Numerical exploration-exploitation tradeoff for large scale function optimization

Rémi Munos

INRIA Lille - Nord Europe
Currently on leave at MSR-NE
http://researchers.lille.inria.fr/~munos/

LSOLDM 2013, Cumberland Lodge
Initial motivation

Monte-Carlo Tree Search in computer-go

MCTS in Crazy-Stone (Rémi Coulom, 2005)

Idea: use bandits at each node of the tree search.
UCB applied to Trees

Uses Upper Confidence Bound (UCB) algorithm [Auer et al., 2002] at each node of the tree

\[ B_j \overset{\text{def}}{=} X_{j,n_j} + \sqrt{\frac{2 \log(n_i)}{n_j}}. \]

**Intuition:**
- Explore first the most promising branches
- Average converges to max
  - Adaptive Multistage Sampling (AMS) algorithm [Chang, Fu, Hu, Marcus, 2005]
  - UCB applied to Trees (UCT) [Kocsis and Szepesvári, 2006]
The MoGo program [Gelly et al., 2006]

Use hierarchy of UCB bandits (UCT) [Kocsis and Szepesvári, 2006]

Features:
- Monte-Carlo evaluation
- Asymmetric tree expansion
- Anytime algo
- Use of features

Very strong program!

MCTS and UCT very successful
No finite-time guarantee for UCT

**Problem:** at each node, the rewards are not i.i.d.
Consider the tree:

The left branches seem better than right branches, thus are explored for a **very** long time before the optimal leaf is eventually reached.

The regret is disastrous:

\[ \mathbb{E} R_n = \Omega(\exp(\exp(\ldots \exp(1) \ldots ))) + O(\log(n)). \]

See [Coquelin and Munos, 2007]
Optimism in the face of uncertainty

“Numerical exploration-exploitation tradeoff”: perform search in simulation using finite numerical resources.

Outline:

- Optimistic optimization of a deterministic Lipschitz functions
- 4 extensions:
  - Locally smooth functions,
  - Tractable algorithm
  - Unknown smoothness,
  - Noisy evaluations
Optimization of a deterministic Lipschitz function

**Problem:** Find online the maximum of $f : X \to R$, assumed to be Lipschitz:

$$|f(x) - f(y)| \leq \ell(x, y).$$

**Protocol:**
- For each time step $t = 1, 2, \ldots, n$ select a state $x_t \in X$
- Observe $f(x_t)$
- Return a state $x(n)$

**Loss:**

$$r_n = f^* - f(x(n)),$$

where $f^* = \sup_{x \in X} f(x)$. 
Lipschitz property → the evaluation of $f$ at $x_t$ provides a first upper-bound on $f$. 

$Lipschitz optimization$  $Local smoothness$  $Hierarchical partitioning$  $Unknown smoothness$  $Noisy evaluations$
Example in 1d (continued)

New point $\rightarrow$ refined upper-bound on $f$. 
Question: where should one sample the next point?
Answer: select the point with highest upper bound!
“Optimism in the face of (partial observation) uncertainty”
Several issues

1. Lipschitz assumption is too strong
2. Finding the optimum of the upper-bounding function may be hard!
3. What if we don’t know the metric $\ell$?
4. How to handle noise?
Local smoothness property

Assumption: \( f \) is "locally smooth" around its max. w.r.t. \( \ell \) where \( \ell \) is a semi-metric (symmetric, and \( \ell(x, y) = 0 \iff x = y \): For all \( x \in \mathcal{X} \),

\[
    f(x^*) - f(x) \leq \ell(x, x^*).
\]
Local smoothness is enough!

Optimistic principle only requires:
- a true bound at the maximum
- the bounds gets refined when adding more points
Deterministic Optimistic Optimization (DOO) builds a hierarchical partitioning of the space where cells are refined according to their upper bounds.

