Sequential Optimization and Computer Experiments

An introduction

Aurélien Garivier

March 23^{rd}, 2016

Institut de Mathématiques de Toulouse
LabeX CIMI
Université Paul Sabatier

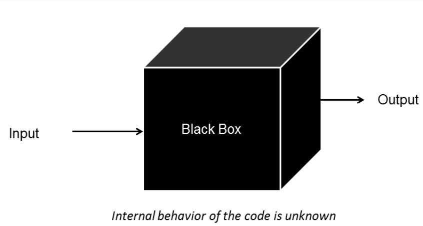
Table of contents

Framework

Given an input $x \in \mathcal{X}$, a (complex) codes returns

$$F(x, U) = f(x) + \eta$$

where U is an independent $\mathcal{U}[0, 1]$ r.v. and $\mathbb{E}[\eta] = 0$. Possibly, $\eta = 0$.



Source: *freesourcecode.net.*

Goal: maximize f

using a sequential choice of inputs.

Examples:

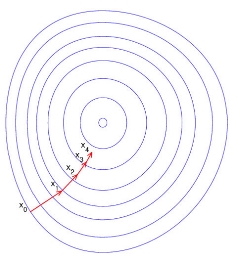
- Numerical Dosimetry (foetus exposure to Radio Frequency Electromagnetic Fields) - Jalla et al., Mascotnum 2013
- Traffic Optimization (find the shortest path from A to B)

Sequential Optimizing Without Noise: Some Ideas

Methods not mentionned here

Gradient descent

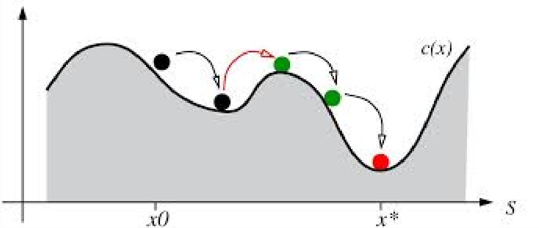
Here: search for global optimum, no convexity hypothesis



Source: www.d.umn.edu/˜deoka001/

Simulated annealing

slowly lower the temperature of a hot material, minimizing the system energy



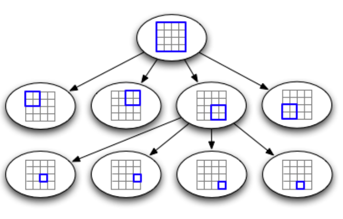
Source: freesourcecode.net

Genetic Algorithms, Cutting Plane methods, Sum of Squares,...

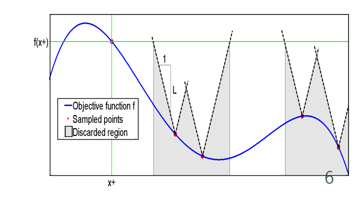
The Branch-and-Bound Paradigm [Munos, 2014, de Freitas et al., 2012]

Also used for discrete and combinatorial optimization problems

- **Branching** = hierarchical partitioning (recursive splitting) of $\mathcal{X}$
- Each cell C has a representative $x_C \in C$
- Assumption: possibility to compute an **upper-bound** of f on each cell (using the regularity of f)
- Start with 1 active cell = $\mathcal{X}$ and $\hat{x} = x_{\mathcal{X}}$
- At each iteration:
  - Pick an active cell C
  - $f(x_C) > f(\hat{x})$, update $\hat{x} := x_C$
  - Split C into sub-cells and desactivate C
  - Set all sub-cells with upper-bounds larger that $f(\hat{x})$ to be active



Source: veendeta.wordpress.com



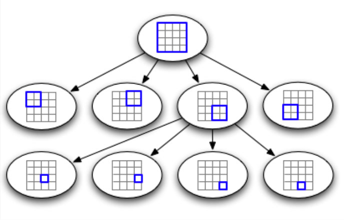
The SOO algorithm Munos [2011]

SOO = Simultaneous Optimistic Optimization

Requires a multi-scale decomposition of $\mathcal{X}$:
$\forall h \geq 0$,

$$\mathcal{X} = \bigcup_{i=1}^{N_h} C_{h,i} \; .$$

Ex: binary splitting.



