Statistical learning approaches for global optimization

Emile Contal

université PARIS-SACLAY

école normale supérieure paris–saclay

NNT : 2016SACLN038

Thèse de doctorat de l'Université Paris-Saclay préparée à l'École Normale Supérieure Paris-Saclay

Ecole doctorale n°574
Ecole doctorale de mathématiques Hadamard
Spécialité de doctorat : Mathématiques appliquées

par

M. Emile Contal

Méthodes d'apprentissage statistique pour l'optimisation globale

Thèse présentée et soutenue à Cachan, le 29 septembre 2016.

Composition du Jury :

M.	Pascal Massart	Professeur Université Paris-Sud	(Président)
M.	Josselin Garnier	Professeur LPMA, University Paris Diderot	(Rapporteur)
M.	Andreas Krause	Professeur associé ETH Zürich	(Rapporteur)
M.	Aurélien Garivier	Professeur IMT, Université Paul-Sabatier	(Examinateur)
M.	Vianney Perchet	Professeur CMLA, ENS Paris-Saclay	(Examinateur)
M.	Nicolas Vayatis	Professeur CMLA, ENS Paris-Saclay	(Directeur de thèse)

Acknowledgments

First of all I would like to thank my adviser Nicolas Vayatis for his precious insights and directions, especially about all the unwritten rules of academic research. Thank you Nicolas, for making this thesis, wonderful opportunities and collaborations possible, as well as for the steady faith in my abilities and for your encouraging support. Many thanks to my colleagues at CMLA, Julien Audiffren, Argyris Kalogeratos, Rémi Lemmonier, Cédric Malherbe, Thomas Moreau, Kévin Scaman and Charles Truong for the delightful time we spent together and for those discussions on maths, the universe and everything. Every day in this team was an unforgettable joy. I greatly appreciated carrying out my thesis alongside Véronique Almadovar, Micheline Brunetti and Virginie Pauchont for their skills and kind organizational backing. Going to conferences all over the world would be less enjoyable without you. It was a pleasure to work with the brilliant minds of Laurent Oudre and Vianney Perchet, I hope our collaborations will continue. I also would like to thank Guillaume Lecué and Sasha Rakhlin for brief but decisive discussions. The exciting partnerships with Rémi Barrois-Muller, Matthieu Robert, Dripta Sarkar and Themistoklis Stefanakis were all incredible chances to learn and share knowledge in diverse concrete perspectives. It is a great honor that Josselin Garnier and Andreas Krause accepted to review this thesis, and to have Aurélien Garivier and Pascal Massart as members of the jury as well. I am grateful to all the anonymous reviewers who provided valuable insights for all the articles I submitted during this thesis. I am pleased to thank Charles-Pierre Astolfi, Alban Levy, David Montoya and Alexandre Robicquet for all the shared moments during this adventure. Finally, my warmest thanks go to Emeline Brulé who has always stood by my side.

Contents

Résumé en français (French Summary) 7
1 Objectifs de la thèse . . . 7
2 Revue de la littérature . . . 8
2.1 Bref historique de l'optimisation de fonctions "boîtes noires" . . . 8
2.2 Résultats théoriques connus . . . 10
3 Organisation du document . . . 14
4 Contributions . . . 14
4.1 Optimisation séquentielle par batch . . . 14
4.2 Optimisation bayésienne et espaces métriques . . . 16
4.3 Optimisation de fonctions non-régulières par ordonnancement . . . 18
4.4 Applications . . . 20

1 Introduction **21**
1.1 Context of the Thesis . . . 22
1.2 Related Work . . . 22
1.2.1 A Short History of Optimization of Expensive Black-Box Functions . . . 23
1.2.2 Theoretical Analysis with the Bandit Framework . . . 24
1.3 Contributions . . . 25
1.3.1 Batch Bayesian Optimization . . . 25
1.3.2 Bayesian Optimization in Metric Spaces . . . 26
1.3.3 Non-Smooth Optimization by Ranking . . . 26
1.3.4 Applications and Efficient Implementations . . . 27
1.4 Outline . . . 27

2 Sequential Global Optimization **29**
2.1 Problem Formulation and Fundamental Ingredients . . . 31
2.1.1 Sequential Optimization via Noisy Measurements . . . 31
2.1.2 Cumulative and Simple Regrets . . . 32
2.1.3 Smoothness, Metric Spaces and Covering Dimensions . . . 33
2.1.4 Bayesian Assumption, Gaussian Processes and Continuity . . . 36
2.1.5 Practical and Theoretical Properties of Priors and Posteriors . . . 43
2.2 Optimization Algorithms and Theoretical Results . . . 45
2.2.1 Stochastic Multi-Armed Bandits . . . 45
2.2.2 Stochastic Linear Bandits . . . 50
2.2.3 Lipschitzian Optimization . . . 51
2.2.4 Bayesian Optimization and Gaussian Processes . . . 56

3 Advances in Bayesian Optimization **61**
3.1 Batch Sequential Optimization . . . 63
3.1.1 Problem Formulation and Objectives . . . 63
3.1.2 Parallel Optimization Procedure . . . 64
3.1.3 Theoretical Analysis . . . 66
3.1.4 Experiments . . . 70
3.1.5 Conclusion and Discussion . . . 72
3.2 Gaussian Processes in Metric Spaces . . . 73

3.2.1 Hierarchical Discretizations of the Search Space . . . 73
3.2.2 Regret Bounds for Bandit Algorithms . . . 77
3.2.3 Efficient Algorithms . . . 82
3.2.4 Tightness Results on Discretization Trees . . . 84
3.2.5 Proof of the Generic Chaining Lower Bound . . . 87
3.2.6 Conclusion and Discussions . . . 90
3.3 Beyond Gaussian Processes . . . 90
3.3.1 Generic Stochastic Processes . . . 90
3.3.2 Quadratic Forms of Gaussian Processes . . . 93
3.3.3 Conclusion and Discussions . . . 96

4 Non-Smooth Optimization and Ranking **99**
4.1 Introduction . . . 101
4.2 Global Optimization and Ranking Structure . . . 101
4.2.1 Setup and Notations . . . 101
4.2.2 The Ranking Structure of a Real-Valued Function . . . 102
4.3 Optimization with Fixed Ranking Structure . . . 105
4.3.1 The RankOpt Algorithm . . . 105
4.3.2 Convergence analysis . . . 106
4.4 Adaptive Algorithm and Stopping Time Analysis . . . 110
4.4.1 The AdaRankOpt Algorithm . . . 110
4.4.2 Theoretical Properties of AdaRankOpt . . . 111
4.5 Computational Aspects . . . 113
4.5.1 General ranking structures . . . 113
4.5.2 Practical Solutions for Particular Ranking Structures . . . 114
4.6 Experiments . . . 117
4.6.1 Protocol of the Empirical Assessment . . . 117
4.6.2 Empirical Comparisons . . . 118
4.7 Conclusion and Discussion . . . 119

5 Applications and Implementation Details **121**
5.1 Efficient Computations and Software Library for Bayesian Optimization . . . 122
5.1.1 Bayesian Inference . . . 122
5.1.2 Gaussian Process Prior Selection and Validation . . . 123
5.1.3 Non-Gaussian Processes . . . 124
5.1.4 Software Library Release . . . 125
5.2 Applications for Physical Simulations . . . 125
5.2.1 Tsunamis Amplification Phenomenon . . . 125
5.2.2 Wave Energy Converters . . . 129
5.3 Applications to Model Calibration . . . 133
5.3.1 Calibration of Force Fields Parameters for Molecular Simulation . . . 133
5.3.2 Hyper-Parameter Optimization and Further Perspectives . . . 134

Conclusion and Perspectives **137**

A Appendix **139**
Attempt for Improved Regret Bounds . . . 139
1 Proof Techniques to Get Rid of the Square Root . . . 139
2 The GP-MI Algorithm and Theoretical Obstacles . . . 142
3 Empirical Assessment . . . 144

Bibliography **145**

Résumé en français

1 Objectifs de la thèse

Cette thèse se consacre à une analyse rigoureuse des algorithmes d'optimisation globale séquentielle. L'optimisation globale apparaît dans de nombreux domaines, y compris les sciences naturelles (Floudas and Pardalos, 2000), le génie industriel (Wang and Shan, 2007), la bioinformatique (Moles et al., 2003), la finance (Ziemba and Vickson, 2006) et beaucoup d'autres. Elle vise à trouver l'entrée d'un système donné qui optimise la sortie. L'objectif d'optimisation est typiquement la maximisation d'une récompense, ou la minimisation d'un coûts. La fonction qui relie l'entrée à la sortie n'est pas connue, mais on dispose d'une manière d'évaluer la sortie pour toute entrée. Les mesures peuvent provenir d'expériences en laboratoire, de simulations numériques, de réponses de capteurs ou n'importe quel retour en fonction de l'application. En particulier, cette fonction peut ne pas être convexe et peut contenir un grand nombre d'optima locaux. Dans ce travail, nous abordons le cas difficile où les évaluations sont coûteuses, ce qui exige de concevoir une sélection rigoureuse des entrées à évaluer. Ainsi, une procédure itérative utilise les mesures acquises précédemment pour choisir la prochaine requête la plus utile. Nous étudions deux objectifs différents, d'une part la maximisation de la somme des récompenses reçues à chaque itération, qui est pertinent pour l'optimisation "en ligne" telle que des essais cliniques ou des systèmes de recommandation ; d'autre part la maximisation de la meilleure récompense trouvée jusqu'à présent, pertinente pour la recherche d'un optimum telle que l'optimisation numérique. La complexité numérique est souvent une préoccupation essentielle pour un praticien, nous décrivons des solutions pratiques tout au long de ce travail et nous fournissons des détails de mise en œuvre. L'objectif est d'apporter de nouveaux concepts issus de la théorie visant à décrire l'efficacité des procédures d'optimisation par rapport à des notions génériques de la complexité du problème.

Notations et critères de performance

On modélise le système à optimiser par une fonction inconnue $f : \mathcal{X} \to \mathbb{R}$. L'espace d'entrée $\mathcal{X}$ peut être fini ou infini, paramétrique ou non paramétrique. Un problème d'optimisation avec contrainte sera simplement modélisé en restreignant $\mathcal{X}$ aux entrées qui satisfont les contraintes. Un algorithme d'optimisation propose des évaluations de f en tout point $x \in \mathcal{X}$, et reçoit alors l'observation associée bruitée $y = f(x) + \epsilon$, où ϵ modélise un bruit centré, additif et indépendant de tout le reste. Nous considérons en particulier le cas du bruit gaussien :

$$\epsilon \sim \mathcal{N}(0, \eta^2).$$

On parlera d'optimisation déterministe lorsque les observations ne sont pas bruitées ($\epsilon = 0$ presque sûrement). Nous nous intéressons au cas où les évaluations de f ont un coût élevé, forçant l'algorithme d'optimisation à choisir avec soin ses requêtes pour minimiser le nombre d'évaluations nécessaires. Nous définissons deux critères de performance d'un algorithme. Soit $x_1, \dots, x_n$ les requêtes de l'algorithme après n itérations. Le *regret simple* est la différence entre l'optimum de la fonction inconnue et la meilleur valeur trouvée jusqu'à présent :

$$S_n = \sup_{x^\star \in \mathcal{X}} f(x^\star) - \max_{i \leqslant n} f(x_i).$$

Cette quantité n'est pas accessible en pratique, et l'enjeu de l'analyse théorique est de fournir des preuves de convergences vers zéro et des vitesses de convergence. Le *regret cumulé* est la somme des différences entre l'optimum inconnu et les valeurs des points évalués :

$$R_n = \sum_{i=1}^{n} \Big(\sup_{x^\star \in \mathcal{X}} f(x^\star) - f(x_i) \Big).$$

Le but d'un algorithme concernant le regret cumulé est d'obtenir des bornes supérieures sous-linéaires les plus faibles possibles. On remarque qu'une borne sur ce regret induit une borne sur le regret simple puisque $S_n \leqslant \frac{R_n}{n}$.

2 Revue de la littérature

Le domaine de la théorie de l'optimisation englobe de nombreuses approches, cette thèse s'inscrit dans les cadres suivants :

- les algorithmes de bandits stochastique à K bras, où $\mathcal{X} = (1, \dots, K)$,
- l'optimisation lipschitzienne, où $\mathcal{X} \subset \mathbb{R}^d$ et f est lipschitzienne,
- l'optimisation bayésienne, où f est tirée suivant un processus stochastique connu.

Dans ce qui suit, nous présentons une courte perspective historique sur les travaux étroitement liés. Sur des questions connexes dans des cadres différents, nous nous référons par exemple aux travaux de Boyd and Vandenberghe (2004) et Bubeck (2015) sur les approches par gradient lorsque la fonction est convexe, ce qui n'est pas adapté à l'optimisation globale puisque trouver un minimum local ne permet pas de contrôler le regret ; les travaux de Sebag and Ducoulombier (1998), Garnier and Kallel (2000) et Eiben and Smith (2003) sur les algorithmes évolutionnaires lorsque le coût des évaluations est faible, qui ne donnent pas de garanties sur le regret ; et les travaux de Papadimitriou and Steiglitz (1982) et Garnier and Kallel (2001) sur l'optimisation combinatoire.

2.1 Bref historique de l'optimisation de fonctions "boîtes noires"

Méthode des surfaces de réponse

Une des façons les plus courantes pour faire face à ce problème consiste à construire et mettre à jour une fonction de substitution à partir des évaluations. Cette fonction de substitution est calculée grâce aux observations et généralise aux entrées inconnues. Cette étape peut être vue comme l'estimation d'un modèle type régression, en particulier lorsque les observations sont

bruitées. On utilise alors traditionnellement l'interpolation polynomiale ou les régressions par noyau et moindres carrés. Les évaluations de cette fonction de substitution sont immédiates. Il est donc facile d'utiliser cette estimation empirique pour choisir la prochaine entrée de la fonction "boîte noire" à mesurer. Cette technique est appelée la méthode des surfaces de réponse (Myers and Montgomery, 1995; Jones, 2001) et a été introduite dans Box and Wilson (1951). Les inconvénients majeurs de ce qui précède sont tout d'abord que la sélection de la famille de substitution—c'est à dire la sélection de modèles—est un sujet crucial et délicat, deuxièmement la présence d'optima locaux pose problème à toutes les approches locales qui n'incluent pas de termes d'exploration globale, et troisièmement ça n'est pas adapté à une analyse théorique complète.

Optimisation lipschitzienne

Motivées par l'objectif de prouver la convergence globale des algorithmes d'optimisation, des techniques sont apparues plus tard utilisant le fait que la fonction inconnue f satisfait une condition de régularité. Ce cadre suppose que f est lipschitzienne, c'est à dire que son gradient est borné. Connaissant un ensemble de valeurs de la fonction et la borne de son gradient, on peut supprimer de l'espace de recherche la région où l'optimum ne peut pas se trouver sans casser la propriété lipschitzienne. En échantillonnant dans la région restante, il est possible d'obtenir des garanties théoriques de convergence. Cette idée date de l'algorithme de Shubert-Mladineo (Shubert, 1972; Mladineo, 1986). Malheureusement, la connaissance de la constante de Lipschitz n'est souvent pas réaliste. Des algorithmes adaptatifs estiment cette constante sur les données acquises au cours de l'optimisation. Cependant, ils peuvent produire une mauvaise vitesse de convergence lorsque la constante de Lipschitz est estimée très grande à cause d'un motif non régulier isolé dans la fonction, ce qui pourrait être une valeur aberrante non pertinente pour la tâche d'optimisation. L'algorithme DIRECT (Jones et al., 1993) résout ce problème en utilisant une recherche dichotomique améliorée qui ne nécessite pas d'estimer la constante de Lipschitz. La robustesse de la méthode résultante en fait un choix encore aujourd'hui utilisé couramment pour l'optimisation globale non bruitée.

Optimisation bayésienne

Un cadre plus moderne, l'optimisation Bayésienne, introduite dans Kushner (1964) et Močkus (1974), surmonte certains des problèmes précédents et s'adapte facilement au bruit des observations. En supposant a priori que la fonction sous-jacente inconnue est une réalisation d'un processus stochastique il est possible de calculer une distribution a posteriori via les données acquises, à partir de laquelle on déduit une espérance et des incertitudes pour les entrées inconnues. Nous nous intéressons alors au comportement moyen d'une procédure d'optimisation où la fonction est stochastique, ou pour obtenir des résultats plus forts, on tâchera de démontrer des propriétés valides avec une forte probabilité. Les distributions a priori les plus courantes sont de loin les processus gaussiens. La régularité de leur covariance implique une hypothèse sur la régularité de la fonction, en forçant les valeurs proches dans l'espace induit par le noyau à être fortement corrélées. Cependant, cette hypothèse est moins restrictive qu'une borne sur la constante de Lipschitz puisqu'il suffit que les contraintes soient vraies avec une forte probabilité. Les stratégies d'optimisation peuvent alors utiliser des intervalles de confiance (Cox and John, 1997; Srinivas et al., 2012), ou un critère intégré comme *Expected Improvement* (Jones et al., 1998) ou *Expected Information Gain* (Hennig

and Schuler, 2012). Nous renvoyons à Brochu et al. (2010) pour une revue des différentes méthodes d'acquisition.

