of uniformity across the space using prime bases

Halton Sampling

1) d prime numbers are chosen (D = dimension)

2) To generate the ith sample, we write the expression of i with base p i.e

   $i = a_0 + a_1 p + a_2 p^2 + \dots,\qquad 0 <= a_i < p$

3) Now, we get a representation of the point as $r(i,p) = a_0 /p + a_1/p^2 + \dots$

4) We now get the ith sample as $(r(i,p_1),r(i,p_2),\dots,r(i,p_d))$ (For d dimensions)

- For 1 dimensional, we use p = 2
- Halton sampling can be easily extended to higher dimensions too by considering more number of primes

Results on linear functions

ACCURACY

TIME STEPS: delta = 0.1

	SR	UE	UCB	UCB-V	MOSS	AE	Hoeffding Race	Bernstein Race
Uniform	0.67	0.63	**0.68**	0.7	0.67	18661	5609	**1369**
Systematic	0.86	0.76	0.81	0.86	**0.91**	3807	1498	**743**
Stratified	0.71	0.57	0.76	**0.79**	0.65	2996	725	**545**
Halton	0.9	0.79	0.85	0.88	**0.91**	2309	569	**492**

Results on quadratic functions

	ACCURACY: Time steps (1000)					TIME STEPS		
	SR	UE	UCB	UCB-V	MOSS	AE	Hoeffding Race	Bernstein Race
Uniform	0.81	0.59	**0.86**	0.78	0.74	1443	543	**488**
Systematic	0.97	0.88	0.94	**0.98**	**0.98**	725	476	**401**
Stratified	**0.95**	0.87	0.94	0.93	0.93	4830	957	**594**
Halton	**0.99**	0.88	0.94	0.98	0.96	886	383	**335**

Note : We tried multiple functions and they gave similar results, the results noted are for the function f(x) = -(x-0.5)^2, which will have a maxima at x =0.5

Results on sinc functions

ACCURACY | TIME STEPS

	SR	UE	UCB	UCB-V	MOSS	AE	Hoeffding Race	Bernstein Race
Uniform	0.51	0.49	0.45	0.6	**0.62**	1187251	98414	**1910**
Systematic	**0.86**	0.67	0.82	0.8	0.81	183888	37265	**1224**
Stratified	0.75	0.69	0.71	0.72	**0.79**	132604	32486	**1883**
Halton	**0.88**	0.69	0.85	0.81	0.84	31264	13464	**1542**

Note : The function used was sin(10(x+0.3))/10*(x+0.3) which has 2 unequal maximas in [0,1] making it a tougher optimization problem than the previous 2. Also for SR and UE here, 10000 budget was used

Continuous Optimization

HOO (Hierarchical Optimistic Optimization)