Source: veendeta.wordpress.com

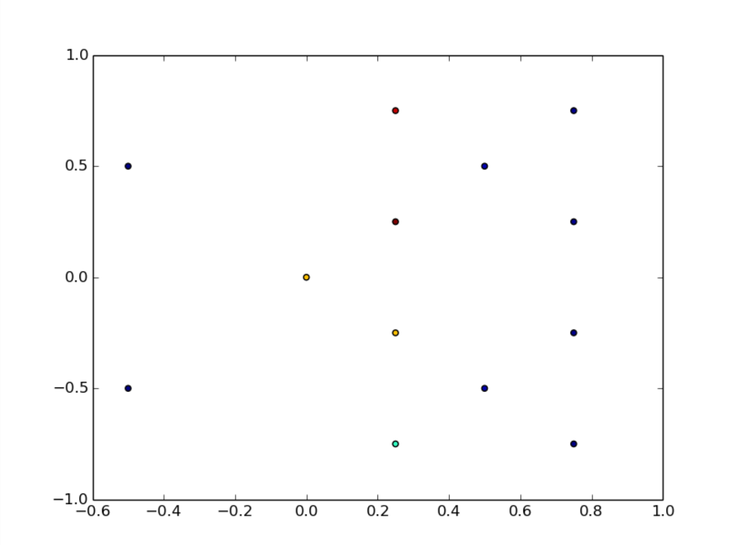
SOO

FOR r=1..R
  FOR every non-empty depth d
    SPLIT the cell $C_{h,i}$ of depth d with highest $f(x)$

No need to know the (possibly anisotropic) regularity of f !

SOO: an Example

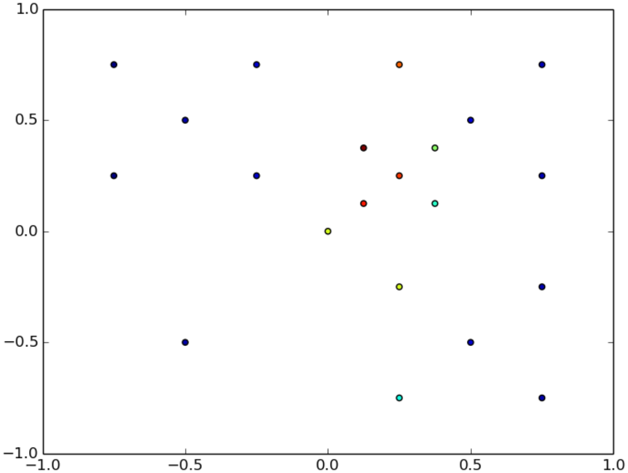
$$f(x, y) = (x - c_1)^2 - 0.05|y - c_2|$$