2.2 Résultats théoriques connus

Modèle des bandits à K bras

L'analyse théorique des algorithmes visant à maximiser la récompense cumulée est souvent présentée dans le cadre des bandits manchots à plusieurs bras. Ce modèle considère qu'on présente à un joueur K bras, les entrées possibles. On note alors $\mathcal{X} = (1, \dots, K)$, et f ne possède aucune structure particulière. Quand le joueur choisit un bras, il reçoit une récompense bruitée distribuée indépendamment des récompenses précédentes, d'une loi dépendante du bras choisi. Afin de maximiser les récompenses, on doit faire face au compromis exploration/exploitation. Le premier algorithme Bayésien dans cette perspective remonte à Thompson (1933), et la première analyse Bayésienne à Gittins (1979). L'analyse théorique du regret cumulé R_n a été effectuée de manière approfondie dans Lai and Robbins (1985) où une borne inférieure générique est présentée. Soit ν_a la distribution d'une observation bruitée du bras $a \in \mathcal{X}$, c'est à dire une gaussienne de moyenne $f(a)$ et de variance η^2 dans le cas gaussien. Les auteurs du précédent article montrent que pour un algorithme dont le regret est sous-linéaire,

$$\liminf_{n\to\infty} \frac{\mathbb{E}\big[N_n(a)\big]}{\log n} \geqslant D_{\mathrm{KL}}(\nu_a \parallel \nu_\star)^{-1},$$

où $N_n(a)$ est le nombre de fois que le bras a a été tiré après n itérations, et $D_{\mathrm{KL}}(\nu_a \parallel \nu_\star)$ est la divergence de Kullback-Leibler entre les distributions des observations du bras a et du bras optimal. Dans le cas gaussien,

$$D_{\mathrm{KL}}(\nu_a \parallel \nu_\star) = \Delta_a^2/(2\eta^2),$$

où $\Delta_a = \sup_{x^\star \in \mathcal{X}} f(x^\star) - f(a)$. En décomposant le regret cumulé, on obtient :

$$\liminf_{n\to\infty} \frac{\mathbb{E}\big[R_n\big]}{\log n} \geqslant 2\eta^2 \sum_{a:\Delta_a>0} \Delta_a^{-1}.$$

Il existe des algorithmes d'optimisation tels que l'on peut prouver une borne supérieure associée. Dans Auer et al. (2002), les auteurs donnent une stratégie obtenant un regret cumulé optimal à constante prêt. Cet algorithme nommé UCB maintient pour tout les bras une borne de confiance supérieure, et évalue à chaque itération le bras dont cette borne est maximale. Soit $\widehat{\mu}_n(a)$ la moyenne empirique d'un bras $a \in \mathcal{X}$:

$$\widehat{\mu}_n(a) = N_n(a)^{-1} \sum_{i=1}^{n} y_i \mathbb{1}\{x_i = a\}.$$

Alors, la borne supérieure de confiance vaut :

$$U_n(a) = \widehat{\mu}_n(a) + \sqrt{2\eta^2 \frac{3\log n}{N_n(a)}},$$

et l'algorithme choisit,

$$x_{n+1} \in \operatorname*{argmax}_{a \in \mathcal{X}} U_n(a).$$

Cet algorithme et son analyse ouvrent la voie à de nombreux travaux dans d'autres cadres. Dans le modèle de bandit, l'étude du regret simple est couramment formulée par le problème d'identification du meilleur bras. On dit qu'un algorithme est (ε, δ)-PAC lorsque, connaissant $\varepsilon > 0$ et $\delta > 0$, il s'arrête après $n_{\varepsilon,\delta}$ itérations et $\mathbb{P}\big[S_{n_{\varepsilon,\delta}} > \varepsilon\big] < \delta$. L'article Mannor and Tsitsiklis (2004) donne une borne inférieure sur l'espérance de $n_{\varepsilon,\delta}$ nécessaire pour satisfaire la propriété (ε, δ)-PAC. Les auteurs prouvent qu'il existe $c_1, c_2 \in \mathbb{R}$ tels que pour ε et δ suffisamment petits,

$$\mathbb{E}\big[n_{\varepsilon,\delta}\big] \geqslant c_1 \frac{K}{\varepsilon^2} \log \frac{c_2}{\delta}.$$

Dans un problème d'optimisation globale classique, ε n'est pas connu, mais la borne inférieure est toujours valide, c'est à dire :

$$\mathbb{P}\left[S_n \geqslant \sqrt{c_1 \frac{K}{n} \log \frac{c_2}{\delta}} \right] \geqslant 1 - \delta.$$

Dans Even-Dar et al. (2006), un algorithme (ε, δ)-PAC est présenté utilisant un nombre d'itérations optimal à facteur multiplicatif près.

Bandits linéaires

L'approche précédente n'est pas adaptée au cas où l'espace de recherche n'est pas fini, par exemple un compact. La première extension que l'on considère est la restriction de f à un espace de fonctions linéaires dans $\mathcal{X}$ un fermé de $\mathbb{R}^d$. Dans ce cadre un optimum se trouve toujours sur $\mathcal{E}$ les points extrêmes de $\mathcal{X}$, c'est à dire les points qui ne sont pas des combinaisons convexes d'autres points. On distingue alors deux cas, suivant la valeur suivante :

$$\Delta = \inf \Big\{ \sup_{x^\star \in \mathcal{X}} f(x^\star) - f(x) : x \in \mathcal{E}, \sup_{x^\star \in \mathcal{X}} f(x^\star) > f(x) \Big\}.$$

Lorsque $\Delta > 0$, par exemple pour $\mathcal{X}$ un polytope, alors le problème est proche du problème des bandits où les bras sont les coins du polytope. L'analyse effectuée par Auer et al. (2007) et Dani et al. (2008) donne la borne inférieure suivante, pour $c \in \mathbb{R}$ et pour tout $u > 0$, pour tout algorithme :

$$\mathbb{P}\Big[\forall n \geqslant 1,\ R_n \geqslant c\eta^2 \Delta^{-1} d^2 \big(u + \log^3 n\big) \Big] \geqslant 1 - e^{-u},$$

ainsi que des algorithmes atteignant ce regret. Lorsque $\Delta = 0$, tel que pour $\mathcal{X}$ sphérique, l'ordre de grandeur du regret cumulé n'est plus poly-logarithmique mais polynomial. Il existe en effet des cas où la borne inférieure suivante est vérifiée pour tout algorithme :

$$\mathbb{E}\big[R_n\big] \geqslant \Omega\big(d\sqrt{n}\big).$$

Optimisation lipschitzienne

L'hypothèse de linéarité de f limite grandement les applications pour certains problèmes d'optimisation globale. Sous des conditions moins strictes que l'hypothèse de fonction cible

linéaire, on trouve des méthodes présentant des garanties théoriques. Par exemple lorsque f satisfait des conditions de Lipschitz affaiblies. Pour le regret cumulé, Kleinberg (2004) étudie le cas unidimensionnel et prouve des limites supérieures et inférieures presque équivalentes, sous continuité de Hölder. Les auteurs montrent que pour tout algorithme,

$$\mathbb{E}\big[R_n\big] \geqslant \Omega\big(n^{2/3}\big),$$

et proposent également un algorithme atteignant cette borne à un facteur multiplicatif en logarithme près :

$$\mathbb{E}\big[R_n\big] \leqslant \mathcal{O}\big(n^{2/3}\log^{1/3} n\big).$$

En dimension supérieure, Kleinberg et al. (2008) et Bubeck et al. (2008) prouvent que le regret de tout algorithme est plus grand que :

$$\mathbb{E}\big[R_n\big] \geqslant \Omega\Big(n^{\frac{d+1}{d+2}}\Big),$$

et présentent de même des algorithmes possédant le regret optimal à un facteur poly-logarithmique près,

$$\mathbb{E}\big[R_n\big] \leqslant \mathcal{O}\Big(n^{\frac{d+1}{d+2}}\big(\log n\big)^{\frac{1}{d+2}}\Big).$$

Ces approches supposent une régularité de f pour une métrique plus générale que la distance euclidienne, et la dimension d est alors définie par des concepts proches des dimensions de Minkowski ou de Hausdorff. Les précédents articles montrent qu'il suffit d'analyser le comportement local de f autour du maximum, et définissent alors des notions de dimensions locales potentiellement plus basses que la dimension globale. Dans Munos (2011), l'auteur étudie le cas où l'algorithme ne connaît pas la régularité de la fonction mais aucun bruit n'affecte les observations. Dans ce cas, un algorithme par partitionnement hiérarchique atteint une convergence exponentielle du regret simple lorsque la dimension locale d vaut 0 et que le partitionnement est suffisamment régulier,

$$S_n \leqslant \mathcal{O}\big(e^{-n}\big).$$

Pour obtenir $d = 0$, il suffit que f puisse être localement bornée par valeurs supérieures et inférieures par le même polynôme à facteur multiplicatif prêt. Cependant, même si l'algorithme n'a pas besoin de connaître la régularité de la fonction, il requiert de connaître un partitionnement hiérarchique de $\mathcal{X}$ adapté à cette régularité. Les auteurs ne présentent pas de manière générique de construire ce partitionnement. Cette approche est étendue au cas bruité par les récents travaux de Valko et al. (2013), Bull (2015) et Grill et al. (2015). Le regret simple obtenu est borné par :

$$\mathbb{E}\big[S_n\big] \leqslant \mathcal{O}\Big(n^{\frac{1}{d+2}}\big(\log n\big)^{\frac{2}{d+2}}\Big),$$

ce qui est optimal à facteurs multiplifactif en logarithme près.

Optimisation bayésienne

L'analyse des algorithmes d'optimisation bayésienne est classiquement effectuée en modélisant la fonction f par un processus gaussien de noyau k fixé et connu. Il n'existe pas à notre

connaissance de bornes inférieures dans le cas général. On sépare dans la suite deux cas selon que les observations soient déterministes ou bruitées. Pour le regret simple avec l'horizon n fixé et connu et des observations déterministes, Grünewälder et al. (2010) prouvent une borne inférieure lorsque le noyau k satisfait une condition de Hölder avec exposant α, alors :

$$\mathbb{E}[S_n] \geqslant \Omega\left(n^{-\frac{\alpha}{2d}} \log^{-\frac{1}{2}} n\right).$$

Les auteurs montrent également qu'un algorithme n'utilisant pas les observations qui effectue ses requêtes selon une grille pré-définie obtient un regret :

$$\mathbb{E}[S_n] \leqslant \mathcal{O}\left(n^{-\frac{\alpha}{2d}}\right).$$

Lorsque $\mathcal{X}$ est fini et que f a presque sûrement un comportement quadratique autour de son maximum, de Freitas et al. (2012) donne un algorithme dont le regret simple décroît exponentiellement vite, pour $a > 0$, avec forte probabilité :

$$S_n \leqslant \mathcal{O}\left(\log^{\frac{1}{2}} |\mathcal{X}| e^{-\frac{an}{\log^{d/4} n}}\right).$$

Dans le cas bruité, les performances se dégradent brutalement. Même lorsque $\mathcal{X} \subset \mathbb{R}^d$ et le noyau est linéaire—ce qui modélise l'optimisation de fonctions linéaires avec un a priori gaussien—Rusmevichientong and Tsitsiklis (2010) prouvent que pour tout algorithme,

$$\mathbb{E}[R_n] \geqslant \Omega\left(d\sqrt{n}\right).$$

L'algorithme GP-UCB (Srinivas et al., 2012) atteint ce regret avec probabilité au moins $1 - e^{-u}$ avec $u > 0$ fixé et connu. Lorsque $\mathcal{X}$ est fini,

$$R_n \leqslant \mathcal{O}\left(\sqrt{n\gamma_n\left(u + \log(n|\mathcal{X}|)\right)}\right),$$

et pour $\mathcal{X}$ non fini lorsque la distribution de la constante de Lipschitz de f possède une queue sous-gaussienne de variance b^2 connue,

$$R_n \leqslant \mathcal{O}\left(\sqrt{dn\gamma_n\left(u + \log(nb)\right)}\right).$$

La quantité γ_n mesure le coût d'exploration de $\mathcal{X}$ en fonction du noyau k et du bruit ϵ, exprimé en termes d'information mutuelle. Pour un noyau linéaire, on obtient $\gamma_n \leqslant \mathcal{O}(d \log n)$, et pour un noyau stationnaire avec décroissance gaussienne, $\gamma_n \leqslant \mathcal{O}\left(\log^{d+1} n\right)$. On remarque que dans le cas où $\mathcal{X}$ est fini la borne obtenue avec forte probabilité est meilleure que la borne inférieure précédente sur l'espérance, ce qui n'est pas une contradiction théorique puisque u est fixé et connu et le regret pourrait être linéaire avec probabilité e^{-u}. Les deux bornes sont équivalentes pour $\mathcal{X}$ non fini.

Nous concluons cette section en mentionnant que dans l'article Srinivas et al. (2012) ainsi que dans Bull (2011), les auteurs étudient également le cas où l'algorithme est Bayésien, mais la fonction f est fixée dans le RKHS du noyau associé. Nous verrons dans la thèse que les fonctions du RKHS sont strictement plus régulières que les réalisations du processus, et que le RKHS est en général de mesure nulle pour le processus.

3 Organisation du document

L'axe principal de ce document porte sur l'optimisation bayésienne par processus gaussiens et des observations bruitées, un cadre similaire à l'article Srinivas et al. (2012). Ce document aborde les questions suivantes :

- Quel est l'impact sur le regret de ne pas recevoir les observations après chaque requête mais par batchs successifs de taille fixée ?
- Comment adapter de manière générique l'optimisation bayésienne à un espace de recherche continu ou nonparamétrique ?
- Est-ce que l'analyse théorique de l'optimisation bayésienne s'étend à des processus non gaussiens ?
- Dans un modèle fréquentiste, peut-on construire un algorithme avec garanties théoriques lorsque f n'est pas régulière voire non continue ?

La suite de ce résumé en français décrit les contributions pour les questions précédemment citées. Le chapitre 1 développe une introduction détaillée de la thèse. Dans le chapitre 2, nous présentons les concepts fondamentaux de l'optimisation globale et exposons les algorithmes principaux et les résultats théoriques associés. Le chapitre 3 est consacré aux trois premiers des points précédents. Nous commençons par analyser une extension de l'optimisation séquentielle où les évaluations sont obtenues par batch et non une par une. Nous présentons ensuite une approche novatrice pour déterminer des bornes de confiances supérieures sur des processus gaussiens dans des espaces métriques qui s'adapte à une régularité arbitraire. Cela permet de concevoir des algorithmes génériques présentant des garanties sur le regret au niveau de l'état de l'art. Nous montrerons que cette approche s'étend naturellement à des processus stochastiques non gaussiens, et nous illustrerons l'intérêt de cette extension pour l'optimisation de formes quadratiques. Ensuite dans le chapitre 4, nous introduisons un nouveau cadre théorique d'optimisation, où la fonction inconnue n'est plus supposée régulière, mais satisfait seulement une condition sur ses ensembles de niveaux. Enfin dans le chapitre 5, nous présentons nos contributions dans plusieurs applications. Nous fournissons des détails d'implémentation, ainsi que le code source de nos implémentations sous forme de librairies libres Matlab et Python.

4 Contributions

4.1 Optimisation séquentielle par batch

La première contribution de cette thèse est un algorithme d'optimisation globale sélectionnant les évaluations par batch. Cela modélise en particulier les cas où les évaluations sont obtenues en parallèle pour le coût d'une seule itération. Les exemples typiques sont l'optimisation numérique avec plusieurs machines, ou la recherche de la réponse maximale d'un capteur lorsque l'on dispose de plusieurs capteurs.