- For \( t = 1 \) to \( n \),
  - Define an upper bound for each cell:
    \[
    B_i = f(x_i) + \text{diam}_\ell(X_i)
    \]
  - Select the cell with highest bound
    \[
    I_t = \arg\max_i B_i.
    \]
  - Expand \( I_t \): refine the grid and evaluate \( f \) in children cells
  - Return \( x(n) \overset{\text{def}}{=} \arg\max\{x_t\}_{1 \leq t \leq n} f(x_t) \)
Near-optimality dimension

Define the **near-optimality dimension** of $f$ as the smallest $d \geq 0$ such that $\exists C, \forall \epsilon$, the set of $\epsilon$-optimal states

$$X_\epsilon \overset{\text{def}}{=} \{ x \in X, f(x) \geq f^* - \epsilon \}$$

can be covered by $C\epsilon^{-d}$ $\ell$-balls of radius $\epsilon$. 

---

Lipschitz optimization  Local smoothness  Hierarchical partitioning  Unknown smoothness  Noisy evaluations
Example 1:
Assume the function is piecewise linear at its maximum:

\[ f(x^*) - f(x) = \Theta(||x^* - x||). \]

Using \( \ell(x, y) = ||x - y|| \), it takes \( O(\epsilon^0) \) balls of radius \( \epsilon \) to cover \( X_\epsilon \). Thus \( d = 0 \).
Example 2:

Assume the function is locally quadratic around its maximum:

\[ f(x^*) - f(x) = \Theta(||x^* - x||^2). \]

For \( \ell(x, y) = ||x - y|| \), it takes \( O(\epsilon^{-D/2}) \) balls of radius \( \epsilon \) to cover \( X_\epsilon \) (of size \( O(\epsilon^{D/2}) \)). Thus \( d = D/2 \).
Example 2:

Assume the function is locally quadratic around its maximum:

\[ f(x^*) - f(x) = \Theta(||x^* - x||^2) \]

For \( \ell(x, y) = ||x - y||^2 \), it takes \( O(\epsilon^0) \) \( \ell \)-balls of radius \( \epsilon \) to cover \( X_\epsilon \). Thus \( d = 0 \).
Example 3:

Assume the function has a square-root behavior around its maximum:

\[ f(x^*) - f(x) = \Theta(||x^* - x||^{1/2}) \]

For \( \ell(x, y) = ||x - y||^{1/2} \) we have \( d = 0 \).
Example 4:

Assume $\mathcal{X} = [0, 1]^D$ and $f$ is locally equivalent to a polynomial of degree $\alpha > 0$ around its maximum (i.e. $f$ is $\alpha$-smooth):

$$f(x^*) - f(x) = \Theta(||x^* - x||^\alpha)$$

Consider the semi-metric $\ell(x, y) = ||x - y||^\beta$, for some $\beta > 0$.

- If $\alpha = \beta$, then $d = 0$.
- If $\alpha > \beta$, then $d = D(\frac{1}{\beta} - \frac{1}{\alpha}) > 0$.
- If $\alpha < \beta$, then the function is not locally smooth wrt $\ell$. 
Lipschitz optimization  Local smoothness  Hierarchical partitioning  Unknown smoothness  Noisy evaluations

Analysis of DOO (deterministic case)

Assume that the $\ell$-diameters of the nodes of depth $h$ decrease exponentially fast with $h$ (i.e., $\text{diam}(h) = c\gamma^h$, for $c > 0 \gamma < 1$).

Example: $\mathcal{X} = [0, 1]^D$ and $\ell(x, y) = \|x - y\|^\beta$ for some $\beta > 0$.

**Theorem 1.**
The loss of DOO is

$$r_n = \begin{cases} \left(\frac{c}{1-\gamma^d}\right)^{1/d} n^{-1/d} & \text{for } d > 0, \\ c\gamma^{n/C-1} & \text{for } d = 0. \end{cases}$$

(Remember that $r_n \overset{\text{def}}{=} f(x^*) - f(x(n))$).
About the local smoothness assumption

Assume $f$ satisfies $f(x^*) - f(x) = \Theta(||x^* - x||^\alpha)$.

Use DOO with the semi-metric $\ell(x, y) = ||x - y||^\beta$:

- If $\alpha = \beta$, then $d = 0$: the true “local smoothness” of the function is known, and exponential rate is achieved.
- If $\alpha > \beta$, then $d = D\left(\frac{1}{\beta} - \frac{1}{\alpha}\right) > 0$: we under-estimate the smoothness, which causes more exploration than needed.
- If $\alpha < \beta$: We over-estimate the true smoothness and DOO may fail to find the global optimum.