$n = 10$

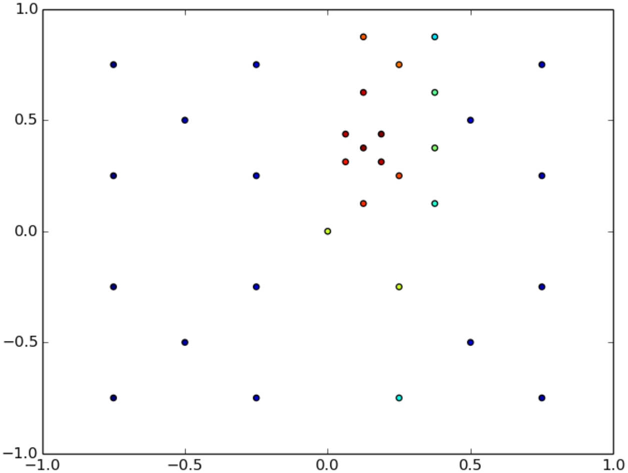
SOO: an Example

$$f(x, y) = (x - c_1)^2 - 0.05|y - c_2|$$



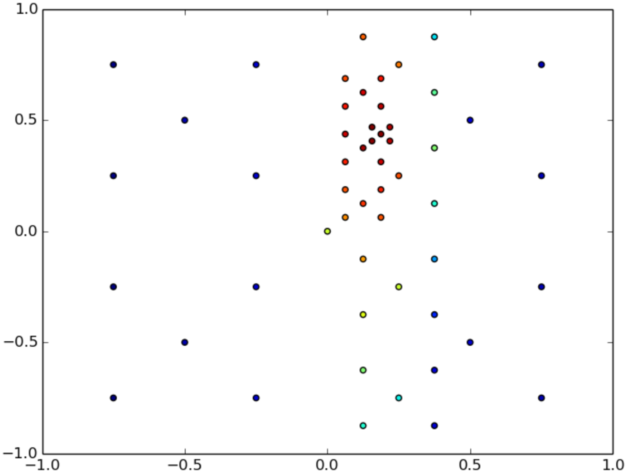
SOO: an Example

$$f(x, y) = (x - c_1)^2 - 0.05|y - c_2|$$



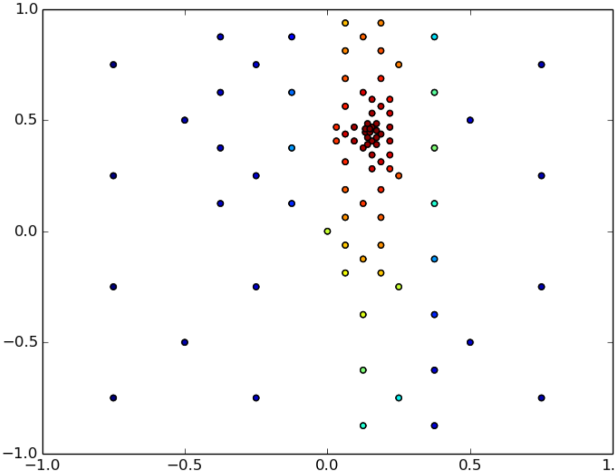
SOO: an Example

$$f(x, y) = (x - c_1)^2 - 0.05|y - c_2|$$



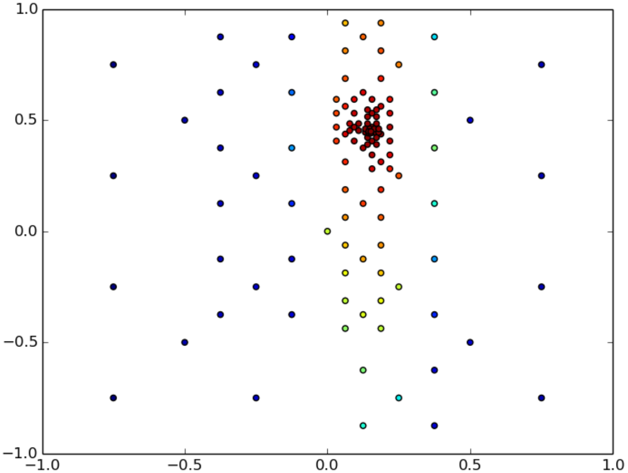
SOO: an Example

$$f(x, y) = (x - c_1)^2 - 0.05|y - c_2|$$



SOO: an Example

$$f(x, y) = (x - c_1)^2 - 0.05|y - c_2|$$



Analysis: Near-Optimality Dimension

For every $\epsilon > 0$, let

$$\mathcal{X}_\epsilon = \left\{x \in \mathcal{X} : f(x) \geq f^* - \epsilon\right\} .$$

Definition: Near-Optimality Dimension

The near-optimality dimension of f is the smallest $d \geq 0$ such that there exists $C > 0$ for which, for all $\epsilon > 0$, the maximal number of disjoint balls of radius ϵ with center in $\mathcal{X}_\epsilon$ is less than $C\epsilon^{-d}$.

Speed of convergence of SOO [Valko et al., 2013]

Theorem: If $\delta(h) = c\gamma^h$ and if the near-optimality dimension of f is $d = 0$, then

$$f^* - f(\hat{x}_t) = O\big(\gamma^t\big) \ .$$

If the near-optimality dimension of f is $d > 0$, then

$$f^* - f(\hat{x}_t) = O\left(\frac{1}{t^{1/d}}\right) \ .$$

Idea of the proof:

For every scale h let

$$\delta(h) = \max_i \sup_{x,x' \in C_{h,i}} f(x) - f(x') \qquad \text{and} \qquad I_h = \big\{C_{h,i} : f(x_{h,i}) + \delta(h) \geq f^*\big\}$$

At every level h, the number of cells splitted before the one containing x^* is at most $|I_h| \leq C\delta(h)^{-d}$.

Thus, after t splits, the algorithm has splitted a cell containing x^* of level at least h_t^* such that $C\sum_{l=0}^{h_t^*} \delta(l)^{-d} \geq t$.

Kriging: Gaussian Process Regression

Kriging

Bayesian model: f is drawn from a random distribution.