Cadre et algorithme

Dans cette section, on note $\{x_{n,k}\}_{0 \leqslant k < K}$ les K points de $\mathcal{X}$ sélectionnés à l'itération n, et de même $y_{n,k}$ les observations bruitées associées. Soit $r_{n,k}$ le regret instantané d'un point :

$$r_{n,k} = \sup_{x \in \mathcal{X}} f(x) - f(x_{n,k}),$$

on distingue alors deux regrets, le regret cumulé complet :

$$R_{n,K} = \sum_{i \leqslant n} \sum_{k < K} r_{i,k},$$

et le regret cumulé par batch :

$$\widetilde{R}_{n,K} = \sum_{i \leqslant n} \min_{k < K} r_{i,k}.$$

Cette dernière variante modélise le cas où le coût d'une itération est fixe, et l'on souhaite optimiser la somme des coûts. Comme auparavant, le regret simple est borné par $n^{-1}\widetilde{R}_{n,K}$. Notre algorithme, GP-UCB-PE pour *Gaussian Process Upper Confidence Bound with Pure Exploration*, se fonde sur le calcul de bornes de confiances supérieures et inférieures en tout point de l'espace. Nous procédons comme dans Srinivas et al. (2012) pour $\mathcal{X}$ fini, soient $\mu_n : \mathcal{X} \to \mathbb{R}$ et $\sigma_n^2 : \mathcal{X} \to \mathbb{R}^+$ les fonctions moyenne et variance a posteriori conditionnées aux nK observations obtenue jusqu'à l'itération n. Pour tout $u > 0$, on peut définir $\beta_n \in \mathcal{X}$ tel que avec probabilité au moins $1 - e^{-u}$ on ait :

$$\forall n \geqslant 1, \forall x \in \mathcal{X},\ f(x) \in \left(\mu_n(x) - \sqrt{\beta_n \sigma_n^2(x)}, \mu_n(x) + \sqrt{\beta_n \sigma_n^2(x)}\right).$$

Sous cet évènement, la position du maximum appartient à l'ensemble $\mathfrak{R}_n$ suivant,

$$\mathfrak{R}_n = \left\{x \in \mathcal{X} : \mu_n(x) + \sqrt{\beta_n \sigma_n^2(x)} \geqslant \sup_{x' \in \mathcal{X}} \mu_n(x') - \sqrt{\beta_n \sigma_n^2(x')}\right\}.$$

L'algorithme sélectionne le premier point du batch comme GP-UCB, c'est à dire :

$$x_{n+1,0} = \operatorname*{argmax}_{x \in \mathcal{X}} \left\{\mu_n(x) + \sqrt{\beta_n \sigma_n^2(x)}\right\}.$$

Les points suivants sont choisis de sorte à maximiser l'information sur f dans $\mathfrak{R}_n$. Pour cela on utilise la stratégie gloutonne d'exploration pure maximisant la variance conditionnelle :

$$x_{n+1,k+1} = \operatorname*{argmax}_{x \in \mathfrak{R}_n} \sigma_{n,k}^2(x),$$

où $\sigma_{n,k}^2$ est également conditionnée à $x_{n+1,0}, \ldots, x_{n+1,k}$. On note que cette sélection peut être effectuée en pratique car $\sigma_{n,k}^2$ ne dépend pas des observations $y_{n+1,k}$ mais seulement des positions $x_{n+1,k}$ choisies précédemment.

Garanties théoriques

Soit γ_{nK} la quantité introduite précédemment. Nous prouvons dans cette thèse les bornes supérieures suivantes pour l'algorithme GP-UCB-PE. Soit $u > 0$ et f un processus gaussien centré de noyau k connu avec $k(\cdot, \cdot) \leqslant 1$ perturbé par un bruit gaussien de variance η^2. Alors

avec $c_\eta = \frac{2}{\log(1+\eta^{-2})}$, le regret cumulé complet de GP-UCB-PE satisfait avec probabilité au moins $1 - e^{-u}$:

$$\forall n \geqslant 1,\ R_{n,K} \leqslant 4\sqrt{c_\eta(n-1)K\beta_n\gamma_{nK} + K\beta_0},$$

et le regret cumulé par batch satisfait :

$$\forall n \geqslant 1,\ \widetilde{R}_{n,K} \leqslant 2\sqrt{c_\eta \frac{n}{K}\beta_n\gamma_{nK}}.$$

Lorsque $K \ll n$ et $\gamma_n \ll n$, la borne supérieure que l'on obtient pour $\widetilde{R}_{n,K}$ est meilleure d'un facteur $\sqrt{K}$ par rapport à celle de R_n pour l'algorithme purement séquentiel GP-UCB, et la borne pour $R_{n,K}$ est équivalente à celle de R_{nK} pour GP-UCB.

4.2 Optimisation bayésienne et espaces métriques

Les bornes de confiance sont au cœur de l'algorithme précédent mais aussi de nombreux autres algorithmes de la littérature présentant des garanties théoriques. Le calcul de bornes de confiance satisfaites uniformément n'est pas aisé pour des processus stochastiques sur un espace continu. L'article Srinivas et al. (2012) traite principalement le cas où $\mathcal{X}$ est fini, où la borne de confiance est obtenue simplement par union sur tout les points de $\mathcal{X}$. Les auteurs montrent que l'algorithme peut être adapté au cas continu et d-dimensionnel lorsque la distribution de la constante de Lipschitz de f possède une queue sous-gaussienne de variance connue. Nous proposons dans ce document une solution adaptative à ce problème.

Algorithmes et bornes supérieures

Pour cela nous faisons appel aux techniques de chaînage générique (Talagrand, 2014). On remarque que lorsque f est un processus gaussien centré, pour tout point $x_1, x_2 \in \mathcal{X}$, la distribution a priori de $f(x_1) - f(x_2)$ est une gaussienne centrée de variance :

$$\ell^2(x_1, x_2) = k(x_1, x_2) + k(x_2, x_2) - 2k(x_1, x_2).$$

Cette fonction est symétrique et satisfait l'inégalité triangulaire, on l'appelle la pseudo-métrique canonique du processus. On a par inégalité classique que :

$$\mathbb{P}\Big[f(x_1) - f(x_2) > \sqrt{2u}\ell(x_1, x_2)\Big] < e^{-u}.$$

On ne peut pas considérer une union sur tout $\mathcal{X}$ lorsqu'il n'est pas dénombrable. Les techniques de chaînage consistent à discrétiser $\mathcal{X}$ de manière hiérarchique, puis à effectuer des bornes de la réunion des probabilités sur les ensembles discrets. Soit $(\mathcal{T}_h)_{h\geqslant 0}$ une séquence croissante de sous-ensembles de $\mathcal{X}$. Pour chaque niveau $h \geqslant 0$ on partitionne $\mathcal{X}$ suivant les cellules de Voronoï des points de $\mathcal{T}_h$. On peut alors montrer que pour tout $u > 0$, avec probabilité au moins $1 - e^{-u}$,

$$\forall h \geqslant 0,\ \forall s \in \mathcal{T}_h,\quad \sup_{x\in \mathrm{Cell}(s)} f(x) - f(s) \leqslant \sup_{x\in \mathrm{Cell}(s)} \sum_{i>h} \sqrt{2u_i}\Delta_{i-1}(x),$$

où $u_i = u + \log|\mathcal{T}_i| + \log(i^2\pi^2/6)$, et $\mathrm{Cell}(s)$ est la cellule de Voronoï de s et enfin $\Delta_i(x)$ est le ℓ-diamètre de la cellule de x pour $\mathcal{T}_i$. Soient $\varepsilon_i = \Delta(\mathcal{X})2^{-i}$ pour $i \in \mathbb{N}$. En construisant $\mathcal{T}_h$ de

sorte à minimiser $|\mathcal{T}_h|$ tel que $\sup_{s\in\mathcal{T}_h}\Delta_h(s)\leqslant\varepsilon_i$, on obtient avec $H(\cdot)$ l'entropie métrique de $\mathcal{X}$ pour ℓ,

$$\forall h\geqslant 0, \forall s\in\mathcal{T}_h,\quad \sup_{x\in\mathrm{Cell}(s)} f(x)-f(s)\leqslant\sum_{i>h}\sqrt{2u+H(\varepsilon_i)+\log(i^2\pi^2/6)}\varepsilon_{i-1},$$

En dimension d, lorsque le noyau est stationnaire et dominé par la norme euclidienne, on a

$$\forall h\geqslant 0,\ \forall s\in\mathcal{T}_h,\quad \sup_{x\in\mathrm{Cell}(s)} f(x)-f(s)\leqslant\mathcal{O}\Big(\sqrt{u+dh}2^{-h}\Big).$$

On peut donc dériver une stratégie d'optimisation qui, à l'itération n, choisit un niveau de discrétisation h_n, et calcule la borne supérieure de confiance sur $\mathcal{T}_{h_n}$. En sélectionnant h_n de sorte que l'erreur d'approximation ne dépasse pas le regret, soit $h_n=\mathcal{O}(\log n)$, nous prouvons dans ce document qu'avec probabilité au moins $1-2e^{-u}$, le regret cumulé est inférieur à :

$$R_n\leqslant\mathcal{O}\Big(\sqrt{c_\eta(u+d\log n)n\gamma_n\log^3 n}\Big).$$

Bornes inférieures

Nous montrons également dans cette thèse un résultat complétant la précédente borne supérieure sur l'erreur induite par la discrétisation. En effet après un élagage minutieux du partitionnement hiérarchique, on obtient avec probabilité au moins $1-e^{-u}$ que, à constantes multiplicatives près :

$$\forall h\geqslant 0,\forall s\in\mathcal{T}_h,\quad \sup_{x\in\mathrm{Cell}(s)} f(x)-f(s)\gtrsim\sup_{x\in\mathrm{Cell}(s)}\sum_{i>h}\sqrt{u_i}\Delta_{i-1}(x).$$

La preuve de ce résultat s'inspire de Talagrand (2014) qui prouve un résultat similaire en espérance. Un ingrédient supplémentaire apporté dans la thèse est également l'utilisation d'une méthode constructive pouvant s'implémenter via un algorithme concret.

Dépasser les processus gaussiens

Le cadre précédent s'étend aux processus non-gaussiens de la manière suivante. Nous remplaçons la pseudo-métrique canonique par :

$$\ell_u(x_1,x_2)=\inf\Big\{s\in\mathbb{R}:\mathbb{P}\big[f(x_1)-f(x_2)>s\big]<e^{-u}\Big\},$$

et étendons les résultats précédents pour toute borne supérieure sur ℓ_u. Pour un processus stochastique général, la fonction ℓ_u n'est pas une pseudo-métrique. On s'intéressera particulièrement aux processus tels qu'il existe une pseudo-métrique $\ell(\cdot,\cdot)$ et une fonction $\psi(\cdot,\cdot)$ qui vérifient :

$$\forall x_1,x_2\in\mathcal{X},\forall\lambda\in I\subset\mathbb{R},\ \log\mathbb{E}\Big[e^{\lambda(f(x_1)-f(x_2))}\Big]\leqslant\psi\big(\lambda,\ell(x_1,x_2)\big).$$

La croissance de ψ en ses deux arguments contrôle la régularité du processus. Plus cette fonction croît lentement et plus le processus est régulier. On a dans ce cas que la fonction ℓ_u est inférieure à l'inverse de la fonction conjuguée, $\ell_u(x_1,x_2)\leqslant\psi^{*-1}\big(u,\ell(x_1,x_2)\big)$, avec $\psi^*(s,\delta)=\sup_{\lambda\in I}\big(\lambda s-\psi(\lambda,\delta)\big)$ et $\psi^{*-1}(u,\delta)=\inf\big\{s\in\mathbb{R}:\psi^*(s,\delta)>u\big\}$. Alors, les notions

géométriques continuent de s'appliquer et nous retrouvons un théorème impliquant l'entropie métrique de $\mathcal{X}$ par rapport à ℓ. Nous illustrons les bénéfices de cette approche en considérant l'optimisation de formes quadratiques de processus gaussiens qui peut modéliser l'optimisation d'erreur des moindres carrés comme une vraisemblance gaussienne avec un a priori uniforme. Nous introduisons un algorithme qui dans le cas de N processus gaussiens élevés au carré obtient un regret inférieur à, avec forte probabilité :

$$R_n \leqslant \mathcal{O}\Big(N\big(\sqrt{n\gamma_n \log n} + \gamma_n\big) + \sqrt{Nn}\log n\Big).$$

4.3 Optimisation de fonctions non-régulières par ordonnancement

Les approches considérées jusqu'à présent requièrent que la fonction inconnue soit régulière. Nous introduisons dans cette thèse un nouveau cadre d'optimisation, où la fonction sous-jacente peut présenter des variations arbitraires, voire des discontinuités, mais telle que ses ensembles de niveaux sont contrôlés. On analyse dans cette partie le cas des observations non bruitées, et nous cherchons à contrôler la distance $\|x^\star - x_{\hat{i}_n}\|$ entre la prédiction et la vraie position d'un maximum global, où précisément $f(x^\star) = \sup_{x\in\mathcal{X}} f(x^\star)$ et $f(x_{\hat{i}_n}) = \max_{i=1,\dots,n} f(x_i)$. L'étendue des valeurs de la fonction étant arbitraire, les regrets simple et cumulé sont également arbitraires.

Structure d'ordonnancement

On définit $r_f : \mathcal{X}\times\mathcal{X} \to \{-1,0,1\}$ la règle d'ordonnancement induite par une fonction f comme le signe de $f(x_1) - f(x_2)$. On remarque que cette règle est stable par composition de f par n'importe quelle fonction monotone, non nécessairement continue. Cette propriété est fondamentale pour déduire des résultats de convergence. Basé sur les règles d'ordonnancement, notre algorithme n'effectue que des comparaisons entre les valeurs de la fonction et est ainsi robuste aux compositions monotones. L'algorithme possède en entrée un ensemble $\mathcal{R}$ de règles d'ordonnancement, et on suppose que $r_f \in \mathcal{R}$. Par exemple, nous considérons en particulier les règles induites par les polynômes de degré N :

$$\mathcal{R}_{\mathcal{P},N} = \Big\{r_f : f \in \mathcal{P}_N(\mathcal{X})\Big\}.$$

On note que $r_f \in \mathcal{R}_{\mathcal{P},N}$ n'implique pas que f soit un polynôme. Nous considérons également les règles pouvant être décrites par N convexes :

$$\mathcal{R}_{\mathcal{C},N} = \Big\{r : \forall x \in \mathcal{X}, \exists C_1,\dots,C_N \subset \mathcal{X},\ \big\{x' \in \mathcal{X} : r(x',x) > 0\big\} = \bigcup_{i=1}^{N} C_i,\ C_i \text{ est convexe}\Big\}.$$

Il est aisé de voir que ces règles correspondent aux fonctions dont les ensembles de niveaux sont des unions d'au plus N convexes.

L'algorithme RankOpt

A chaque itération n, la requête suivante x_{n+1} est tirée uniformément dans le sous-espace de $\mathcal{X}$ où le maximum peut se trouver sans briser l'hypothèse que $r_f \in \mathcal{R}$. Formellement, on définit

la perte empirique $L_n(r) = \frac{2}{n(n+1)} \sum_{1 \leqslant i < j \leqslant n} \mathbb{1}\{r_f(x_i, x_j) \neq r(x_i, x_j)\}$, et le sous-ensemble actif $\mathcal{R}_n = \{r \in \mathcal{R} : L_n(r) = 0\}$. L'algorithme RankOpt choisit alors :

$$x_{n+1} \sim \mathcal{U}\big(\{x \in \mathcal{X} : \exists r \in \mathcal{R}_n,\ r(x, x_{\hat{i}_n}) \leqslant 0\}\big).$$

Les bornes sur la convergence de cet algorithme s'expriment en fonction de l'étroitesse des ensembles de niveaux autour du maximum. Soit $x^\star$ la position du maximum, $\alpha \geqslant 0$ et $c_\alpha > 0$ tels que les ensembles de niveaux $f^{-1}(y) = \{x \in \mathcal{X} : f(x) = y\}$ satisfont :

$$\sup_{x \in \inf^{-1}(y)} \|x^\star - x\| \leqslant c_\alpha \inf_{x \in f^{-1}(y)} \|x^\star - x\|^{1/(1+\alpha)}.$$

On remarque que lorsque les ensembles de niveaux rétrécissent dans toute les directions avec une vitesse du même ordre, alors $\alpha = 0$. Nous prouvons la convergence suivante pour l'algorithme RankOpt, pour tout $u > 0$, avec probabilité au moins $1 - e^{-u}$:

$$\|x^\star - x_{\hat{i}_n}\| \leqslant C_\alpha \Big(\frac{u}{n}\Big)^{\frac{1}{d(1+\alpha)^2}},$$

où $C_\alpha = c_\alpha^{\frac{2+\alpha}{1+\alpha}} \Delta(\mathcal{X})^{\frac{1}{(1+\alpha)^2}}$ et $\Delta(\mathcal{X})$ est le diamètre de $\mathcal{X}$.