**DOO heavily depends on our knowledge of the true local smoothness.**
Experiments [1]

\( f(x) = \frac{1}{2}(\sin(13x) \sin(27x) + 1) \) satisfies the local smoothness assumption with

- \( \ell_1(x, y) = 14|x - y| \) (i.e., \( f \) is globally Lipschitz), \( d = 1/2 \)
- \( \ell_2(x, y) = 222|x - y|^2 \) (i.e., \( f \) is locally quadratic), \( d = 0 \)
Experiments [2]

Using $\ell_1(x, y) = 14|x - y|$ (i.e., $f$ is globally Lipschitz). $n = 150$. 

The trees $T_n$ built by DOO after $n = 150$ evaluations.
Experiments [3]

Using $\ell_2(x, y) = 222|x - y|^2$ (i.e., $f$ is locally quadratic). $n = 150$. 

The trees $T_n$ built by DOO after $n = 150$ evaluations.
Experiments [4]

<table>
<thead>
<tr>
<th>$n$</th>
<th>uniform grid</th>
<th>DOO with $\ell_1$ ($d = 1/2$)</th>
<th>DOO with $\ell_2$ ($d = 0$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>$1.25 \times 10^{-2}$</td>
<td>$2.53 \times 10^{-5}$</td>
<td>$1.20 \times 10^{-2}$</td>
</tr>
<tr>
<td>100</td>
<td>$8.31 \times 10^{-3}$</td>
<td>$2.53 \times 10^{-5}$</td>
<td>$1.67 \times 10^{-7}$</td>
</tr>
<tr>
<td>150</td>
<td>$9.72 \times 10^{-3}$</td>
<td>$4.93 \times 10^{-6}$</td>
<td>$4.44 \times 10^{-16}$</td>
</tr>
</tbody>
</table>

Loss $r_n$ for different values of $n$ for a uniform grid and DOO with the two semi-metric $\ell_1$ and $\ell_2$. 
What if the smoothness is unknown?

Previous algorithms heavily rely on the knowledge or the local smoothness of the function (i.e. knowledge of the best metric).

**Question:** When the smoothness is unknown, is it possible to implement the optimistic principle for function optimization?
DIRECT algorithm [Jones et al., 1993]

Assumes $f$ is Lipschitz but the Lipschitz constant $L$ is unknown.

The DIRECT algorithm expands simultaneously all nodes that may potentially contain the maximum for some value of $L$.

Be optimistic for all $L$
Illustration of DIRECT

The sin function and its upper bound for $L = 2$. 
Illustration of DIRECT

The sin function and its upper bound for $L = 1/2$. 
Limitations of DIRECT

- No finite-time analysis (only the consistency property $\lim_{n \to \infty} r_n = 0$ in [Finkel and Kelley, 2004])
- Global Lipschitz assumption is too strong!

We want to extend to

- any function **locally smooth** w.r.t. $\ell$,
- for **any semi-metric** $\ell$
- and provide performance guarantees.
Simultaneous Optimistic Optimization (SOO)

[Munos, 2011]

- Expand several leaves simultaneously
- SOO expands at most one leaf per depth
- SOO expands a leaf only if its value is larger that the value of all leaves of same or lower depths.
- At round $t$, SOO does not expand leaves with depth larger than $h_{\text{max}}(t)$

Be optimistic at all scales
**SOO algorithm**

**Input:** the maximum depth function $t \mapsto h_{\text{max}}(t)$

**Initialization:** $\mathcal{T}_1 = \{(0, 0)\}$ (root node). Set $t = 1$.

**while** True **do**

Set $\nu_{\text{max}} = -\infty$.

**for** $h = 0$ to min(depth($\mathcal{T}_t$), $h_{\text{max}}(t)$) **do**

Select the leaf $(h, j) \in \mathcal{L}_t$ of depth $h$ with max $f(x_{h,j})$ value