Gaussian Process: for every t and every $x_1, \ldots, x_t \in \mathcal{X}$,

$$\begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_t) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, K_t = \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) & \ldots & k(x_1, x_t) \\ k(x_2, x_1) & k(x_2, x_2) & \ldots & k(x_2, x_t) \\ \vdots & \vdots & \vdots & \vdots \\ k(x_t, x_1) & k(x_t, x_2) & \ldots & k(x_t, x_t) \end{pmatrix} \right)$$

where $k : \mathcal{X} \times \mathcal{X} \to R$ is a covariance function.

Possibility to incorporate Gaussian noise: $\vec{Y}_t = \vec{f}_t + \vec{\epsilon}_t$.

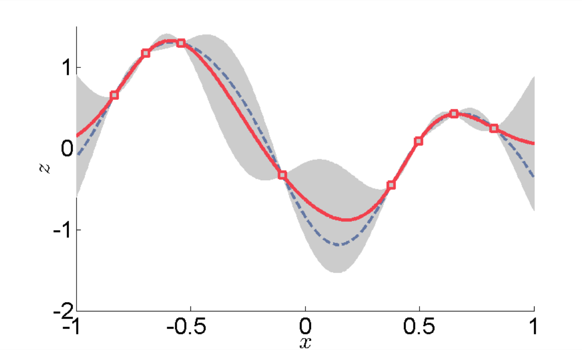
Why kriging?

Conditionally on $\mathcal{F}_t$, f is still a Gaussian process:

$$\mathcal{L}(f|\mathcal{F}_t) = GP\Big(\mu_t : u \mapsto k_t(u)^T K_t^{-1} \vec{Y}_t, k_t : u, v \mapsto k(u,v) - \vec{k}_t(u)^T K_t^{-1} \vec{k}_t(v)\Big)$$

where

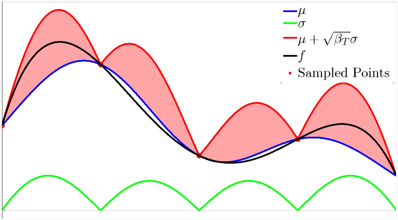
$$\vec{k}_t(u) = \begin{pmatrix} k(u, x_1) \\ k(u, x_2) \\ \vdots \\ k(u, x_t) \end{pmatrix} .$$



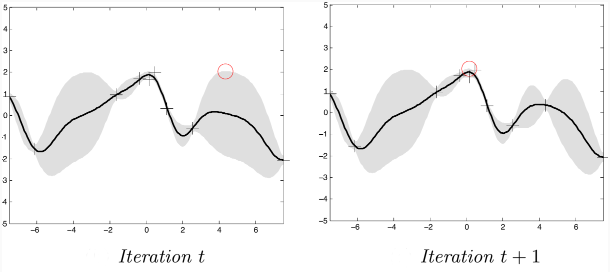
Source: wikipedia

The GP-UCB Algorithm [Srinivas et al., 2012]

- Initialization: space-filling (LHS)
- Iteration t:
    - For every $x \in \mathcal{X}$, compute $u(x)$ = quantile of $f(x)$ conditionally on $\mathcal{F}_{t-1}$ of level $1 - 1/t$
    - Choose $X_t = \operatorname{argmax}_{x \in \mathcal{X}} u(x)$
    - Observe $Y_t = F(X_t, U_t)$



Source: de Freitas et al. [2012]



Source: Srinivas et al. [2012]

GP-UCB: Convergence

Two kinds of results:

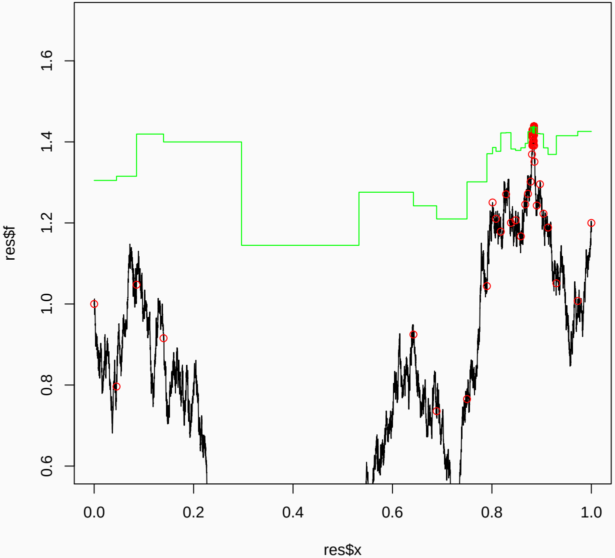
- If f is really **drawn from the Gaussian Process**: for the Gaussian kernel the cumulated regret is bounded as

$$\mathbb{E}\left[\sum_{t=1}^{T} f^* - f(X_t)\right] = O\Big(\sqrt{T}\big(\log(T)\big)^{\frac{d+1}{2}}\Big) .$$

- If f has a **small norm in the RKHS** corresponding to the kernel k (= regularity condition), similar results

Also Expected Improvement (similar idea, slightly different criterion), see Vazquez and Bect [2010].