L'algorithme adaptatif AdaRankOpt

En pratique il n'est pas souvent possible de connaître $\mathcal{R}$ tel que $r_f \in \mathcal{R}$. Nous proposons une extension de l'algorithme précédent qui prend en entrée une suite croissante d'ensembles de règles d'ordonnancement, $\mathcal{R}_1 \subset \mathcal{R}_2 \subset \ldots$, et on suppose simplement qu'il existe un $N^\star$ inconnu tel que $r_f \in \mathcal{R}_{N^\star}$. L'algorithme AdaRankOpt est paramétré par $p \in (0, 1)$ et effectue alternativement deux tâches, avec probabilité p tirer uniformément un point dans $\mathcal{X}$ afin de déterminer $N^\star$, avec probabilité $1 - p$ tirer un point comme l'algorithme AdaRankOpt avec $\mathcal{R} = \mathcal{R}_{\min\{N : \min_{r \in \mathcal{R}_N} L_n(r) = 0\}}$ le plus petit $\mathcal{R}$ qui soit consistant avec les données. Soit $L(r) = \mathbb{P}_{X, X' \overset{\text{iid}}{\sim} \mathcal{U}(\mathcal{X})}\big[r_f(X, X') \neq r(X, X')\big]$ la perte réelle. On définit la complexité de Rademacher d'une structure d'ordonnancement $\mathcal{R}$ (Clémençon et al., 2008) comme :

$$\mathbb{E}_{X_1, \ldots, X_n \overset{\text{iid}}{\sim} \mathcal{U}(\mathcal{X})} \left[\sup_{r \in \mathcal{R}} \frac{1}{\lfloor n/2 \rfloor} \left| \sum_{i=1}^{\lfloor n/2 \rfloor} \epsilon_i \mathbb{1}\Big\{ r_f(X_i, X_{\lfloor n/2 \rfloor + i}) \neq r(X_i, X_{\lfloor n/2 \rfloor + i}) \Big\} \right| \right].$$

Alors, lorsqu'il existe $V > 0$ tel que la complexité de Rademacher de $\mathcal{R}_{N^\star - 1}$ est bornée par $\sqrt{V/n}$, on prouve avec probabilité au moins $1 - e^{-u}$:

$$\|x^\star - x_{\hat{i}_n}\| \leqslant C_\alpha \Big(\frac{u + \log 2}{n - n_u}\Big)^{\frac{1}{d(1+\alpha)^2}},$$
$$n_u = \Big\lfloor 10 \frac{V + u + \log 4}{p \inf_{r \in \mathcal{R}_{N^\star - 1}} L(r)^2} \Big\rfloor.$$

Nous présentons également dans ce document des formulations alternatives des concepts introduits afin de faciliter l'implémentation pratique de ces algorithmes. Nous comparons ensuite empiriquement leurs performances face à des compétiteurs classiques.

4.4 Applications

Les thématiques de recherche de la thèse ont été constamment menées avec des perspectives d'applications réelles. Nous présentons dans les paragraphes suivants deux études en mécanique des fluides.

Phénomènes d'amplification de tsunami

Les contributions sur l'optimisation bayésienne par batch ont été motivées par une collaboration avec des chercheurs sur les tsunamis. Grâce à un code numérique nous avons simulé l'impact d'une île sur la vague d'un tsunami, certaines configurations pouvant l'amplifier. En optimisant les paramètres géométriques de l'île par rapport au ratio d'amplification nous avons découvert quel est le pire cas, ce qui apporte une information cruciale sur ce phénomène. Nous avons utilisé le nouvel algorithme par batch pour calculer plusieurs simulations en parallèle et gagner un temps considérable.

Séries de convertisseurs d'énergie des vagues

Cette étude analyse les configurations spatiales de convertisseurs d'énergie des vagues. Des séries de tels convertisseurs se trouvent proche des côtes pour produire de l'électricité. La position de ces appareils les uns par rapport aux autres a un impact important sur l'énergie totale produite puisque des interférences peuvent avoir lieu. Notre but est d'optimiser les coordonnées en x et y de 40 appareils. L'énergie totale produite est calculée grâce à des simulations numériques. Comme la dimension de l'espace de recherche est grande et qu'une seule simulation demande deux semaines, nous devions considérer des approximations de l'objectif. Nous avons proposé avec succès une relaxation du problème utilisant le nouvel algorithme d'optimisation introduit précédemment.

Publications associées

- Sarkar, D., Contal, E. Vayatis, N., and Dias, F. (2016). Prediction and optimization of wave energy converter arrays using a machine learning approach. *Renewable Energy*, 97 :504–517.
- Malherbe, C., Contal, E., and Vayatis., N. (2016). A ranking approach to global optimization. In *Proceedings of the 33st International Conference on Machine Learning (ICML)*.
- Contal, E., Malherbe, C., and Vayatis, N. (2015). Optimization for gaussian processes via chaining. *NIPS Workshop on Bayesian Optimization*.
- Sarkar, D., Contal, E., Vayatis, N., and Dias, F. (2015). A machine learning approach to the analysis of wave energy converters. *Proceedings of the 34th International Conference on Ocean, Offshore and Arctic Engineering (OMAE)*.
- Contal, E., Perchet, V., and Vayatis, N. (2014). Gaussian process optimization with mutual information. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*.
- Stefanakis, T. S., Contal, E., Vayatis, N., Dias, F., and Synolakis, C.E. (2014). Can small islands protect nearby coasts from tsunamis ? An active experimental design approach. *Proceedings of the Royal Society of London A : Mathematical, Physical and Engineering Sciences*, 470(2172).
- Contal, E., Buffoni, D., Robicquet, A., and Vayatis, N. (2013). Parallel Gaussian process optimization with upper confidence bound and pure exploration. In *Proceedings of the European Conference on Machine Learning (ECML)*.

1 Introduction

This chapter introduces the main motivations and context of the work presented in the document. A review of global optimization methods and relevant bandit algorithms is provided, and the main contributions of the thesis are summarized.

Contents

1.1 Context of the Thesis . . . 22
1.2 Related Work . . . 22
1.2.1 A Short History of Optimization of Expensive Black-Box Functions . . . 23
Response Surface Methodology . . . 23
Lipschitzian Optimization . . . 23
Bayesian Optimization . . . 23
1.2.2 Theoretical Analysis with the Bandit Framework . . . 24
Multi-armed Bandits . . . 24
Continuous Bandits . . . 24
Gaussian Processes . . . 25
1.3 Contributions . . . 25
1.3.1 Batch Bayesian Optimization . . . 25
An Efficient Algorithm for Parallel Queries . . . 25
Theoretical Guarantees on the Convergence Speed . . . 25
1.3.2 Bayesian Optimization in Metric Spaces . . . 26
Adaptive Partitioning Trees and Generic Chaining . . . 26
Upper and Lower Bounds . . . 26
Beyond Gaussian Processes . . . 26
1.3.3 Non-Smooth Optimization by Ranking . . . 26
Function-Value Free Optimization . . . 26
Novel Optimization Framework . . . 26
1.3.4 Applications and Efficient Implementations . . . 27
Tsunami Amplification Phenomena . . . 27
Wave Energy Converters Array . . . 27
Efficient Implementations . . . 27
1.4 Outline . . . 27

1.1 Context of the Thesis

This dissertation is dedicated to a rigorous analysis of sequential global optimization algorithms. Sequential global optimization is encountered in numerous domains including natural sciences (Floudas and Pardalos, 2000), engineering design (Wang and Shan, 2007), bioinformatics (Moles et al., 2003), finance (Ziemba and Vickson, 2006) and many others. It aims at finding the input of a given system optimizing the output. The optimization could be for instance the maximization of a *reward*, or the minimization of a cost. The function which links the input to the output is not explicit, but we are provided a way to evaluate the output for any input. Measurements could come from laboratory experiments, numerical simulations, sensors responses or any feedback depending on the application. In particular this function is not supposed to be convex and may display many local optima. In this work we tackle the challenging case where the *evaluations are expensive*, which requires to design a careful selection of the input to evaluate. In this view, an iterative procedure uses the previously acquired measures to choose the next query predicted to be the most useful. We study two different goals, either to maximize the sum of the rewards received at each iteration, which is relevant for "online" optimization such as clinical trials or recommendation systems; or to maximize the best reward found so far, relevant for optimum search such as numerical optimization. The present thesis stresses on the *theoretical properties* of the proposed methods. The numerical complexity is often a critical concern for a practitioner, we consistently come up with tractable solutions throughout this work and provide implementation details. The objective is to bring new concepts from the theory aiming at describing the efficiency of the optimization procedures with respect to generic notions of complexity of the problem. This permits to develop novel algorithms with guaranteed performance in well defined settings.

1.2 Related Work

The field of optimization theory encompasses lots of various frameworks, this thesis comprises the following fields:

- multi-armed bandits,
- Lipschitzian optimization,
- Bayesian optimization.

In what follows, we present a short historical perspective on the closely related works. For related matters but with different settings, we refer to the works of Boyd and Vandenberghe (2004) and Bubeck (2015) on gradient-based approaches when the function is convex, which are not suited for global optimization since the knowledge of a local minimum does not permit to control the regret; the works and reviews of Sebag and Ducoulombier (1998), Garnier and Kallel (2000) and Eiben and Smith (2003) on evolutionary algorithms when the evaluations are not expensive, which do not provide guarantees on the regret; and the works of Papadimitriou and Steiglitz (1982) and Garnier and Kallel (2001) for combinatorial optimization.

1.2.1 A Short History of Optimization of Expensive Black-Box Functions

Response Surface Methodology

One of the most standard way to deal with this problem is to build and maintain a *surrogate function* from the measurements. This surrogate function is computed to fit the observations and generalize to any unknown input. This step can be thought of as model estimation like regression, especially when the observations are noisy. It traditionally uses polynomial interpolation or kernel-based regression with least-squares fitting term. Evaluations of this surrogate are instantaneous. It is therefore easy to use this empirical estimation to select which input to evaluate next. This technique is referred as the Response Surface Methodology (Myers and Montgomery, 1995; Jones, 2001) and was introduced in Box and Wilson (1951). The major drawbacks of the above are first that model selection is a crucial and tricky subject, second that the presence of local minima is troublesome for approaches that do not include global exploration criterion, and third that it is not adequate for a complete theoretical analysis.

Lipschitzian Optimization

Motivated by the objective of proving global convergence of optimization algorithms, few techniques emerged later using the fact that the unknown function satisfies a *smoothness* condition. It is assumed in this respect that this function is Lipschitz-continuous, that is its gradient is bounded. Knowing a set of values of the function and the bound on its gradient, one can safely remove from the search space the region where the optimum cannot lie without breaking the Lipschitz property. By sampling in the remaining region, it is easy to obtain theoretical guarantees of convergence. This idea goes back to the Shubert-Mladineo algorithm (Shubert, 1972; Mladineo, 1986). Yet, the knowledge of the Lipschitz constant is often not realistic. Adaptive algorithms estimate this constant on the data acquired during the optimization. However they may produce bad convergence speed when the Lipschitz constant is found to be very large because of an isolated rough pattern in the function, which could be an outlier not relevant for the optimization task. The DiRect algorithm (Jones et al., 1993) solves this by using an improved dichotomy search which does not require to estimate the Lipschitz constant. The robustness of the resulting method makes it still a commonly used choice today for noiseless global optimization.

Bayesian Optimization

A more modern framework, Bayesian optimization, introduced in Kushner (1964) and Močkus (1974), overcomes some of the former issues and easily adapts to observation noise. With the prior assumption that the unknown underlying function is a realization of some *stochastic process*, it is possible to compute a *posterior distribution* given the acquired data, from which one deduces expectations and uncertainties for unknown inputs. We are then interested in the expected behavior of an optimization procedure where the function is stochastic, or to get stronger results we aim at proving properties that hold *with high probability*. By far the most common prior distributions are Gaussian processes. The smoothness of the covariance induces an assumption on the smoothness of the function by enforcing near-by locations to have correlated values. Yet, this is a less stringent hypothesis than a bound on the Lipschitz

constant since it suffices that the constrains hold with high probability, and not necessarily everywhere. Optimization strategies may then use confidence intervals (Cox and John, 1997; Srinivas et al., 2012), integrated criterion like the Expected Improvement (Jones et al., 1998) or Expected Information Gain (Hennig and Schuler, 2012). See Brochu et al. (2010) for a review of various acquisition functions.

1.2.2 Theoretical Analysis with the Bandit Framework

Multi-armed Bandits

The theoretical analysis of algorithms aiming at maximizing the cumulative reward is often presented under the multi-armed bandit framework. This model considers that a player is presented K arms, the possible inputs. When she chooses an arm, she receives a noisy reward sampled independently from the previous rewards, from a distribution depending on the chosen arm. In order to maximize the rewards, one has to deal with the *exploration-exploitation tradeoff*. The first Bayesian algorithm in this view goes back to Thompson (1933), and the first Bayesian analysis to Gittins (1979). The *cumulative regret*, the sum of the differences between the rewards obtained and the optimum reward, has been considered in depth in Lai and Robbins (1985) where a general lower bound is presented. Contemporary proof techniques lead to finer upper and lower bounds on the cumulative regret. For multi-armed bandit in the frequentist setting, Auer et al. (2002) give a finite time analysis when the rewards are bounded, and Cappé et al. (2013) propose an asymptotically optimal algorithm for more general distributions of rewards. For the Bayesian aspect Kaufmann et al. (2012) give an asymptotically optimal algorithm for binary rewards. In the bandit framework, the problem of identifying the best input is analyzed using the *simple regret*, that is the difference between the best reward obtained and the optimum. Mannor and Tsitsiklis (2004) and Even-Dar et al. (2006) prove both lower and upper bounds on the number of iterations needed to obtain a given simple regret with given probability. Bubeck et al. (2009) explore the link between low cumulative regret and low simple regret, and show that it is impossible to have both together. Audibert et al. (2010) analyze the problem of identifying exactly the best arm with a fixed budget and exhibit an almost optimal algorithm. Finally, Kaufmann et al. (2016) propose an asymptotically optimal algorithm which finds the best arms with a fixed confidence.

Continuous Bandits

The recent literature considered the extension to continuous settings under relaxed Lipschitz assumptions. For the cumulative regret, Kleinberg (2004) focused on the one-dimensional case and prove almost tight upper and lower bounds under Hölder continuity. Auer et al. (2007) improve this assumption by considering only local smoothness. Dani et al. (2008) provide tight lower and upper bounds in the linear case, and Rusmevichientong and Tsitsiklis (2010) show Bayesian counterparts. And Bubeck et al. (2011) extend the algorithms for more generic metric spaces with local constrains and define a new complexity dimension. With respect to the simple regret analysis, Kleinberg et al. (2008) suggest a similar approach with a closely related complexity dimension. Munos (2011) studies the case where the algorithm is agnostic to the smoothness of the function but no noise affects the observations. Bull (2015) introduces an almost optimal algorithm with noisy observations and only little assumptions

on the smoothness. Finally, Grill et al. (2015) combine previous methods and weaken the smoothness assumption.

Gaussian Processes

The theoretical study of Bayesian optimization is a newer interest. We refer to Srinivas et al. (2012) for analysis of the cumulative regret, where the techniques from bandit theory and information theory are leveraged and adapted to the case of Bayesian optimization. For the simple regret with deterministic observations (no noise), Grünewälder et al. (2010) give an optimal algorithm with a fixed budget, but with low practical feasibility. Bull (2011) provides an analysis of the Expected Improvement criterion, and de Freitas et al. (2012) show that the deterministic context allows one to obtain exponential convergence rate with additional smoothness conditions. The previous approach cannot be used in practice, and Wang et al. (2014) overcome this impracticability by combining the previous works along the same lines as Grill et al. (2015) from the previous paragraph.

1.3 Contributions

The core axis of this work focuses on Bayesian optimization with Gaussian processes with noisy observations, a setting similar to Srinivas et al. (2012). We first analyze extensions of the sequential optimization procedure where the evaluations are acquired over successive batches. We then present a novel approach to compute upper confidence bounds in metric spaces, which adapts to arbitrary smoothness and allows to design generic algorithms with state-of-the-art regret bounds. Next, we introduce a novel optimization framework where the function is not supposed to be smooth, but satisfies some conditions on the inclusion of its level sets. Finally, we present contributions for several applications and practical considerations.

1.3.1 Batch Bayesian Optimization

The first contribution presented in this thesis is a global optimization algorithm querying a batch of evaluations at each iteration instead of a single one. This approach accounts for the case where the evaluations are acquired in *parallel* for the cost of a single iteration. The typical examples are numerical optimization with a cluster of computing machines, or optimization of sensor measurement based on several sensors.

An Efficient Algorithm for Parallel Queries

We introduce an efficient algorithm for this problem. We use a modification of the GP-UCB algorithm from Srinivas et al. (2012) together with lower confidence bounds to focus the evaluations in a *relevant region*. This relevant region is defined as the set of inputs having a sufficiently large posterior probability of producing a value at least as high as the maximum observed so far.

Theoretical Guarantees on the Convergence Speed

Our approach follows the lines of Desautels et al. (2012), but the use of the relevant region leads to finer results on the regret bounds. We prove a convergence rate and upper bounds on the cumulative regret of our algorithm. We establish that the cumulative regret of our

strategy is similar, up to constants, to the one of the sequential algorithm that reads directly the observations. When the cost of a batch of K query is the same as the cost of a single query, we show that our algorithm converges faster that the state-of-the-art with an improvement of $\sqrt{K}$ in typical scenarios.