**if** $f(x_{h,i}) > \nu_{\text{max}}$ **then**

Expand the node $(h, i)$, Set $\nu_{\text{max}} = f(x_{h,i})$, Set $t = t + 1$

**if** $t = n$ **then** return $x(n) = \arg \max_{(h,i) \in \mathcal{T}_n} x_{h,i}$

**end if**

**end for**

**end while.**
Lipschitz optimization
Local smoothness
Hierarchical partitioning
Unknown smoothness
Noisy evaluations
Lipschitz optimization
Local smoothness
Hierarchical partitioning
Unknown smoothness
Noisy evaluations
<table>
<thead>
<tr>
<th>Lipschitz optimization</th>
<th>Local smoothness</th>
<th>Hierarchical partitioning</th>
<th>Unknown smoothness</th>
<th>Noisy evaluations</th>
</tr>
</thead>
</table>

![Graph with points and line segments](image-url)
Lipschitz optimization
Local smoothness
Hierarchical partitioning
Unknown smoothness
Noisy evaluations
<table>
<thead>
<tr>
<th>Lipschitz optimization</th>
<th>Local smoothness</th>
<th>Hierarchical partitioning</th>
<th>Unknown smoothness</th>
<th>Noisy evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="Diagram" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lipschitz optimization</td>
<td>Local smoothness</td>
<td>Hierarchical partitioning</td>
<td>Unknown smoothness</td>
<td>Noisy evaluations</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------</td>
<td>--------------------------</td>
<td>--------------------</td>
<td>-------------------</td>
</tr>
</tbody>
</table>

![Diagram](image-url)
<table>
<thead>
<tr>
<th>Lipschitz optimization</th>
<th>Local smoothness</th>
<th>Hierarchical partitioning</th>
<th>Unknown smoothness</th>
<th>Noisy evaluations</th>
</tr>
</thead>
</table>

![Diagram with blue dots and circles]
Performance of SOO

**Theorem 2.**
For any semi-metric $\ell$ such that

- $f$ is locally smooth w.r.t. $\ell$
- The $\ell$-diameter of cells of depth $h$ is $c\gamma^h$
- The near-optimality dimension of $f$ w.r.t. $\ell$ is $d = 0$,

by choosing $h_{\text{max}}(n) = \sqrt{n}$, the expected loss of SOO is

$$r_n \leq c\gamma^{\sqrt{n}/C-1}$$

In the case $d > 0$ a similar statement holds with $\mathbb{E}r_n = \tilde{O}(n^{-1/d})$. 
Performance of SOO

Remarks:

• Since the algorithm does not depend on $\ell$, the analysis holds for the best possible choice of the semi-metric $\ell$ satisfying the assumptions.

• **SOO does almost as well as DOO optimally fitted** (thus “adapts” to the unknown local smoothness of $f$).
Again for the function $f(x) = (\sin(13x) \sin(27x) + 1)/2$ we have:

<table>
<thead>
<tr>
<th>$n$</th>
<th>uniform grid</th>
<th>DOO with $\ell_1$</th>
<th>DOO with $\ell_2$</th>
<th>SOO</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>$1.25 \times 10^{-2}$</td>
<td>$2.53 \times 10^{-5}$</td>
<td>$1.20 \times 10^{-2}$</td>
<td>$3.56 \times 10^{-4}$</td>
</tr>
<tr>
<td>100</td>
<td>$8.31 \times 10^{-3}$</td>
<td>$2.53 \times 10^{-5}$</td>
<td>$1.67 \times 10^{-7}$</td>
<td>$5.90 \times 10^{-7}$</td>
</tr>
<tr>
<td>150</td>
<td>$9.72 \times 10^{-3}$</td>
<td>$4.93 \times 10^{-6}$</td>
<td>$4.44 \times 10^{-16}$</td>
<td>$1.92 \times 10^{-10}$</td>
</tr>
</tbody>
</table>
The case $d = 0$ is non-trivial!

Example:

- $f$ is locally $\alpha$-smooth around its maximum:

  $$f(x^*) - f(x) = \Theta(\|x^* - x\|^{\alpha}),$$

  for some $\alpha > 0$.

- SOO algorithm does not require the knowledge of $\ell$.

- Using $\ell(x, y) = \|x - y\|^{\alpha}$ in the analysis, all assumptions are satisfied (with $\gamma = 3^{-\alpha/D}$ and $d = 0$, thus the loss of SOO is $r_n = O(3^{-\sqrt{n}\alpha}/(CD))$ (stretched-exponential loss).

- This is almost as good as DOO optimally fitted!