GP-UCB is not limited to smooth functions: BrownUCB



Optimizing in the presence of noise

Sequential Strategies

A strategy is a triple:

- **Sampling rule**: X_t is $\mathcal{F}_{t-1}$-measurable, where

$$Y_t = F(X_t, U_t) \text{ and } \mathcal{F}_t = \sigma(X_1, Y_1, \ldots, X_t, Y_t) \ .$$

- **Stopping rule**: the number of observations τ is a stopping time wrt $(\mathcal{F}_t)_t$.
- **Decision rule**: $\hat{x}$ is $\mathcal{F}_\tau$-measurable.

Objectives

At least three relevant goals:

- **Cumulated regret**: $\tau = T$, maximize $\mathbb{E}\big[\sum_{t=1}^{T} Y_t\big]$
  ex: clinical study protocol
- **Simple regret**: $\tau = T$, minimize $f^* - \mathbb{E}\big[f(\hat{x}_T)\big]$
- **PAC analysis**: among (ϵ, δ)-Probably Approximately Correct methods satisfying

$$\mathbb{P}\big(f(\hat{x}) \geq f^* - \epsilon\big) \geq 1 - \delta \;,$$

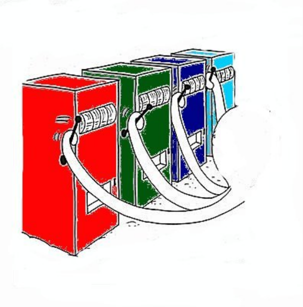
  minimize the sample complexity $\mathbb{E}[\tau]$.

The Bandit Approach

Advertisement: Perchet Course

In this workshop, 2 introductory lectures by Vianney Perchet:

- **Lecture 1** (thursay 9:00): the stochastic case (as above)
- **Lecture 2** (friday 9:00): adversarial case (game-theoretic/robust approach)



Here: optimization point of view.

Simplistic case: $\mathcal{X}$ finite

- $\mathcal{X} = \{1, \dots, K\}$
- $f \in [0, 1]^K$, no structure
- $F(x, U) \sim \mathcal{B}(f_x)$
- $(\epsilon, \delta)-$PAC analysis
- $\epsilon = 0$.

Ex: extreme clinical trials in dictatorship.

Not so simple!

Racing Algorithms: Successive Elimination [Even-Dar et al., 2006, Kaufmann and Kalyanakrishnan, 2013]

- At start, all arms are active;
- Then, repeatedly cycle thru active arms until only one arm is still active
- At the end of a cycle, eliminate arms with statistical evidence of sub-optimality: desactivate x if

$$\max \hat{f}(t) - \hat{f}_x(t) \geq 2\sqrt{\frac{\log(Kt^2/\delta)}{t}}$$

Theorem: Sucessive Elimination is $(0,\delta) - PAC$ and, with probability at least $1-\delta$,

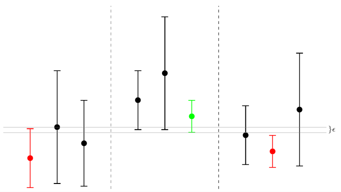
$$\tau_\delta = O\left(\sum_{x \neq x^*} \frac{\log \frac{K}{\delta \Delta_x}}{\Delta_x^2}\right)$$

where for all $x \in \{1, \ldots, K\}$, $\Delta_x = f^* - f(x)$.

The LUCB Algorithm [Kaufmann and Kalyanakrishnan, 2013]

See also Kalyanakrishnan et al. [2012], Gabillon et al. [2012], Jamieson et al. [2014].

- Maintain, at every step, a **lower-** and an **upper-confidence bound** for each arm;
- Successively **draw** the best empirical arm and the challenger with highest upper-confidence bound;
- **Stop** when, for some $x \in \mathcal{X}$, the lower bound on f_x is by ϵ of the highest upper-bound of the other arms.