1.3.2 Bayesian Optimization in Metric Spaces

Upper confidence bounds are at the center of the GP-UCB algorithm and many other global optimization schemes. Computing confidence bounds on a stochastic process which holds everywhere is not an easy task when the input space is continuous. One can think of estimating the maximum of an infinite number of correlated random variables.

Adaptive Partitioning Trees and Generic Chaining

The classical method found in the optimization literature, such as Srinivas et al. (2012) and related works, is to build a finite *discretization* of the input space and then to compute upper bounds for every individual discrete input. This is often unsatisfactory since the impact of the approximation on the global optimization results is delicate, making the choice of the sharpness of the approximation not clear. We solve this question by introducing partitions built in a greedy and adaptive fashion.

Upper and Lower Bounds

We prove later that these partitions are optimal up to constant multiplicative factors, which permits to derive tight lower and upper bounds for the stochastic process. This result holds for arbitrary metric spaces, potentially *nonparametric*, without any additional assumption. Our algorithm is inspired by the theory of stochastic processes, and enjoys efficient practical implementations.

Beyond Gaussian Processes

Finally, we show that our techniques can be extended to more complex stochastic processes. We expose that popular optimization settings are not captured by Gaussian processes, like optimization of quadratic forms. We then exhibit a modification of the previous algorithm that adapts to these sophisticated Bayesian priors, and enjoys similar guarantees on the convergence rates.

1.3.3 Non-Smooth Optimization by Ranking

All the previously mentioned approaches do not hold when the underlying function is not smooth around its optimum, or even discontinuous.

Function-Value Free Optimization

To alleviate this constraint, we give a global optimization algorithm which is *function-value-free* and still presents efficient theoretical guarantees for deterministic optimization. This algorithm only relies on pairwise comparisons.

Novel Optimization Framework

We define in this respect a notion of *ranking structure*, a condition on the level sets of the unknown function. Our strategy is then to select the queries in the relevant region. In this

non-Bayesian setting the relevant region is the set of inputs for which a function whose ranking structure is consistent with the observations produces a value at least as high as the best point seen so far. We provide here convergence rates on the distance between the best query and the location of the true optimum.

1.3.4 Applications and Efficient Implementations

The research topics of this thesis have been consistently driven by *real-life scenarios*. The following paragraphs present two projects from fluid mechanics which motivated the previous contributions. Then, a short description of practical implementations is given.

Tsunami Amplification Phenomena

The work on batch Bayesian optimization comes from a collaboration with researchers on tsunami analysis. We use a numerical code to simulate the impact of an island on the wave of a tsunami. Some particular configurations may amplify the tsunami. By optimizing the geometrical parameters of the island with respect to the amplification ratio, we discover what is the worst case, and therefore obtain crucial knowledge on this phenomenon. We used the novel batch algorithm to perform several simulations in parallel, leading to a significant gain of time.

Wave Energy Converters Array

This project studies the spatial configuration of Wave Energy Converters. Groups of such devices are found in the sea near the coast and produce electricity using the waves. The positions of the devices with respect to each other have a significant impact on the total energy produced, since interference may happen. Our aim is to optimize the x and y coordinates of 40 Wave Energy Converters. The total energy is computed using numerical simulations. Since the dimension of the search space is large and a single simulation requires two weeks, we had to carefully approximate the objective function to simplify the setting. We propose a successful *relaxation* of this problem using the novel global optimization approaches introduced above.

Efficient Implementations

Since the aim of the research presented in this thesis is to tackle real challenges, the practicability of the proposed solutions has a significant impact. The new algorithms we derive are consistently implemented and empirically assessed. In order to render the execution possible on casual laptops, a lot of care is taken to perform the computation in an efficient way. When *approximations* are necessary, we consider the heuristic as a part of our algorithm and analyze the repercussion on the theoretical guarantees. All the code from the contributions in Bayesian optimization is available as Matlab and Python packages.

1.4 Outline

The core of this dissertation is organized as follows.
In Chapter 2, we formalize the problem of sequential global optimization. We describe precisely the *mathematical foundations* for both the noisy case and the deterministic case, and both the Bayesian and non-Bayesian approach. We review some known results from the literature having a crucial importance for the following.

In Chapter 3, we focus on *novel advances* for Bayesian optimization. We first present results for the batch setting, and then continue on the extension to arbitrary metric spaces. We prove *theoretical guarantees* of performance, and perform numerical experiments. Finally, we show that Bayesian optimization is well suited for distribution beyond Gaussian processes, and we outline a novel algorithm with an example application.
In Chapter 4, we introduce a new framework for deterministic optimization of *non-smooth* functions. We express the difficulty of the optimization problem with respect to *ranking structures*. We first propose an efficient solution knowing the ranking structure of the underlying function, and then show that this can be adapted when we remove this specification while the theoretical properties are preserved.
Chapter 5 is dedicated to results obtained for various *applications* of the previous work. We detail computation techniques leveraged to get efficient implementations. We then describe applications in fluid mechanics and computational chemistry.
Finally, in Appendix A, we present an attempt of proof techniques to obtain improved cumulative regret. Unfortunately, these results cannot be applied within the standard setting of Bayesian optimization. Even if the theoretical guarantees do not hold, the algorithm we derive exhibits fast convergence toward the optimum on our empirical assessments.

Sequential Global Optimization

2

This chapter presents the state-of-the-art of the theoretical advances in sequential global optimization. In Section 2.1, we first define the setting and objectives, and show relevant properties. We then review, in Section 2.2, existing algorithms from multi-armed bandits, linear bandits, Lipschitzian optimization and Bayesian optimization. We provide available guarantees and impossibility results for those multiple frameworks.

Contents

2.1 Problem Formulation and Fundamental Ingredients **31**
2.1.1 Sequential Optimization via Noisy Measurements 31
The Input Space 31
The Gaussian Noise Model 31
2.1.2 Cumulative and Simple Regrets 32
Simple Regret 32
Cumulative Regret 32
2.1.3 Smoothness, Metric Spaces and Covering Dimensions 33
Fixed Function in Metric Spaces 33
Covering Dimension and Metric Entropy 34
Why this Abstraction is Useful 35
Near-Optimality Dimension 35
Illustrative Examples of Zero Near-Optimality Dimensions 36
2.1.4 Bayesian Assumption, Gaussian Processes and Continuity 36
Mathematical Foundations of Stochastic Processes 37
Introduction to Gaussian Processes 37
Posterior Distribution and Bayesian Inference 39
Stochastic Smoothness of Gaussian Processes 40
Reproducing Kernel Hilbert Spaces 41
2.1.5 Practical and Theoretical Properties of Priors and Posteriors 43
Design of the Prior 43
Empirical Confidence Intervals 43
Likelihood Maximization and Frequentist Properties of Bayesian Inference 44
2.2 Optimization Algorithms and Theoretical Results **45**
2.2.1 Stochastic Multi-Armed Bandits 45
Lower Bounds on the Cumulative Regret 45
Asymptotically Optimal Algorithms for the Cumulative Regret 46
High Probability Guarantees 48
Lower Bounds on the Simple Regret 48
Optimal Algorithms for Pure Exploration Problems 49
2.2.2 Stochastic Linear Bandits 50
Algorithms and Logarithmic Upper Bounds on the Regret 50
Polynomial Regret and Lower Bounds 51
Upper and Lower Bounds for the Simple Regret 51
2.2.3 Lipschitzian Optimization 51
One-Dimensional Spaces 52
Multi-dimensional Spaces 52
Partitioning Trees 53
Unknown Smoothness and Adaptive Algorithms 55

2.2.4 Bayesian Optimization and Gaussian Processes 56
Simple Regret for Gaussian Processes with Deterministic Observations 56
Simple and Cumulative Regrets with Noisy Observations 57

2.1 Problem Formulation and Fundamental Ingredients

In this first section, we present the setup for optimization which is considered in this thesis. Although the primary objective of this section is theoretical, we attempt to give insights and intuitions on every notion involved. This part of the thesis does not contain real mathematical contributions, yet, some concepts are defined using novel presentations to encompass multiple existing works in the same framework. Here and in what follows, the $\triangleq$ symbol means *equal by definition* and the $\equiv$ symbol denotes simplifications of notations. The expectation of a random variable X is written $\mathbb{E}[X]$. For real numbers a and b, $a \wedge b$ (resp. $a \vee b$) denotes the minimum (resp. maximum) of a and b, and $(a)_+ \triangleq 0 \vee a$.

2.1.1 Sequential Optimization via Noisy Measurements

The Input Space

The system to be optimized is modeled by an unknown function $f : \mathcal{X} \to \mathbb{R}$. The input set $\mathcal{X}$ is the search space, which could be either finite or infinite, parametric or nonparametric. In the K-armed bandit framework $\mathcal{X}$ is the set of all arms, that is its cardinality is K and it has no particular structure. For continuous optimization of d parameters, $\mathcal{X}$ is typically a subset of $\mathbb{R}^d$. Less common optimization settings also fit in our model, like optimization over structured spaces such as graphs or shapes. It is easy to define constrained optimization problems from a mathematical point of view, since it suffices to restrict $\mathcal{X}$ to the set of points which satisfy the constraints.

The Gaussian Noise Model

An optimization algorithm is given the ability to query the function at any point $x \in \mathcal{X}$, to receive the associated noisy evaluation $y \triangleq f(x) + \epsilon$, where ϵ models an independent centered additive noise, that is $\mathbb{E}[y] = f(x)$. We will refer to the case without noise ($\epsilon = 0$ almost surely) as deterministic optimization. These evaluations are supposed to be expensive, the optimization procedure should use the least possible amount of queries to attain its goal. The additive noise can describe different scenarios. First the noise may come from the process of acquisition of the observations. For example if one uses sensor responses, the sensor itself may produce noisy outputs. Similarly, if one computes simulations performing numerical approximations, the software used may invoke randomness such as MCMC solvers. Finally, even outside these cases practitioners often consider that the output is noisy, in order to render the algorithms robust again model misspecification. The Gaussian distribution with known variance is the most common assumption for the distribution of the noise, although some variants appear in the literature such as bounded noise with unknown variance or any subgaussian distribution. Formally, an optimization algorithm $\mathcal{A} : \bigcup_{n\geqslant 0} (\mathcal{X} \times \mathbb{R})^n \to \mathcal{X}$ is a function taking as input the history of queries and noisy observations $(x_1, y_1, \dots, x_n, y_n)$ and returning the next query $x_{n+1} \in \mathcal{X}$. In the sequel we denote $(\mathcal{F}_n)_{n\geqslant 1}$ the filtration generated by the history of available information:

$$\mathcal{F}_n \triangleq \sigma(x_1, y_1, \dots, x_n, y_n). \tag{2.1}$$

Here and in what follows, y_n will denote $y_n \equiv f(x_n) + \epsilon_n$. The noise variables $(\epsilon_n)_{n\geqslant 0}$ are independent and centered, and x_{n+1} is $\mathcal{F}_n$-measurable.

The Gaussian assumption on the noise is arbitrary to some extent, but one may argue that it is the natural approach. First, this distribution maximizes entropy at given variance. Second, by the central limit theorem any normalized average of numbers of independent noises converges to a Gaussian distribution. Knowing the variance of the noise is required by the theoretical analysis. In practice, this is often a hyper-parameter selected with cross-validation (see Section 2.1.5).

2.1.2 Cumulative and Simple Regrets

Depending on the application, the aim of the optimization algorithm may vary. We consider in this dissertation two different goals. The first objective is to find the best input as fast as possible. The second is to maximize the sum of the rewards. We refer to Bubeck and Cesa-Bianchi (2012) for additional discussions on these objectives.

Simple Regret

Let $\mathcal{A}$ be an optimization algorithm and $x_1, \dots, x_n$ its queries until iteration n, that is $x_1 = \mathcal{A}(\varnothing), x_2 = \mathcal{A}\big((x_1, y_1)\big), x_3 = \mathcal{A}\big((x_1, y_1), (x_2, y_2)\big)$ and so on. The simple regret is defined as the difference in function value between the unknown optimum and the best point found so far:

$$S_n \equiv S_n(\mathcal{A}, f, \epsilon, \mathcal{X}) \triangleq \sup_{x^\star \in \mathcal{X}} f(x^\star) - \max_{i \leqslant n} f(x_i). \tag{2.2}$$

This quantity is not available in practice, the purpose of the theoretical analysis is to prove convergence speed toward zero for $\mathcal{A}$ according to the properties of f and $\mathcal{X}$. Remark that the simple regret uses only function values, so the eventual presence of multiple global maxima in f does not perturb our setting. This evaluation is well suited for problems similar to numerical optimization, where all the (computational) cost of each query is fixed and independent from the output. The typical example is hyper-parameter optimization in Machine Learning (Snoek et al., 2012). Some authors define the simple regret differently (Audibert et al., 2010; Bubeck et al., 2011), for instance the optimization algorithm may be allowed to output two locations at each iterations, the queries x_n and the guess x_n^g. In this framework the observations are given for x_n only, and the simple regret is taken using x_n^g. We do not consider this alternative definition to simplify the sequel.

Cumulative Regret

Using the same notations as above, the (pseudo-) cumulative regret is defined as the sum of the differences between the true optimum and the queried points:

$$\begin{aligned} R_n \equiv R_n(\mathcal{A}, f, \epsilon, \mathcal{X}) &\triangleq \sum_{i=1}^{n} \left(\sup_{x^\star \in \mathcal{X}} f(x^\star) - f(x_i) \right) \\ &= n \sup_{x^\star \in \mathcal{X}} f(x^\star) - \sum_{i=1}^{n} f(x_i). \end{aligned} \tag{2.3}$$

Generally this quantity does not converge, and we are interested in proving sublinear growth. It is used to model online settings where the cost of evaluations is the regret. The typical example is clinical trials, where we aim at minimizing the total wrong decisions. This also encompasses other settings such as online advertising and recommendation. Note that one

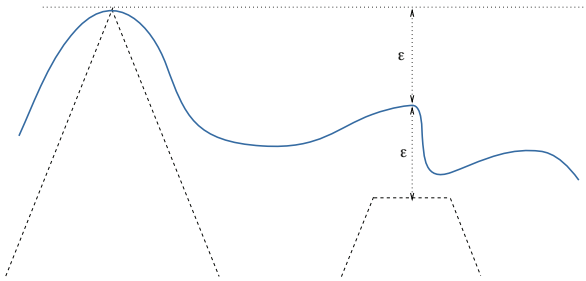


Figure 2.1. – Example of function satisfying the weak-Lipschitz assumption (Bubeck et al., 2011)

can deduce convergence rate of the simple regret from upper bounds on the cumulative regret, since $S_n \leqslant n^{-1} R_n$. The obtained convergence speed on S_n cannot be faster than $\mathcal{O}(n^{-1})$, which is not a limitation in the noisy case since the presence of noise typically leads to simple regret at least in $\mathcal{O}(n^{-1/2})$.

2.1.3 Smoothness, Metric Spaces and Covering Dimensions

We first define the non-Bayesian optimization framework. Here the unknown function f is assumed to belong to a known class $\mathcal{C}$ of functions satisfying a certain smoothness condition. The aim of this section is to define precisely the various notions of smoothness that we will subsequently use. For an algorithm $\mathcal{A}$, we are looking for results under the form:

$$\forall f \in \mathcal{C},\ \forall u > 0,\ \mathbb{P}\Big[\forall n \geqslant 1,\ R_n \leqslant U_1(\mathcal{X}, \mathcal{C}, n, \eta, u)\Big] \geqslant 1 - e^{-u},$$

with U_1 as small as possible, or results for the expected regret:

$$\forall f \in \mathcal{C},\ \forall n \geqslant 1,\ \mathbb{E}\big[R_n\big] \leqslant U_2(\mathcal{X}, \mathcal{C}, n, \eta),$$

and similarly for the simple regret S_n. Here the randomness comes from the observation noise only.