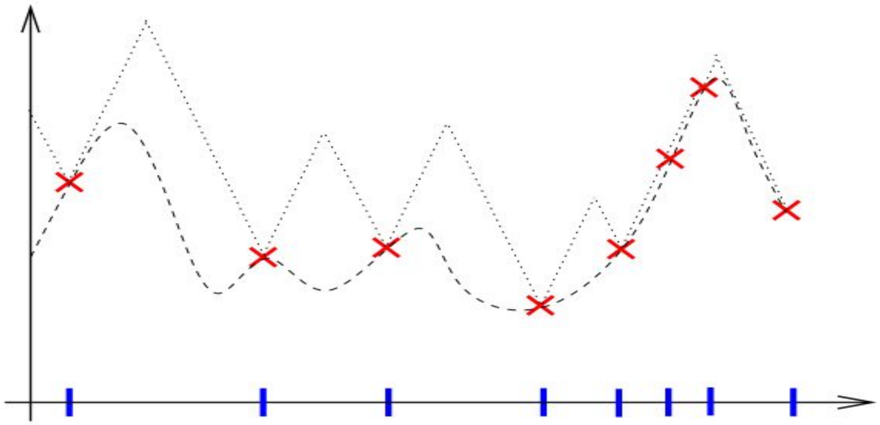
- The selection of the new subspace and candidate point is done in an optimistic manner, by assuming that the function values in unexplored regions are higher than the observed values
- Once a new point is selected for evaluation, the algorithm updates the hierarchical structure by adding or removing candidate points in the corresponding subspace. This process is repeated until convergence is achieved.

```
[0, (0, 1), -0.829989102409106, -0.01348547487626095, 100, -1.3169709007532902]
[1, (0, 0.5), -1.4577681565765739, -1.4577681565765739, 2, -3.803734182865921]
[2, (0, 0.25), -0.9591464739729312, -0.9591464739729312, 1, -4.034000732743224]
[3, (0, 0.125), inf, inf, 0, 0]
[3, (0.125, 0.25), inf, inf, 0, 0]
[2, (0.25, 0.5), inf, inf, 0, 0]
[1, (0.5, 1), -0.829989102409106, -0.7596540545055743, 98, -1.266220629689767]
[2, (0.5, 0.75), -0.8509636322173617, -0.8509636322173617, 8, -1.9639466453620353]
[3, (0.5, 0.625), -0.4639835513598631‌3, -0.46398355135986313, 3, -2.224157474612174]
[4, (0.5, 0.5625), 0.7096389582998428, 0.7096389582998428, 1, -2.32681530047045]
[5, (0.5, 0.53125), inf, inf, 0, 0]
[5, (0.53125, 0.5625), inf, inf, 0, 0]
[4, (0.5625, 0.625), 0.996253516234425, 0.996253516234425, 1, -2.040200742535868]
[5, (0.5625, 0.59375), inf, inf, 0, 0]
[5, (0.59375, 0.625), inf, inf, 0, 0]
[3, (0.625, 0.75), -0.2741144045025601, -0.2741144045025601, 4, -1.7995415338877065]
[4, (0.625, 0.6875), 1.0766848832921625, 1.0766848832921625, 1, -1.9597693754781305]
[5, (0.625, 0.65625), inf, inf, 0, 0]
[5, (0.65625, 0.6875), inf, inf, 0, 0]
[4, (0.6875, 0.75), 0.3992175838814506, 0.3992175838814506, 2, -1.7483484424078966]
[5, (0.6875, 0.71875), 1.286267931960219, 1.286267931960219, 1, -1.748906326810074]
[6, (0.6875, 0.703125), inf, inf, 0, 0]
[6, (0.703125, 0.71875), inf, inf, 0, 0]
```

Sample output of the tree obtained

Optimistic Optimization

Sample the point with the highest upper bound



Parameters: Two real numbers $\nu_1 > 0$ and $\rho \in (0, 1)$, a sequence $(\mathcal{P}_{h,i})_{h \geq 0, 1 \leq i \leq 2^h}$ of subsets of $\mathcal{X}$.

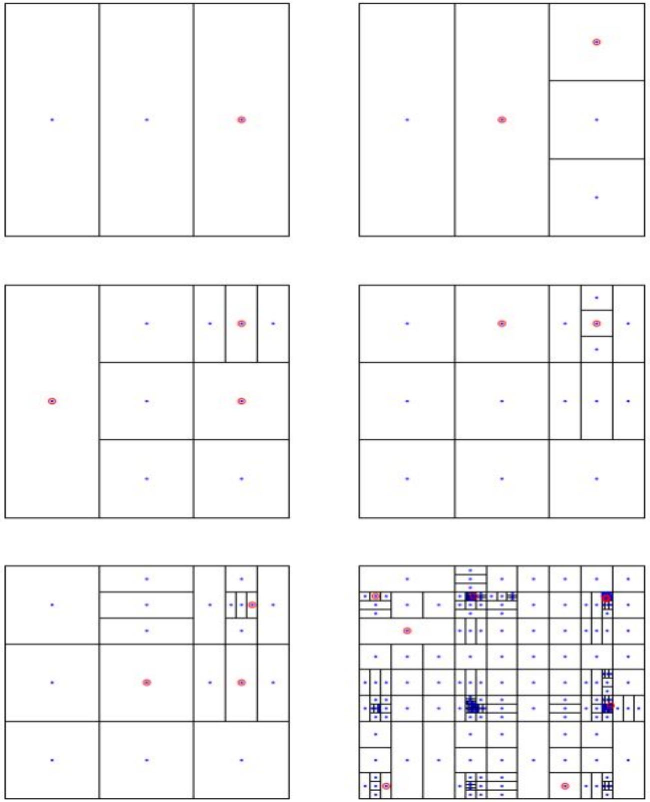
Auxiliary function $\text{LEAF}(\mathcal{T})$: outputs a leaf of $\mathcal{T}$.