Theorem: The sample complexity $\mathbb{E}[\tau]$ of LUCB (with adequate confidence bounds) is upper-bounded by $O\big(H_\epsilon \log(H_\epsilon/\delta)\big)$, where

$$H_\epsilon = \sum_x \frac{1}{(\Delta_x \vee \epsilon/2)^2} \ ,$$

$\Delta_{x^*} = f(x^*) - \max_{x \neq x^*} f(x)$ and, for $x \neq x^*$, $\Delta_x = f(x^*) - f(x)$.

Lower bound [Garivier and Kaufmann, 2016]

Let $\Sigma_K = \{\omega \in \mathbb{R}_+^k : \omega_1 + \cdots + \omega_K = 1\}$ and

$$\mathcal{A}(f) := \{g \in [0,1]^K : \operatorname{argmax} f \neq \operatorname{argmax} g\} ,$$

Theorem: For any δ-PAC strategy and any function $f \in [0,1]^K$,

$$\mathbb{E}[\tau_\delta] \geq T^*(f)\,\mathrm{kl}(\delta, 1-\delta),$$

where

$$T^*(f)^{-1} := \sup_{w \in \Sigma_K} \inf_{g \in \mathcal{A}(f)} \sum_{x=1}^{K} w_x\,\mathrm{kl}(f_x, g_x)$$

and $\mathrm{kl}(x,y) := x\log(x/y) + (1-x)\log((1-x)/(1-y))$.

Note: $\mathrm{kl}(\delta, 1-\delta) \approx \log(1/\delta)$ when $\delta \to 0$.

About the Computation of $T^*(f)$ and w^*

The proof shows that the maximizer $w^*(f)$ of

$$\sup_{w \in \boldsymbol{\Sigma}_K} \inf_{g \in \mathcal{A}(f)} \sum_{x=1}^{K} w_x d(f_x, g_x)$$

is the optimal proportion of arm draws.

Introducing

$$I_\alpha(y,z) := \alpha \mathrm{kl}\big(y, \alpha y + (1-\alpha)z\big) + (1-\alpha)\mathrm{kl}\big(z, \alpha y + (1-\alpha)z\big) \ ,$$

one can see that

$$T^*(f)^{-1} = \sup_{w \in \boldsymbol{\Sigma}_K} \min_{x \neq 1} \, (w_1 + w_x) I_{\frac{w_1}{w_1 + w_x}}(f_1, f_x) \ .$$

$T^*(f)$ and $w^*(f)$ can be computed by a succession of scalar equation resolutions, and one proves that:

1. For all $f \in [0,1]^K$ and all $1 \leq x \leq K$, $w_x^*(f) > 0$;
2. $w^*(f)$ is continuous in every f;
3. If $f_1 > f_2 \geq \cdots \geq f_K$, one has $w_2^*(f) \geq \cdots \geq w_K^*(f)$.

Tracking Strategy: Sampling Rule [Garivier and Kaufmann, 2016]

Let $\hat{f}(s)$ be the ML-estimator of f based on observations $Y_1, \ldots, Y_s$.

For every $\epsilon \in (0, 1/K]$, let $w^\epsilon(f)$ be a L^∞ projection of $w^*(f)$ on $\Sigma_K^\epsilon = \left\{(w_1, \ldots, w_K) \in [\epsilon, 1] : w_1 + \cdots + w_K = 1\right\}$. Let $\epsilon_t = (K^2 + t)^{-1/2}/2$ and

$$X_{t+1} \in \operatorname*{argmax}_{1 \le x \le K} \sum_{s=0}^{t} w_x^{\epsilon_s}(\hat{f}(s)) - \mathbb{1}\{X_s = x\} \ ,$$

Then for all $t \geq 1$ and $x \in \{1, \ldots, K\}$,
$N_x(t) = \sum_{s=0}^{t} \mathbb{1}\{X_s = x\}(t) \geq \sqrt{t + K^2} - 2K$ and

$$\max_{1 \le x \le K} \left| N_x(t) - \sum_{s=0}^{t-1} w_x^*(\hat{f}(s)) \right| \le K(1 + \sqrt{t}) \ .$$

Tracking Strategy: Stopping Rule [Garivier and Kaufmann, 2016]

For $x \in \{0,1\}^*$ let $p_\theta(x) = \theta^{\sum x}(1-\theta)^{\sum(1-x)}$.