Fixed Function in Metric Spaces

In the classical global optimization setting, either the input space $\mathcal{X}$ is finite, or the unknown function f is assumed to belong to a restricted functional space. If f was arbitrary on non-finite $\mathcal{X}$, then any optimization algorithm would need an infinite amount of queries to be certain about the location of the optimum. Lipschitz optimization considers that f is Lipschitz-continuous for a metric ℓ such that $(\mathcal{X}, \ell)$ is totally bounded, that is it exists a constant $K \in \mathbb{R}$ such that for all $x_1, x_2 \in \mathcal{X}$:

$$\big|f(x_1) - f(x_2)\big| \leqslant K\ell(x_1, x_2), \tag{2.4}$$

and we define $\|f\| \equiv \|f\|_{\text{Lip}}$ to be the smallest such constant. The typical example is the euclidean d-parametric hyper-cube with $\mathcal{X} = [-B, B]^d$ and $\ell(x_1, x_2) = \|x_1 - x_2\|_2$. This generic definition allows to define smoothness in nonparametric spaces. A common extension

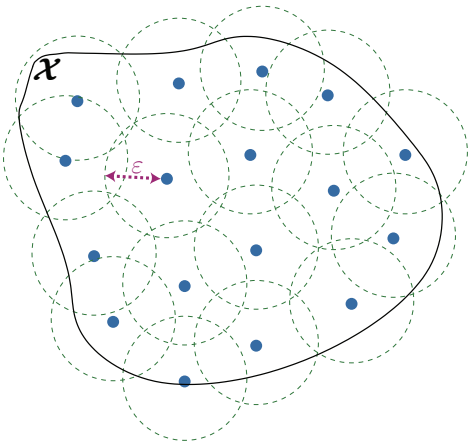


Figure 2.2. – An ε-net for the Euclidean metric that needs 17 points to cover $\mathcal{X}$

of Lipschitz-continuity is Hölder-continuity of order $\alpha > 0$, as in Kleinberg (2004), which enforces:

$$|f(x_1) - f(x_2)| \leqslant \|f\| \, \ell(x_1, x_2)^\alpha.$$

The parameter α controls the smoothness of f. We often do not require that ℓ is actually a metric but only a pre-metric or any positive "similarity measure" that satisfies $\ell(x_1, x_1) = 0$, and can include the parameter α directly in ℓ. Some authors (Bubeck et al., 2008) weaken this assumption for points away from an optimum. For $x^\star \in \mathcal{X}$ such that $f(x^\star) = \sup_{x \in \mathcal{X}} f(x)$, they require the following one-sided inequality called weak-Lipschitz smoothness:

$$f(x^\star) - f(x_1) \leqslant f(x^\star) - f(x_2) + \max\Big\{ f(x^\star) - f(x_2), \|f\| \, \ell(x_1, x_2) \Big\}. \tag{2.5}$$

A illustration of this constraint is shown in Figure 2.1. Other authors (such as Munos (2011)) only require a local constraint around the optimum:

$$f(x^\star) - f(x_1) \leqslant \|f\| \, \ell(x^\star, x_1). \tag{2.6}$$

Covering Dimension and Metric Entropy

Remark that if $(\mathcal{X}, \ell)$ is not totally bounded, such as $\mathbb{R}^d$ for the euclidean metric, then it is easy to see that any algorithm cannot approach the optimum of all functions satisfying Eq. 2.4 in a finite number of queries. In the following, we denote by $\mathcal{B}(x, \varepsilon)$ the ℓ-ball of radius ε centered in x, and $\Delta(\mathcal{X})$ the ℓ-diameter of $\mathcal{X}$:

$$\begin{aligned} \mathcal{B}(x, \varepsilon) \equiv \mathcal{B}(x, \varepsilon, \mathcal{X}, \ell) &\triangleq \big\{ x' \in \mathcal{X} : \ell(x, x') \leqslant \varepsilon \big\}, \\ \Delta(\mathcal{X}) \equiv \Delta(\mathcal{X}, \ell) &\triangleq \sup_{x_1, x_2 \in \mathcal{X}} \ell(x_1, x_2). \end{aligned}$$

The covering numbers are useful to measure the "size" of the search space. They are defined as the minimal number of ℓ-balls of given radius needed to cover $\mathcal{X}$,

$$N(\varepsilon) \equiv N(\varepsilon, \mathcal{X}, \ell) \triangleq \inf \Big\{ |T| : T \subseteq \mathcal{X}, \, \forall x \in \mathcal{X}, \, \exists t \in T, \; x \in \mathcal{B}(t, \varepsilon) \Big\}. \tag{2.7}$$

Thanks to the Lipschitz assumption, these numbers are the minimal numbers of queries required to approximate f everywhere with a precision of $\varepsilon \|f\|$. The logarithm of the covering number,

$$H(\varepsilon) \equiv H(\varepsilon, \mathcal{X}, \ell) \triangleq \log N(\varepsilon, \mathcal{X}, \ell), \tag{2.8}$$

is called the metric entropy of $\mathcal{X}$. The covering dimension (Kleinberg et al., 2008) can be defined using the metric entropy as follows.

Definition 2.1 (Covering Dimension)**.** *The covering dimension is the smallest scalar d such that the rate of the ratio of the metric entropy by $\log \varepsilon^{-1}$ is d, precisely:*

$$\dim(\mathcal{X}, \ell) \triangleq \inf \Big\{ d \in \mathbb{R} : \exists c \in \mathbb{R},\ \forall \varepsilon > 0,\ H(\varepsilon) \leqslant c - d \log \varepsilon \Big\}.$$

In the sequel, we say that the dimension is *attained* when the infimum is attained. The concept of covering dimension is closely related to the Minkowski or Hausdorff dimensions. For the d-parametric hyper-cube and the euclidean metric we have $\dim(\mathcal{X}, \ell) = d$, and this definition extends naturally to any totally bounded metric spaces.

Why this Abstraction is Useful

Although it may be difficult to imagine at this stage, we have found that the above definitions may provide a solid basis for significant methodological improvements with impact on real-life optimization problems. It is a convenient way to incorporate structural knowledge of the system directly in the similarity ℓ, without modifying the optimization algorithm. For example with $\mathcal{X} = \mathbb{R}^d$, knowing that the unknown function is symmetric or periodic can be expressed by taking an appropriate similarity ℓ. The resulting covering numbers become much smaller than with the euclidean distance, that is the optimization procedure will accumulate more information at each query. As a second example, take for instance the optimization of a shape defined by a quasi-convex path in the two-dimensional plane, such as an airfoil. A naive representation would be to discretize the shape with m two-dimensional points and use the euclidean distance in $\mathbb{R}^{2m}$. When taking a similarity ℓ that encodes invariance by permutation of the two-dimensional points, one reduces the covering numbers by a factor of $m!$. Furthermore, a nonparametric ℓ such as transportation distances may be invoked to optimize the shape with arbitrarily fine discretization without increasing the covering dimension.

Near-Optimality Dimension

Recent literature in optimization (Bubeck et al., 2011; Munos, 2011; Grill et al., 2015) shows that measuring the local behavior around the optimum is enough to prove convergence rates. For a fixed function f, calling $\mathcal{X}_\varepsilon \equiv \mathcal{X}_\varepsilon(f, \mathcal{X})$ the level set of ε-optimal points:

$$\mathcal{X}_\varepsilon \triangleq \Big\{ x \in \mathcal{X} : f(x) \geqslant \sup_{x^\star \in \mathcal{X}} f(x^\star) - \varepsilon \Big\}, \tag{2.9}$$

we can define the *near-optimality dimension* which measures the growth of the metric entropy of the near-optimal sets.

Definition 2.2 (NEAR-OPTIMALITY DIMENSION). *The near-optimality dimension* $\dim_\rho$ *of parameter* $\rho > 0$ *is the smallest scalar* d *such that the metric entropy of* $\mathcal{X}_\varepsilon$ *with radius* $\rho\varepsilon$ *is smaller that* $d \log \varepsilon^{-1}$. *Precisely,*

$$\dim_\rho(\mathcal{X}, \ell) \equiv \dim_\rho(\mathcal{X}, \ell, f) \triangleq \inf \Big\{ d \in \mathbb{R} : \exists c \in \mathbb{R},\ \forall \varepsilon > 0,\ H(\rho\varepsilon, \mathcal{X}_\varepsilon, \ell) \leqslant c - d \log \varepsilon \Big\}. \quad (2.10)$$

This alternative dimension is well suited for measuring the complexity of the search space with respect to the global optimization objective. Similar definitions, the *zooming dimensions*, are proposed in Kleinberg et al. (2008) and Bull (2015). It is at the heart of many *branch-and-bound* algorithms like Zooming (Kleinberg et al., 2008), HOO (Bubeck et al., 2008), DOO and SOO (Munos, 2011), TaxonomyZoom (Slivkins, 2011), StoSOO (Valko et al., 2013), ATB (Bull, 2015) or POO (Grill et al., 2015). We will see that for the deterministic case, the convergence rate of optimization algorithm looks like $n^{-1/\dim_\rho(\mathcal{X},\ell)}$ and e^{-n} when $\dim_\rho(\mathcal{X}, \ell) = 0$. It is interesting to note that the near-optimality dimension can be much smaller than the covering dimension.

Illustrative Examples of Zero Near-Optimality Dimensions

Example 1. As a simple example, take $\mathcal{X} = [-1, 1]^d$ and $f : x \mapsto 1 - \|x\|_p^\alpha$ for $\alpha \geqslant 1$ and $p \in \mathbb{R} \cup \{\infty\}$. We can choose $\ell(x_1, x_2) = \|x_1 - x_2\|_p^\alpha$ so that $\mathcal{X}_\varepsilon = \mathcal{B}(0, \varepsilon)$, the centered ℓ-ball of radius ε. Therefore for all $\rho > 0$ it exists a constant c_ρ such that $N(\rho\varepsilon, \mathcal{X}_\varepsilon, \ell) = c_\rho$. We conclude $\dim_\rho(\mathcal{X}, \ell) = 0$.

Example 2. To construct a generic example, we take an arbitrary $\mathcal{X}$ with norm $\|\cdot\|$ and f with maximum at $x^\star$, and we set ℓ according to the local modulus of continuity of f at the optimum,

$$\ell(x_1, x_2) = \omega_f^\star(\|x_1 - x_2\|) \text{ where } \omega_f^\star(\delta) = f(x^\star) - \inf_{x \in \mathcal{B}(x^\star, \delta)} f(x).$$

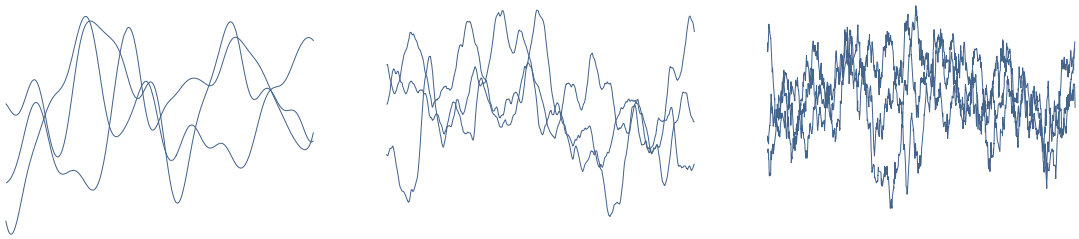
Then Eq. 2.6 directly holds. In order to obtain $\dim_\rho(\mathcal{X}, \ell) = 0$, it suffices that it exists a constant $0 < c \leqslant 1$ such that for all $x \in \mathcal{X}$ we have $f(x^\star) - f(x) \geqslant c\ell(x^\star, x)$, that is upper- and lower-envelopes of f around $x^\star$ are of the same order. Then $\mathcal{X}_\varepsilon \subseteq \mathcal{B}(x^\star, c^{-1}\varepsilon)$ and $\dim_\rho(\mathcal{X}, \ell) = 0$ as soon as $(\mathcal{X}, \ell)$ has finite doubling constant.

2.1.4 Bayesian Assumption, Gaussian Processes and Continuity

Instead of analyzing optimization for f in a given set of smooth functions, the Bayesian optimization framework consists in assuming a probability distribution $\mathcal{G}$ over functions $\mathcal{X} \to \mathbb{R}$, and analyze what happens with high probability when the objective function is a realization of this probability distribution. Here for an algorithm $\mathcal{A}$, we are looking for results under the form:

$$\forall u > 0,\ \mathbb{P}\Big[\forall n \geqslant 1,\ R_n \leqslant U_3(\mathcal{G}, n, u)\Big] \geqslant 1 - e^{-u},$$

where U_3 should be as small as possible, or in expectation, $\forall n \geqslant 1,\ \mathbb{E}\big[R_n\big] \leqslant U_4(\mathcal{G}, n)$, where the probability measure also includes both the noise and the randomness of f.



Squared Exponential | Matérn $\nu = 3/2$ | Matérn $\nu = 1/2$

Figure 2.3. – Realizations of Gaussian processes with different kernels

Mathematical Foundations of Stochastic Processes

Formally, f is modeled as a stochastic process indexed by $\mathcal{X}$ having values in $\mathbb{R}$, that is for every $x \in \mathcal{X}$, $f(x)$ is a real random variable. We recall the measure-theoretic definition of a stochastic process and fix our notations below.

Definition 2.3 (Stochastic Process)**.** *Let $(\Omega, \Sigma, \mathbb{P})$ be a probability space. A stochastic process indexed by $\mathcal{X}$ is a function $f : \mathcal{X} \times \Omega \to \mathbb{R}$ such that for all $x \in \mathcal{X}$, $f(x) \equiv f(x, \cdot)$ is a random variable on Ω. To simplify the sequel we also use the following notation $f \equiv \{f(x, \cdot)\}_{x \in \mathcal{X}}$.*

We model the independence of the noise sequence $\epsilon = (\epsilon_n)_{n \geqslant 0}$ by considering the product of the probability spaces of both f and ϵ. The expectations $\mathbb{E}[\cdot]$ and probabilities $\mathbb{P}[\cdot]$ are taken on this product space.

In this setting the distribution of $\sup_{x^\star \in \mathcal{X}} f(x^\star)$ is not trivial. To avoid unnecessary measurability issues, we always assume that f is separable, that is it exists $\mathcal{X}'$ a countable dense subset of $\mathcal{X}$ such that the following holds with probability one:

$$\forall x \in \mathcal{X}\,, \exists x_1, x_2, \ldots \in \mathcal{X}' \text{ s.t. } x_i \to x \text{ and } f(x_i) \to f(x).$$

Note that is suffices that one of the following properties holds (Giné and Nickl, 2015):

1. $\mathcal{X}$ is countable;
2. f is continuous with probability one;
3. $\mathcal{X}$ is well-ordered and f is right-continuous with probability one.

With this assumption the supremum of f is well defined and we have:

$$\sup_{x^\star \in \mathcal{X}} f(x^\star) = \sup_{\substack{\mathcal{X}' \subseteq \mathcal{X} \\ |\mathcal{X}'| < \infty}} \sup_{x^\star \in \mathcal{X}'} f(x^\star).$$

Introduction to Gaussian Processes

The most common (and almost the only) stochastic processes used in Bayesian optimization are Gaussian processes. Intuitively, Gaussian processes are extension of multivariate normal variable to continuous domains. Instead of having a mean vector they have a mean function, and instead of a covariance matrix they have a covariance function. We state here the proper definition of Gaussian processes.

Definition 2.4 (GAUSSIAN PROCESS). *A stochastic process f on $\mathcal{X}$ is a Gaussian process if for all integer $n \geqslant 1$ and all $x_1, \ldots, x_n \in \mathcal{X}$ the finite dimensional marginal $\big(f(x_1), \ldots, f(x_n)\big)$ is multivariate normal. The function $m : x \in \mathcal{X} \mapsto \mathbb{E}\big[f(x)\big]$ is called the mean of the process and the function $k : x_1, x_2 \mapsto \mathbb{E}\big[\big(f(x_1) - m(x_1)\big)\big(f(x_2) - m(x_2)\big)\big]$ is called the covariance or kernel of the process. We write:*

$$f \sim \mathcal{GP}(m, k).$$

Thanks to the Kolmogorov consistency theorem, given a function $m : \mathcal{X} \to \mathbb{R}$ and a symmetric and non-negative definite function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ there exists a Gaussian process $f \sim \mathcal{GP}(m, k)$. We say that a kernel is *stationary* when it is a function of a distance of its arguments:

$$\forall x_1, x_2 \in \mathcal{X},\ k(x_1, x_2) = \dot{k}\big(\,\|x_1 - x_2\|\,\big),$$

for an appropriate norm $\|\cdot\|$. The kernel k of a Gaussian process controls the smoothness of the realizations. For stationary kernels, the faster it decays toward zero, the rougher the realizations are. Intuitively, the values at nearby locations are highly correlated, but the values at distant locations are independent. In the sequel we will often focus on the following three kernel functions, which are typically used in the Bayesian optimization literature. For x_1 and x_2 in $\mathbb{R}^d$, they are defined as follows:

$$\text{Linear:} \qquad k(x_1, x_2) = x_1^\top x_2, \tag{2.11}$$

$$\text{Squared Exponential:} \qquad k(x_1, x_2) = e^{-\frac{1}{2}\|x_1 - x_2\|_2^2}, \tag{2.12}$$

$$\text{Matérn with } \nu > 0: \qquad k(x_1, x_2) = \int_{\mathcal{X}} \frac{e^{ix^\top(x_1 - x_2)}}{\big(1 + \|x\|_2^2\big)^{\nu + d/2}}\,\mathrm{d}x \tag{2.13}$$

$$= \frac{2^{1-\nu}}{\Gamma(\nu)}\big(\sqrt{2\nu}\,\|x_1 - x_2\|_2\big)^\nu K_\nu\big(\sqrt{2\nu}\,\|x_1 - x_2\|_2\big),$$

where K_ν is the modified Bessel function of the second kind which for a real number z is equal to $\frac{1}{2}(\frac{1}{2}z)^\nu \int_0^\infty \exp(-t - z^2/(4t))t^{-\nu-1}\,\mathrm{d}t$. The Matérn kernel with parameter ν is rarely implemented with this level of generality. When 2ν is an odd integer, it simply writes as a product of an exponential and a polynomial:

$$\begin{aligned} k(x_1, x_2) &= h_{2\nu}\big(\,\|x_1 - x_2\|_2\big)e^{-\sqrt{2\nu}\|x_1 - x_2\|_2}, \\ \text{with } h_1(r) &= 1, \\ h_3(r) &= 1 + \sqrt{3}r, \\ h_5(r) &= 1 + \sqrt{5}r + \tfrac{5}{3}r^2. \end{aligned}$$