  (Extends to the case $f(x^*) - f(x) \approx \sum_{i=1}^{D} c_i |x_i^* - x_i|^{\alpha_i}$)
The case \( d = 0 \)

More generally, any function whose upper- and lower envelopes around \( x^* \) have the same shape: \( \exists c > 0 \) and \( \eta > 0 \), such that

\[
\min(\eta, c\ell(x, x^*)) \leq f(x^*) - f(x) \leq \ell(x, x^*), \quad \text{for all } x \in \mathcal{X}.
\]

has a near-optimality \( d = 0 \) (w.r.t. the metric \( \ell \)).
Example of functions for which $d = 0$

\[ \ell(x, y) = c\|x - y\|^2 \]
Example of functions with $d = 0$

\[
\ell(x, y) = c\|x - y\|^{1/2}
\]
\[ d = 0? \]

\[ \ell(x, y) = c\|x - y\|^{1/2} \]
\( d > 0 \)

\[
f(x) = 1 - \sqrt{x} + (-x^2 + \sqrt{x}) \star (\sin(1/x^2) + 1)/2
\]

The lower-envelope is of order 1/2 whereas the upper one is of order 2. We deduce that \( d = 3/2 \) and \( r_n = O(n^{-2/3}) \).
SOO versus DIRECT

- **SOO is much more general than DIRECT**: the function is only locally smooth and the space is semi-metric.
- **Finite-time analysis of SOO**
- **SOO is a rank-based algorithm**: any transformation of the values while preserving their rank will not change anything in the algorithm. Thus extends to the optimization of function given as pair-wise comparisons.
How to handle noise?

The evaluation of $f$ at $x_t$ is perturbed by noise:

$$y_t = f(x_t) + \epsilon_t,$$

with $E[\epsilon_t | x_t] = 0$. 

\[ f(x_t) \]

\[ f^* \]
Stochastic SOO (StoSOO)

Extends SOO to stochastic evaluations:

- Select the cells $X_i$ (at most one per depth) according to SOO based on the UCBs:

$$\hat{\mu}_{i,t} + c\sqrt{\frac{\log n}{T_i(t)}},$$

and get one more value $y_t = f(x_i) + \epsilon_t$ of $f$ at $x_i$.

- If $T_i(t) \geq k$, then split the cell $X_i$.

Remark: This really looks like UCT, except that

- several cells are selected at each round,
- a cell is split only after observing $k$ values.
Performance of StoSOO

**Theorem 3 (Valko et al., 2013).**

For any semi-metric $\ell$ such that

- $f$ is locally smooth w.r.t. $\ell$
- The $\ell$-diameters of the cells decrease exponentially fast with their depth,
- The near-optimality dimension of $f$ w.r.t. $\ell$ is $d = 0$,

by choosing $k = \frac{n}{(\log n)^3}$, $h_{\text{max}}(n) = (\log n)^{3/2}$, the expected loss of StoSOO is

$$\mathbb{E}r_n = O\left(\frac{(\log n)^2}{\sqrt{n}}\right).$$

This is almost as good as HOO [Bubeck et al., 2011] and Zooming [Kleinberg et al., 2008] optimally fitted! Complementary to the adaptive-treed bandits of [Bull, 2013].
Range of application

All illustrations are in Euclidean spaces $[0, 1]^D$ only.

But there are many other semi-metric spaces...

- Trees (games, ...)
- Graphs (social networks, ...),
- Combinatorial spaces (shortest paths problems, ...)
- Other structured spaces (policies in MDPs, ...)

We only require:

- the search space $\mathcal{X}$ to be equipped with a semi-metric $\ell$,
- a nested (hierarchical) partitioning of the space,
- $f$ to satisfy a local smoothness property w.r.t. $\ell$,
- $\ell$ may or may not be known.
Conclusions

Provide a measure of the complexity of optimization.

This multi-scale optimistic optimization

- provides an efficient exploration of the search space by exploring the most promising areas first
- provides a natural transition from global to local search
- Performance depends on the “smoothness” of the function around the maximum w.r.t. some metric,
  - and a measure of the quantity of near-optimal solutions,
  - and our knowledge or not of this smoothness.
Thanks !!!

See the review paper

*From bandits to Monte-Carlo Tree Search: The optimistic principle applied to optimization and planning.*

from my web page:

http://chercheurs.lille.inria.fr/~munos/