Chernoff's Stopping Rule (see Chernoff [1959]): for $1 \leq x, z \leq K$ let

$$\begin{aligned} Z_{x,z}(t) &= \log \frac{\max_{f'_x \geq f'_z} p_{f'_x}\left(\underline{X}^x_{N_x(t)}\right) p_{f'_z}\left(\underline{X}^z_{N_z(t)}\right)}{\max_{f'_x \leq f'_z} p_{f'_x}\left(\underline{X}^x_{N_x(t)}\right) p_{f'_z}\left(\underline{X}^z_{N_z(t)}\right)} \\ &= N_x(t)\, d\big(\hat{f}_x(t), \hat{f}_{x,z}(t)\big) + N_z(t)\, d\big(\hat{f}_z(t), \hat{f}_{x,z}(t)\big) \end{aligned}$$

if $\hat{f}_x(t) \geq \hat{f}_z(t)$, and $Z_{x,z}(t) = -Z_{z,x}(t)$.

The stopping rule is defined by:

$$\tau_\delta = \inf\left\{ t \geq 1 : Z(t) := \max_{x \in \{1,\dots,K\}} \min_{z \in \{1,\dots,K\} \setminus \{x\}} Z_{x,z}(t) > \beta(t,\delta) \right\}$$

where $\beta(t,\delta)$ is the threshold to be tuned.

Optimality Result

Proposition: For every $\delta \in]0,1[$, whatever the sampling strategy, Chernoff's stopping rule with

$$\beta(t,\delta) = \log\left(\frac{2t(K-1)}{\delta}\right)$$

ensures that for all $f \in [0,1]^K$, $\mathbb{P}\left(\tau_\delta < \infty, \hat{x}_{\tau_\delta} \neq x^*\right) \leq \delta$.

Theorem: With the sampling rule and the stopping rule given above, $\tau_\delta < \infty$ a.s. and

$$\mathbb{P}\left(\limsup_{\delta \to 0} \frac{\tau_\delta}{\log(1/\delta)} \leq T^*(f)\right) = 1.$$

Bandit Algorithms for the Continuous Case

Kriging: GP-UCB Srinivas et al. [2012]

If $f \sim GP(0, k)$ and if for all $t \geq 1$

- $Y_t = f(X_t) + \epsilon_t$
- the noise ϵ_t is Gaussian

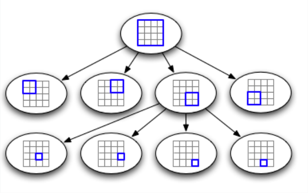
then $\vec{Y}_t$ is still a Gaussian vector.

$\implies$ the covariance kernel is modified, but one can still compute $\mathbb{E}[f(x)|\mathcal{F}_t]$ for every x, and apply the GP-UCB algorithm!

Works well in pratice, but limited guarantees.

Extensions to Continuous Spaces [Munos, 2014]

HOO maintains, for every cell $C_{h,i}$, two upper-confidence bounds (UCB) on $\max_{x \in C_{h,i}} f(x)$: $B_{h,i}$ based on all observations on the cell, and $U_{h,i} = \min\{B_{h,i}, U_{h+1,j}\}$ computed from the children j of $C_{h,i}$.



Source: *veendeta.wordpress.com*

The HOO Algorithm [Bubeck et al., 2011]

FOR t=1..T

  GO DOWN the tree picking each time the node with highest $U_{i,h}$ until a leave is met

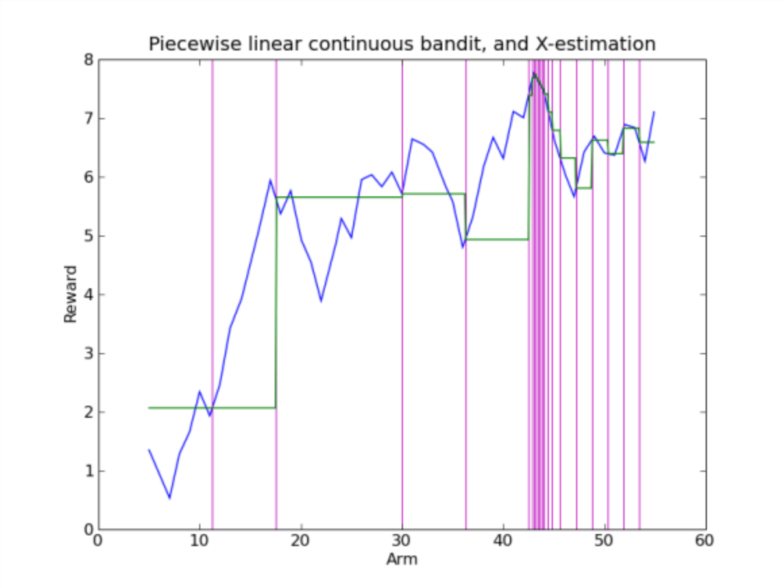
  PICK a point at random in leave cell

  UPDATE $U_{h,i}$ and $B_{h,i}$ of all nodes in the path

Theorem: If f has near-optimality dimension d, then cumulated regret of HOO is upper-bounded as