We remark that a Gaussian process with Matérn kernel of parameter $\nu = 1/2$ is also known as a Ornstein-Uhlenbeck process. And we finally note that the Matérn kernel converges to the squared exponential kernel when $\nu \to \infty$. Samples from a Gaussian process with squared exponential kernel are infinitely differentiable, and with a Matérn kernel of parameter ν are differentiable k times for the larger integer $k < \nu$. Realizations of Gaussian processes

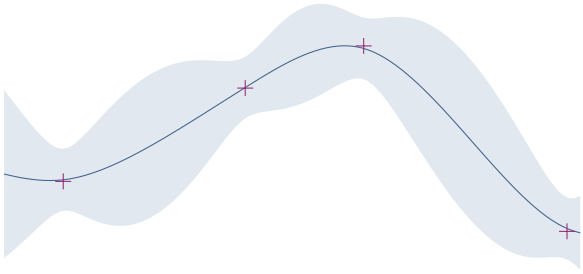


Figure 2.4. – Posterior distribution of a Gaussian process in one dimension, with $n = 4$ noisy observations (red crosses). The blue line is the posterior expectation μ_n, and the Gray area is delimited by $\mu_n(\cdot) \pm \sqrt{\beta_n}\sigma_n(\cdot)$, where σ_n^2 is the posterior variance and β_n a confidence parameter.

with various kernels are illustrated in Figure 2.3. Another notable Gaussian process is the Brownian motion on $[0, 1]$, equal to the centered Gaussian process with kernel:

$$k(x_1, x_2) = x_1 \wedge x_2.$$

Posterior Distribution and Bayesian Inference

Since the effect of the mean function m is only additive, we assume without loss of generality that the unknown function to be optimized is a realization of a centered Gaussian process $f \sim \mathcal{GP}(0, k)$. We assume further that the noise variables ϵ_i are distributed as independent centered Gaussian $\mathcal{N}(0, \eta^2)$ like previously. The theoretical analysis presented in this thesis requires the covariance function k and the variance of the noise η^2 to be known in advance. In practice this is not often the case, we will consider in Section 2.1.5 methods to learn k and η from the observations. Having in hand an history of noisy measurements $x_1, y_1, \ldots, x_n, y_n$, Bayesian inference consists in computing the posterior distribution of f given the observations. The fame of Gaussian processes in Bayesian optimization is partially explained by the handy property that the posterior distribution is another Gaussian process,

$$f \,|\, \mathcal{F}_n \sim \mathcal{GP}(\mu_n, k_n),$$

where $\mathcal{F}_n$ is the associated σ-field from Eq. 2.1, and the posterior mean μ_n and posterior covariance k_n enjoy the closed formulae below. We denote by $X_n \triangleq \{x_i\}_{i \leqslant n}$ the set of queries and $\mathbf{Y}_n \triangleq [y_i]_{i \leqslant n}$ the column vector of noisy observations. and $\mathbf{K}_n \triangleq [k(x_i, x_j)]_{i,j \leqslant n}$ the kernel matrix of the data. Then for all $x \in \mathcal{X}$, let $\mathbf{k}_n(x) \triangleq [k(x,x_i)]_{i \leqslant n}$ be the column vector of the kernel evaluation between x and the queries X_n. We have:

$$\mu_n(x) = \mathbf{k}_n(x)^\top \mathbf{C}_n^{-1} \mathbf{Y}_n, \tag{2.14}$$

$$k_n(x, x') = k(x, x') - \mathbf{k}_n(x)^\top \mathbf{C}_n^{-1} \mathbf{k}_n(x'), \tag{2.15}$$

where $\mathbf{C}_n$ denotes $\mathbf{K}_n + \eta^2\mathbf{I}$ and $\mathbf{I}$ the identity matrix. We will denote by $\sigma_n^2(x)$ the posterior variance at any point x:

$$\sigma_n^2(x) \triangleq k_n(x, x). \tag{2.16}$$

Intuitively from a regression point of view, $\mu_n(\cdot)$ forms a prediction for the unknown values of f at any locations, and $\sigma_n^2(\cdot)$ can be seen as the uncertainty of this prediction, as shown in Figure 2.4. The posterior mean μ_n interpolates and extrapolates the noisy observations in a similar way to kernel regression with Tikhonov regularization, since it is the solution of:

$$\operatorname*{argmin}_{\mu:\mathcal{X}\to\mathbb{R}} \left\{ \sum_{i=1}^{n} \big(y_i - \mu(x_i)\big)^2 + \eta^2 \|\mu\|_{\mathcal{H}_k}^2 \right\},$$

where $\|\cdot\|_{\mathcal{H}_k}$ is the RKHS norm associated to the kernel k, which we will present later in this section.

Stochastic Smoothness of Gaussian Processes

Since for a Gaussian process the marginals $\big(f(x_1), f(x_2)\big)$ are bivariate normal for all $x_1, x_2 \in \mathcal{X}$, we know that $f(x_1) - f(x_2)$ is distributed as a centered Gaussian. Let $\ell(x_1, x_2)$ be its standard deviation, that is the L^2-distance between the random variables $f(x_1)$ and $f(x_2)$:

$$\begin{aligned}\ell^2(x_1, x_2) &\triangleq \mathbb{E}\Big[\big(f(x_1) - f(x_2)\big)^2\Big] \\ &= k(x_1, x_1) + k(x_2, x_2) - 2k(x_1, x_2).\end{aligned}$$

The function $\ell(\cdot, \cdot)$ defines a pseudo-metric on $\mathcal{X}$ and we call it the canonical pseudo-distance of the process f. The geometrical properties of $(\mathcal{X}, \ell)$ play a crucial role for Bayesian optimization. We review useful properties of this particular pseudo-metric space similar to the one from the previous section. Unfortunately, the fact that f is stochastic renders the analysis difficult. We first remark that for all $x_1, x_2 \in \mathcal{X}$ since we have:

$$f(x_1) - f(x_2) \sim \mathcal{N}\big(0, \ell^2(x_1, x_2)\big),$$

we know using classical Gaussian concentration that for all $u > 0$,

$$\mathbb{P}\Big[f(x_1) - f(x_2) \leqslant \sqrt{2u}\ell(x_1, x_2)\Big] \geqslant 1 - e^{-u}.$$

Therefore if one has $|\mathcal{X}| \leqslant m$, a union bound leads to,

$$\mathbb{P}\Big[\forall x_1, x_2 \in \mathcal{X},\ f(x_1) - f(x_2) \leqslant \sqrt{2u}\ell(x_1, x_2)\Big] \geqslant 1 - m^2 e^{-u}.$$

However, we cannot simply obtain an inequality similar to Eq. 2.4 holding everywhere with high probability when $|\mathcal{X}|$ is not bounded. In fact, even the continuity of the realizations on arbitrary $\mathcal{X}$ is not trivial. Borell (1975) and Cirel'son et al. (1976) proved that f is

almost surely continuous everywhere if and only if the expected global modulus of continuity converges:

$$\lim_{\delta \to 0} \mathbb{E}\big[\omega_f(\delta)\big] = 0 \text{ where } \omega_f(\delta) \equiv \omega_f(\delta, \mathcal{X}, \ell) \triangleq \sup_{\substack{x_1, x_2 \in \mathcal{X} \\ \ell(x_1, x_2) \leqslant \delta}} \big(f(x_1) - f(x_2)\big).$$

In Chapter 3, we describe techniques using the metric entropy $H(\varepsilon, \mathcal{X}, \ell)$ from Eq. 2.8 to handle such stochastic quantities involving supremum. It can be shown (Adler and Taylor, 2009; Giné and Nickl, 2015) that it exists a constant $c \in \mathbb{R}$ such that for all Gaussian processes:

$$\mathbb{E}\big[\omega_f(\delta)\big] \leqslant c \int_0^\delta \sqrt{H(\varepsilon)}\, \mathrm{d}\varepsilon,$$

and that f is almost surely continuous when $\int_0^{\Delta(\mathcal{X})} \sqrt{H(\varepsilon)}\, \mathrm{d}\varepsilon < \infty$. We can already comment at this point that if the covering dimension $\dim(\mathcal{X}, \ell)$ from Definition 2.1 is finite, then the previous integral is finite and the stochastic process is almost surely continuous. For instance, we may take the Ornstein-Uhlenbeck process with stationary kernel $\dot{k}(r) = e^{-\sqrt{2}r}$. For this process, $\ell(x_1, x_2) \leqslant 2 \|x_1 - x_2\|_2^{1/2}$, hence on a compact $\mathcal{X} \subset \mathbb{R}^d$, there exists a constant $c_{\mathcal{X}} \in \mathbb{R}$ such that $H(\varepsilon) \leqslant c_{\mathcal{X}} + 2d \log \varepsilon^{-1}$. That is the covering dimension attains $2d$ and the process is almost surely continuous. Yes, it is well known that this process is not differentiable.

Reproducing Kernel Hilbert Spaces

The realizations of Gaussian processes are closely linked to functional spaces called reproducing kernel Hilbert spaces (RKHS). These spaces play an important role in statistical learning theory, notably for Support Vector Machines (Cristianini and Shawe-Taylor, 2000; Schölkopf and Smola, 2002). We show here how they are linked to the Gaussian process model. Let k by the kernel of a Gaussian process f, we define $\mathcal{H}_k$ the RKHS of k as follows:

Definition 2.5 (Reproducing Kernel Hilbert Space)**.** *Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a symmetric and non-negative definite function. Define the functional space $\mathcal{C}$ of the linear span of $\{k(x, \cdot)\}_{x \in \mathcal{X}}$ that is,*

$$\mathcal{C} \triangleq \left\{ h : \mathcal{X} \to \mathbb{R} : \exists m \geqslant 1, s_1, \ldots, s_m \in \mathcal{X}, a_1, \ldots, a_m \in \mathbb{R},\ h(\cdot) = \sum_{i=1}^m a_i k(x_i, \cdot) \right\},$$

endowed with the inner product:

$$\left\langle \sum_{i=1}^l a_i k(x_i, \cdot), \sum_{i=1}^m b_i k(x'_i, \cdot) \right\rangle_{\mathcal{H}_k} \triangleq \sum_{\substack{i \leqslant l \\ j \leqslant m}} a_i b_j k(x_i, x'_j).$$

The reproducing kernel Hilbert space $\mathcal{H}_k$ of k is defined by the completion of $\mathcal{C}$ by this inner product.

Functions in the RKHS enjoy the following reproducing kernel property:

$$\forall h = \sum_{i=1}^m a_i k(x_i, \cdot) \in \mathcal{C}, x \in \mathcal{X},\ \big\langle h, k(x, \cdot) \big\rangle_{\mathcal{H}_k} = \sum_{i=1}^m a_i k(x_i, x) = h(x),$$

and it is the only Hilbert space satisfying this property. By denoting $\|h\|_{\mathcal{H}_k}^2 = \langle h, h \rangle_{\mathcal{H}_k}$ and $\mathcal{G}$ the linear span of the random variables $\{f(x)\}_{x \in \mathcal{X}}$, we have that the function $\phi : \mathcal{G} \to \mathcal{C}$ defined as $\phi(g)(x) \triangleq \mathbb{E}[g \cdot f(x)]$ for $x \in \mathcal{X}$, is a linear isometry between $(\mathcal{G}, \|\cdot\|_{L^2})$ and $(\mathcal{C}, \|\cdot\|_{\mathcal{H}_k})$. That is, we can alternatively define the RKHS $\mathcal{H}_k$ of a Gaussian process f as the Hilbert space of functions, with $\bar{\mathcal{G}}$ is the closure of $\mathcal{G}$ in $L^2(\mathcal{X})$,

$$\left\{x \mapsto \mathbb{E}[g \cdot f(x)] : g \in \bar{\mathcal{G}}\right\},$$

with inner product:

$$\langle \mathbb{E}[g_1 f], \mathbb{E}[g_2 f] \rangle_{\mathcal{H}_k} \triangleq \mathbb{E}[g_1 g_2].$$

We can immediately deduce that the expectation map $x \mapsto \mathbb{E}[f(x)]$ lies in $\mathcal{H}_k$. However, the realizations of f are typically not in $\mathcal{H}_k$. As a first example, let us consider the RKHS of the Brownian motion process on $\mathcal{X} = [0, 1]$. We have seen before that we obtain $k(x_1, x_2) = x_1 \wedge x_2$. It can be shown (van der Vaart and van Zanten, 2008; Giné and Nickl, 2015) that:

$$\begin{aligned} \mathcal{H}_k &= \left\{g : g(0) = 0, g \text{ is absolutely continuous}, g' \in L^2(\mathcal{X})\right\}, \\ \langle g_1, g_2 \rangle_{\mathcal{H}_k} &= \int_0^1 g_1'(x) g_2'(x) \, \mathrm{d}x. \end{aligned}$$

This space is well known and usually called the Cameron-Martin space, and $\mathbb{P}[f \in \mathcal{H}_k] = 0$. We conclude this section on Gaussian processes by stating a useful representation property. Let $\lambda_1 \geqslant \lambda_2 \geqslant \ldots$ and $\psi_1, \psi_2, \ldots$ be the eigenvalues and eigenfunctions of the Hilbert-Schmidt integral operator $T_k : L^2(\mathcal{X}) \to L^2(\mathcal{X})$ defined by $T_k(g)(\cdot) \triangleq \int_{\mathcal{X}} k(x, \cdot) g(x) \, \mathrm{d}x$, that is we have:

$$\begin{aligned} \int_{\mathcal{X}} k(x, \cdot) \psi_i(x) \, \mathrm{d}x &= \lambda_i \psi(\cdot), \\ \int_{\mathcal{X}} \psi_i(x) \psi_j(x) \, \mathrm{d}x &= \mathbb{1}\{i = j\}. \end{aligned}$$

Mercer's theorem gives that for all $x_1, x_2 \in \mathcal{X}$,

$$k(x_1, x_2) = \sum_{i \geqslant 1} \lambda_i \psi_i(x_1) \psi_j(x_2),$$

and that these series converge absolutely and uniformly. Furthermore we have that $\{\sqrt{\lambda_i} \psi_i\}_{i \geqslant 1}$ is a complete orthonormal system of $\mathcal{H}_k$. Finally, the Gaussian process f can be represented as the following Karhunen-Loève expansion:

$$f(\cdot) = \sum_{i \geqslant 1} \sqrt{\lambda_i} \psi_i(\cdot) X_i, \tag{2.17}$$

$$X_i \overset{\text{iid}}{\sim} \mathcal{N}(0, 1). \tag{2.18}$$

This permits to obtain the following results:

$$\mathbb{P}\big[f \in \mathcal{H}_k\big] = \begin{cases} 1 & \text{if } \mathcal{H}_k \text{ is finite dimensional,} \\ 0 & \text{otherwise,} \end{cases}$$

and that $\mathbb{P}\big[\, \|f - h\|_\infty < \epsilon\big] > 0$ for any $h \in \mathcal{H}_k$ and $\epsilon > 0$ and $\mathbb{P}\big[f \in \overline{\mathcal{H}_k}\big] = 1$ where $\overline{\mathcal{H}_k}$ is the closure of $\mathcal{H}_k$ in the Banach subspace of $\mathbb{R}^{\mathcal{X}}$ endowed with supremum norm. Functions in the RKHS are smoother than samples from the corresponding Gaussian process, and the distribution of the process put positive probability on any tight neighborhood of functions from $\mathcal{H}_k$.

2.1.5 Practical and Theoretical Properties of Priors and Posteriors of Gaussian Processes

We conclude Section 2.1 by outlining central properties of the Gaussian process model, and links with the frequentist model.