Initialization: $\mathcal{T} = \{(0, 1)\}$ and $B_{1,2} = B_{2,2} = +\infty$.

for $n = 1, 2, \ldots$ **do** ▷ Strategy HOO in round $n \geq 1$
  $(h, i) \leftarrow (0, 1)$ ▷ Start at the root
  $P \leftarrow \{(h, i)\}$ ▷ P stores the path traversed in the tree
  while $(h, i) \in \mathcal{T}$ **do** ▷ Search the tree $\mathcal{T}$
    if $B_{h+1,2i-1} > B_{h+1,2i}$ **then** ▷ Select the "more promising" child
      $(h, i) \leftarrow (h + 1, 2i - 1)$
    else if $B_{h+1,2i-1} < B_{h+1,2i}$ **then**
      $(h, i) \leftarrow (h + 1, 2i)$
    else ▷ Tie-breaking rule
      $Z \sim \text{Ber}(0.5)$ ▷ e.g., choose a child at random
      $(h, i) \leftarrow (h + 1, 2i - Z)$
    end if
    $P \leftarrow P \cup \{(h, i)\}$
  end while
  $(H, I) \leftarrow (h, i)$ ▷ The selected node
  Choose option x in $\mathcal{P}_{H,I}$ and evaluate it ▷ Arbitrary selection of an option
  Receive corresponding reward Y
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{(H, I)\}$ ▷ Extend the tree
  for all $(h, i) \in P$ **do** ▷ Update the statistics T and $\hat{\mu}$ stored in the path
    $T_{h,i} \leftarrow T_{h,i} + 1$ ▷ Increment the counter of node (h, i)
    $\hat{\mu}_{h,i} \leftarrow \left(1 - 1/T_{h,i}\right)\hat{\mu}_{h,i} + Y/T_{h,i}$ ▷ Update the mean $\hat{\mu}_{h,i}$ of node (h, i)
  end for
  for all $(h, i) \in \mathcal{T}$ **do** ▷ Update the statistics U stored in the tree
    $U_{h,i} \leftarrow \hat{\mu}_{h,i} + \sqrt{(2 \log n)/T_{h,i}} + \nu_1 \rho^h$ ▷ Update the U–value of node (h, i)
  end for
  $B_{H+1,2I-1} \leftarrow +\infty$ ▷ B–values of the children of the new leaf
  $B_{H+1,2I} \leftarrow +\infty$
  $\mathcal{T}' \leftarrow \mathcal{T}$ ▷ Local copy of the current tree $\mathcal{T}$
  while $\mathcal{T}' \neq \{(0, 1)\}$ **do** ▷ Backward computation of the B–values
    $(h, i) \leftarrow \text{LEAF}(\mathcal{T}')$ ▷ Take any remaining leaf
    $B_{h,i} \leftarrow \min\Big\{U_{h,i}, \max\{B_{h+1,2i-1}, B_{h+1,2i}\}\Big\}$ ▷ Backward computation
    $\mathcal{T}' \leftarrow \mathcal{T}' \setminus \{(h, i)\}$ ▷ Drop updated leaf (h, i)
  end while
end for

SOO (Simultaneous Optimistic Optimization)

- SOO algorithm uses optimistic estimates of the function values.
- Assumes that the function values at unexplored points are higher than the observed values
- Quickly explores the search space and converges to a good solution



SOO at work

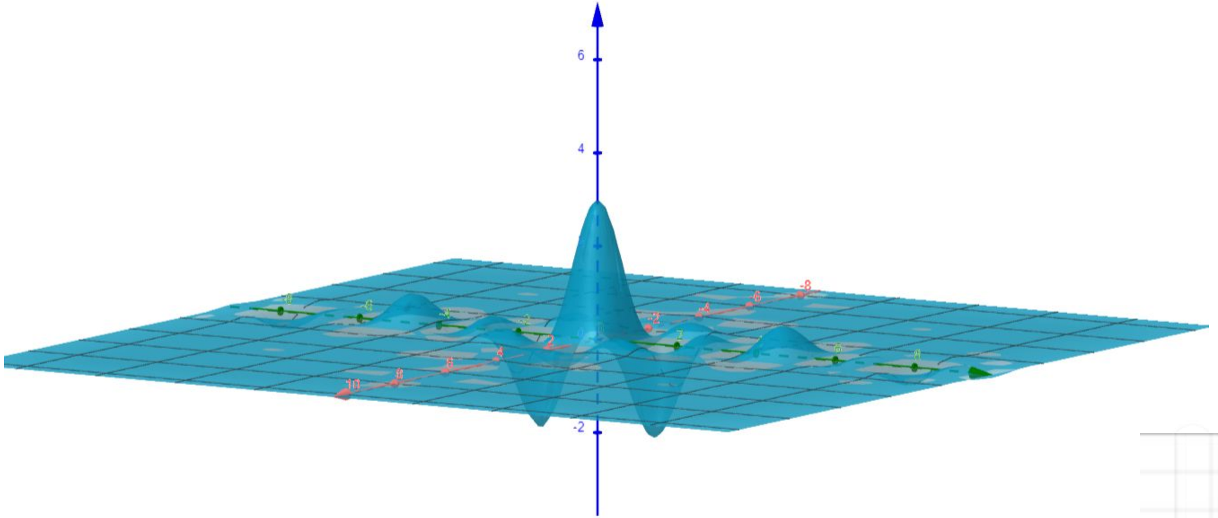
Source: [Fig. 1, 1]