$$R_T = O(T^{(d+1)/(d+2)} \log^{1/(d+2)}(T)) \ .$$

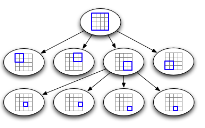
The HOO Algorithm



Source: aresearch.wordpress.com

The StoSOO Algorithm of Valko et al. [2013]

Stochastic Simulataneous Optimistic Optimization: instead of $f(x)$, use an upper-confidence bound.



StoSOO

FOR r=1..R
  FOR every non-empty depth d
    PICK the cell $C_{h,i}$ of depth d
    with highest upper-confidence bound on $f(x_{C_{h,i}})$
    IF $x_{C_{h,i}}$ has been evaluated $T/\log^3(T)$ times
      THEN SPLIT it
      ELSE evaluate at $x_{C_{h,i}}$

Theorem: If the near-optimality dimension of f is $d = 0$, then

$$\mathbb{E}[f^* - f(\hat{x}_T)] = O\left(\frac{\log^2(T)}{\sqrt{T}}\right) .$$

Conclusion

Still a lot to be done, from both ends:

- Kriging: Powerful and versatile algorithms, but with very low guarantees.
- Optimal bandit algorithms for very limited settings, to be extended!
- Adaptivity to the problem difficulty: function regularity, partitioning scheme.

References

S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári. X-armed bandits. *Journal of Machine Learning Research*, 12:1587–1627, 2011.

H. Chernoff. Sequential design of Experiments. *The Annals of Mathematical Statistics*, 30(3):755–770, 1959.

Nando de Freitas, Alex Smola, and Masrour Zoghi. Regret Bounds for Deterministic Gaussian Process Bandits. Technical Report arXiv:1203.2177, Mar 2012. URL `https://cds.cern.ch/record/1430976`. Comments: 17 pages, 5 figures.

References II

E. Even-Dar, S. Mannor, and Y. Mansour. Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

V. Gabillon, M. Ghavamzadeh, and A. Lazaric. Best Arm Identification: A Unified Approach to Fixed Budget and Fixed Confidence. In *Advances in Neural Information Processing Systems*, 2012.

A. Garivier and E. Kaufmann. Optimal Best Arm Identification with Fixed Confidence, 2016.

K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck. lil'UCB: an Optimal Exploration Algorithm for Multi-Armed Bandits. In *Proceedings of the 27th Conference on Learning Theory*, 2014.

References III

S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone. PAC subset selection in stochastic multi-armed bandits. In *International Conference on Machine Learning (ICML)*, 2012.

E. Kaufmann and S. Kalyanakrishnan. Information complexity in bandit subset selection. In *Proceeding of the 26th Conference On Learning Theory.*, 2013.

R. Munos. *From bandits to Monte-Carlo Tree Search: The optimistic principle applied to optimization and planning.*, volume 7. Foundations and Trends in Machine Learning, 2014.

References IV

Rémi Munos. Optimistic optimization of a deterministic function without the knowledge of its smoothness. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *NIPS*, pages 783–791, 2011. URL `http://dblp.uni-trier.de/db/conf/nips/nips2011.html#Munos11`.

N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

References V

Michal Valko, Alexandra Carpentier, and Rémi Munos. Stochastic simultaneous optimistic optimization. In Sanjoy Dasgupta and David Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 19–27. JMLR Workshop and Conference Proceedings, May 2013. URL `http://jmlr.csail.mit.edu/proceedings/papers/v28/valko13.pdf`.

Emmanuel Vazquez and Julien Bect. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and Inference*, 140(11): 3088 – 3095, 2010. ISSN 0378-3758. doi: http://dx.doi.org/10.1016/j.jspi.2010.04.018. URL `http://www.sciencedirect.com/science/article/pii/S0378375810001850`.

Questions?