Results: HOO

Functions	HOO (Time (in s) | Range of Solution | Objective Value)	Actual Value
sin(x)	2.52 | [(-1.5712890625,-1.5703125)] | -0.99	-π/2 | -1
(sin(13x)sin(27x) + 1)/2	2.42 | [(-1.33642578125, -1.3359375)] | 0.997	-1.336 | 1
cos(x) + cos(y)	2.52 | [(1.99609375, 2), (-2, -1.9921875)] | -0.82	[π π] | -2
sin(x)sin(y)sin(3x)cos(2y)/x^2y	2.51 | [(0.0, 0.0078125), (1.4296875, 1.4375)] | -1.99	-2

Results: SOO

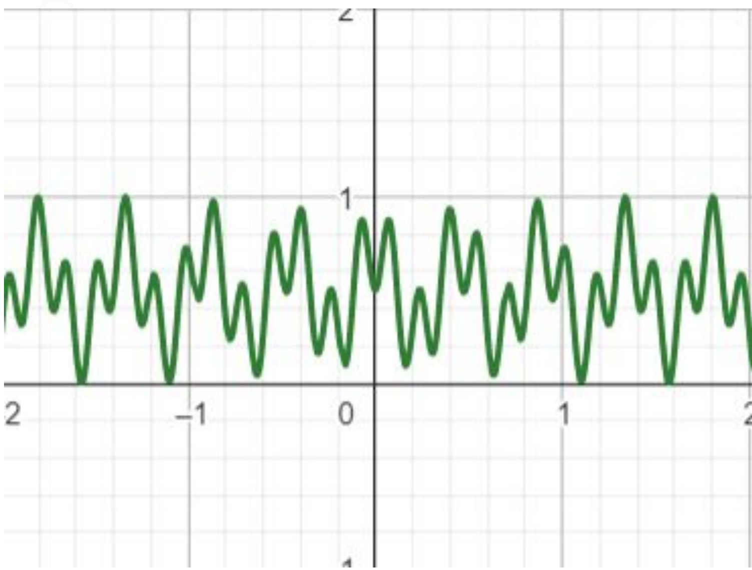
<table>
<tr><th>Functions</th><th>SOO (Time (in s)|
Best Solution | Objective Value)</th><th>Actual Value</th></tr>
<tr><td>sin(x)</td><td>0.41 | -1.5708 | -0.99</td><td>-π/2 | -1</td></tr>
<tr><td>(sin(13x)sin(27x) + 1)/2</td><td>0.82 | -1.5707 | 1.29*10^(-9)</td><td>-π/2 | 0</td></tr>
<tr><td>cos(x) + cos(y)</td><td>0.74 | [3.1491 3.1405] | -1.99</td><td>[π π] | -2</td></tr>
<tr><td>sin(x)sin(y)sin(3x)cos(2y)/x^2y</td><td>0.59 | [-0.0021 -1.4308] | -1.99</td><td>-2</td></tr>
</table>



Pictorial representation of

$$\sin(x)\ \sin(y)\ \sin(3\,x)\ \frac{\cos(2\,y)}{x^2\,y}$$

Pictorial representation of $\dfrac{\sin(13\,x)\ \sin(27\,x)+1}{2}$



Courtesy: Geogebra

Limitations & Future work

- Currently the sampling techniques employed don't take into account the nature of the function being optimized, we will explore other sampling methods too which indeed take into account the nature of the evaluations
- To test HOO on a wide variety of higher dimensional black box functions
- To optimize the current code
- To further refine the bounds obtained via HOO

References

[1] P. Preux, R. Munos and M. Valko, "Bandits attack function optimization," 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 2014, pp. 2245-2252, doi: 10.1109/CEC.2014.6900558.

[2] (n.d.). Bandit View on Noisy Optimization. CS769 IITB. https://www.cse.iitb.ac.in/~ganesh/cs769/notes/readingAssignment/BanditViewOnNoisyOptimisation.pdf

[3] Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. 2011. X-Armed Bandits. J. Mach. Learn. Res. 12, null (2/1/2011), 1655–1695.

Thank You!