Design of the Prior

When we think of the unknown function f as a realization of a Gaussian process, the kernel function should reflect all the available prior knowledge, typically its smoothness. In the previous section, we reviewed three popular kernels. The choice of a linear kernel (Eq. 2.11) implies the assumption that f is linear, and its parameters have multivariate standard normal prior. The squared exponential kernel (Eq. 2.12) is a prior on infinitely differentiable functions $\mathbb{R}^d \to \mathbb{R}$. The Matérn kernel with parameter ν (Eq. 2.13) models functions $\mathbb{R}^d \to \mathbb{R}$ that are differentiable k times for the largest integer $k < \nu$. Different kernels may be combined to form a prior with multiple properties, typically by addition of two kernels. For example, consider the addition of a quadratic kernel and a squared exponential kernel. This prior models roughly-quadratic functions with additional smooth variations. As stated in Section 2.1.3, if the unknown function satisfies some invariances such as symmetry or periodicity, an adequate kernel avoids to waste queries during the optimization procedure. When the input space in not a subset of $\mathbb{R}^d$ but another structured space, some care must be taken to design a positive semi-definite kernel. A practitioner may pick and combine multiple kernels from the large literature on this subject, we refer to Gärtner et al. (2003); Borgwardt and Kriegel (2005); Vishwanathan et al. (2010) for graphs kernels, Lodhi et al. (2002); Leslie et al. (2002) for string kernels, or Neuhaus and Bunke (2006) for edit-distance based kernels.

Empirical Confidence Intervals

A key property from the Gaussian process framework is that after observing n noisy evaluations, the posterior distribution at any location $x \in \mathcal{X}$ has a normal distribution of expectation $\mu_n(x)$ and variance $\sigma_n^2(x)$, with μ_n and σ_n^2 defined in Eq. 2.14, 2.16. We can then define an upper confidence bound U_n and a lower confidence bound L_n, such that f is included in the interval with high probability,

$$U_n(x) \triangleq \mu_n(x) + \sqrt{\beta_n \sigma_n^2(x)}, \tag{2.19}$$

$$\text{and } L_n(x) \triangleq \mu_n(x) - \sqrt{\beta_n \sigma_n^2(x)}. \tag{2.20}$$

Let $u > 0$, $a > 1$ and $\mathcal{X}$ be finite. We fix accordingly,

$$\beta_n \triangleq 2u + 2\log\big(|\mathcal{X}|n^a\zeta(a)\big), \tag{2.21}$$

where ζ is the Riemann zeta function. For instance $\beta_n = 2u + 2\log\big(|\mathcal{X}|n^2\pi^2/6\big)$. We then have the following guarantee by union bounds on $n \in \mathbb{N}$ and $x \in \mathcal{X}$:

$$\mathbb{P}\Big[\forall n \geqslant 1, \forall x \in \mathcal{X},\ f(x) \in \big(L_n(x), U_n(x)\big)\Big] \geqslant 1 - e^{-u}. \tag{2.22}$$

The upper and lower confidence bounds are illustrated on Figure 2.4 respectively by the upper and lower envelopes of the gray area. The region delimited in that way, the high confidence intervals, contains the unknown f with high probability. This statement will be a main element in the subsequent analysis in Section 3.1, and we will see in Section 3.2 tools to adapt to continuous search spaces.

Likelihood Maximization and Frequentist Properties of Bayesian Inference

In a real scenario, the unknown function may take values at an unknown scale, and similarly the input dimensions may have unequal importance. To rectify these uncertainties, the kernel is typically calibrated by scale and bandwidth parameters. Since the natural approach in the Bayesian model is to consider the maximizer of the posterior likelihood, the parameters of the kernel are then selected by maximizing the posterior likelihood. Other approaches are often used to prevent over-fitting problems, like cross validation or maximization of the pseudo-likelihood (Rasmussen and Williams, 2006). Unfortunately, there is only few known frequentist statistical guarantees under this perspective, that is when f is not assumed to be a realization of a known prior distribution. In van der Vaart and van Zanten (2011) the authors show that when f is a fixed function in the support of the prior, then the posterior distribution is consistent, in the sense that the L_2-risk converges to zero as the number of observations increases. They give guarantees on the convergence rates, and demonstrate that when the smoothness of the kernel fits the smoothness of f the obtained rates are polynomial with optimal exponent, but when the smoothness is overestimated the rates can be logarithmic, as for a squared exponential kernel on a function only differentiable a finite number of times. They use two possible notions of smoothness, the Hölder spaces $C^{\nu,\alpha}(\mathbb{R}^d)$, that is functions with ν continuous derivatives and whose ν-th derivative is α-Hölder-continuous; and Sobolev space $H^\nu(\mathbb{R}^d)$ with finite Sobolev norm:

$$\|f\|_\nu^2 \triangleq \int \big(1 + \|\lambda\|^2\big)^\nu |\hat{f}(\lambda)|^2 \,\mathrm{d}\lambda,$$

where $\hat{f}$ is the Fourier transform $\hat{f}(\lambda) \triangleq (2\pi)^{-d}\int e^{i\lambda^\top x} f(x)\,\mathrm{d}x$. In van der Vaart and van Zanten (2009) the same authors propose a prior built on the squared exponential kernel with inverse Gamma "hyper"-prior on the bandwidth parameters. They prove that when f belongs to $C^{\nu,\alpha}(\mathbb{R}^d)$ then the posterior obtained by full Bayesian inference obtained optimal convergence rate without the knowledge of ν and α. However these results are only proved for classification or regression and not for an active setting. Therefore in the sequel, we consider that the prior used by our algorithms is always the true prior that generates the unknown function in a passive batch setting.

2.2 Optimization Algorithms and Theoretical Results

The second section of this chapter presents fundamental results for sequential optimization in the various assumptions introduced above. This section does not aim at forming a complete review of the literature but rather establishing comparisons of the state-of-the-art under the same framework. We do not cover adversarial settings, and we refer to Bubeck and Cesa-Bianchi (2012) for a complete and concise review of this problem.

2.2.1 Stochastic Multi-Armed Bandits

In the stochastic multi-armed bandits framework, the search space $\mathcal{X}$ is finite and f possesses no particular structure. Let K be the cardinality of $\mathcal{X}$, we denote the arms by integers without loss of generality:

$$\mathcal{X} = (1, 2, \dots, K).$$

Multi-armed bandits are usually defined with an arbitrary distribution $\nu_a \in \mathcal{D}$ for each arm $a \in \mathcal{X}$, where the set of allowed distributions $\mathcal{D}$ is known but ν_a is unknown (Bubeck and Cesa-Bianchi, 2012). The observations y_i for arm $x_i \in \mathcal{X}$ are then independent realizations of ν_{x_i}. In our setting, $f(x_i) = \mathbb{E}\big[y_i\big]$ and $\epsilon_i = f(x_i) - y_i$. Our definition of the cumulative regret R_n from Eq. 2.3 corresponds to the pseudo-regret. We mainly focus on the case where the noise ϵ_i are independent Gaussian random variables $\mathcal{N}(0, \eta^2)$ to be consistent with further settings, that is $\mathcal{D}$ is the set of Gaussian distribution with arbitrary mean and fixed variance. A common extension it to consider various distribution such as one-parameter exponential families or nonparametric distributions (Cappé et al., 2013; Perchet and Rigollet, 2013).

For an algorithm $\mathcal{A}$ playing arms $x_1, \dots, x_n$, the cumulative regret can be decomposed as follows:

$$R_n = \sum_{i=1}^{n} \sup_{x^\star} f(x^\star) - f(x_i) = \sum_{a=1}^{K} \Delta_a N_n(a),$$

where Δ_a is the gap between the value of arm a and the optimum:

$$\Delta_a \equiv \Delta_a(f, \mathcal{X}) \triangleq \sup_{x^\star \in \mathcal{X}} f(x^\star) - f(a),$$

and $N_n(a)$ is the (random) number of times arm a has been selected until iteration n:

$$N_n(a) \equiv N_n(a, \mathcal{A}, f, \epsilon) \triangleq \sum_{i=1}^{n} \mathbb{1}\{x_i = a\}.$$

This particular decomposition leads to upper and lower bounds on the cumulative regret as presented below.

Lower Bounds on the Cumulative Regret

In Lai and Robbins (1985) the authors prove a significant lower bound on the expected number of times a sub-optimal arm should be queried. Thanks to the previous decomposition

Algorithm 1: UCB with log-Laplace ψ and exploration parameter $\beta > 2$

$\forall a \leqslant K,\ N(a) \leftarrow 0$
for $n = 0, 1, \ldots$ **do**
  for $a = 1, \ldots, K$ **do**
    $\widehat{\mu}_n(a) \leftarrow N(a)^{-1} \sum_{i=1}^{n} y_i \mathbb{1}\{x_i = a\}$
    $U(a) \leftarrow \widehat{\mu}_n(a) + \psi^{*-1}\big(N(a)^{-1} \beta \log n\big)$
  end
  $x_{n+1} \leftarrow \operatorname{argmax}_{a \leqslant K} U(a)$
  $y_{n+1} \leftarrow$ **Query**(x_{n+1})
  $N(x_{n+1}) \leftarrow N(x_{n+1}) + 1$
end

of R_n, this allows to derive a lower bound for the expected cumulative regret. They show that for any algorithm with sub-linear cumulative regret, for all sub-optimal arm a one has:

$$\liminf_{n \to \infty} \frac{\mathbb{E}\big[N_n(a)\big]}{\log n} \geqslant D_{\mathrm{KL}}(\nu_a \parallel \nu_\star)^{-1}, \tag{2.23}$$

where $D_{\mathrm{KL}}(\nu_a \parallel \nu_\star)$ is the KL-divergence between the distribution of the noisy rewards for arm a and an optimal arm. For Gaussian noise with fixed variance η^2, one has:

$$D_{\mathrm{KL}}(\nu_a \parallel \nu_\star) = \frac{\Delta_a^2}{2\eta^2}. \tag{2.24}$$

That is the expected cumulative regret of any algorithm satisfies:

$$\liminf_{n \to \infty} \frac{\mathbb{E}\big[R_n\big]}{\log n} \geqslant 2\eta^2 \sum_{a : \Delta_a > 0} \Delta_a^{-1}. \tag{2.25}$$

This asymptotic lower bound becomes large when there is one Δ_a close to zero for an almost optimal arm a. Nevertheless, this lower bound is tight in the sense that there exists an algorithm, presented in the next paragraph, with matching upper bound.

Asymptotically Optimal Algorithms for the Cumulative Regret

The canonical strategy to face the noise is to sample multiple times the same arms and consider the empirical average to reduce the uncertainty. Since the noise variables are independent, the Cramer-Chernoff bounding method (Boucheron et al., 2013) is a useful tool to control the empirical confidence intervals. First, let ψ denote the logarithm of the Laplace transform of the noise variables:

$$\psi(\lambda) \triangleq \log \mathbb{E}\big[e^{\lambda \epsilon_1}\big].$$

Then we define ψ^* the convex conjugate of ψ (also known as Legendre-Fenchel transform):

$$\psi^*(s) \triangleq \sup_{\lambda \in \mathbb{R}} \big(\lambda s - \psi(\lambda)\big).$$

Algorithm 2: KL-UCB with set of distributions $\mathcal{D}$ and exploration parameter $\beta > 2$

$\forall a \leqslant K,\ N(a) \leftarrow 0$
for $n = 0, 1, \ldots$ **do**
  for $a = 1, \ldots, K$ **do**
    $\widehat{\nu}_a \leftarrow$ **Empirical**$\big(\mathcal{D}, \{y_i : i < n,\ x_i = a\}\big)$
    $U(a) \leftarrow \sup\Big\{\mathbb{E}[\nu] : \nu \in \mathcal{D},\ D_{\mathrm{KL}}(\widehat{\nu}_a \,\|\, \nu) \leqslant N(a)^{-1}(\log n + \beta \log\log n)\Big\}$
  end
  $x_{n+1} \leftarrow \mathrm{argmax}_{a \leqslant K}\, U(a)$
  $y_{n+1} \leftarrow$ **Query**(x_{n+1})
  $N(x_{n+1}) \leftarrow N(x_{n+1}) + 1$
end

Markov's inequality tells us that for all $s > 0$ we have:

$$\mathbb{P}\big[\epsilon_1 > s\big] \leqslant e^{-\psi^*(s)}.$$

So, we can denote ψ^{*-1} the generalized inverse of ψ^*,

$$\psi^{*-1}(u) \triangleq \inf\big\{s \in \mathbb{R} : \psi^*(s) > u\big\},$$

which leads to the following concentration inequality:

$$\mathbb{P}\Big[\epsilon_1 > \psi^{*-1}(u)\Big] < e^{-u}.$$

Now if we query N times arm a, by independence of the noise variables we directly obtain:

$$\mathbb{P}\Big[f(a) - \widehat{\mu}_n(a) > \psi^{*-1}\big(N^{-1}u\big)\Big] < e^{-u},$$

where $\widehat{\mu}_n(a) \triangleq N_n(a)^{-1} \sum_{i=1}^n y_i \mathbb{1}\{x_i = a\}$ is the empirical average of arm a. This motivated the UCB algorithm, presented in Algorithm 1. The UCB algorithm queries arms maximizing $\widehat{\mu}_n(a) + \psi^{*-1}\big(N_n(a)^{-1}u\big)$ for appropriate $u > 0$, which is an upper confidence bound for $f(a)$, hence the name. For η-subgaussian noise distribution, we have:

$$\psi_\eta(\lambda) = \frac{\eta^2\lambda^2}{2},$$

so by simple calculus the convex dual and its inverse simplify to:

$$\psi_\eta^*(s) = \frac{s^2}{2\eta^2} \text{ and } \psi_\eta^{*-1}(u) = \sqrt{2\eta^2 u}.$$

It is well known that the expected cumulative regret of the UCB algorithm is asymptotically optimal up to constants, on bounded or Gaussian noise (Auer et al., 2002; Bubeck and Cesa-Bianchi, 2012). This algorithm paved the way for many other settings including Lipschitzian and Bayesian optimization, as it will be discussed in the next sections.

It is possible to build algorithms for which the expected cumulative regret follows exactly the asymptotic given in Eq. 2.25 for any noise distributions, with the right constant. The KL-UCB algorithm (Cappé et al., 2013) is designed to match the terms of Eq. 2.23. The upper confidence bound from the dual of the log-Laplace is replaced by finer KL-divergence. We detail this method in Algorithm 2 for generic noise distributions in $\mathcal{D}$. The procedure **Empirical**$(\mathcal{D}, Y)$ computes the projection on $\mathcal{D}$ of the empirical distribution of observations Y. For Gaussian noise with variance η^2, the intriguing value $U(a)$ in KL-UCB simplifies with Eq. 2.24 to:

$$U(a) = \widehat{\mu}_n(a) + \sqrt{2\frac{\eta^2 \beta_n}{N_n(a)}},$$

The algorithm is then similar to the previous UCB strategy up to logarithmic terms in the exploration parameter. We refer the reader to Audibert et al. (2009), Auer and Ortner (2010) and Perchet and Rigollet (2013) for refined analysis of the expected cumulative reward in multi-armed bandits.

High Probability Guarantees

For multi-armed bandits, high probability results of the form $\mathbb{P}[R_n \leqslant g(K, \mathcal{D}, n, u)] \geqslant 1 - e^{-u}$ are limited when the algorithm is agnostic to the time horizon n. Salomon and Audibert (2011) proved that in the general case, no algorithm can have both a logarithmic expected cumulative regret and concentration around the expectation better than $1 - (\log n)^{-\alpha}$ for any $\alpha > 0$. Yet, if the algorithm knows the value u for the log-probability, then one can modify the UCB algorithm with refined upper confidence bound to obtain constant cumulative regret. Such an algorithm for subgaussian noise is the improved UCB algorithm (Abbasi-Yadkori et al., 2011), that uses the confidence intervals:

$$U_n(a) \triangleq \widehat{\mu}_n(a) + \sqrt{\frac{N_n(a)+1}{N_n^2(a)}\Big(1 + 2(u + \log K) + \log\big(N_n(a)+1\big)\Big)^{\frac{1}{2}}},$$

and displays a surprising constant cumulative regret with high probability:

$$\mathbb{P}\left[\forall n \geqslant 1,\ R_n \leqslant c \sum_{i:\Delta_i > 0} \Big(\Delta_i + \Delta_i^{-1}\Big(u + \log\big(K\Delta_i^{-1}\big)\Big)\Big)\right] \geqslant 1 - e^{-u},$$

where $c \in \mathbb{R}$.

Lower Bounds on the Simple Regret

There are several approaches in the multi-armed bandit literature to study the problem of finding the optimum. The PAC model (Even-Dar et al., 2002) focuses on the number of queries needed to obtain a simple regret S_n (Eq. 2.2) of at most ε with probability $1 - \delta$. If an algorithm is (ε, δ)-PAC, then it stops at $n_{\varepsilon,\delta}$ such that $\mathbb{P}[S_{n_{\varepsilon,\delta}} > \varepsilon] < \delta$, and the algorithm knows in advance ε and δ. In Mannor and Tsitsiklis (2004), the authors exhibit a lower bound on $n_{\varepsilon,\delta}$ needed to have the (ε, δ)-PAC guarantee in the case of Bernoulli observations. They prove that it exists $c_1, c_2 \in \mathbb{R}$ such that for all $\varepsilon > 0$ small enough and $\delta > 0$ small enough,

$$\mathbb{E}\big[n_{\epsilon,\delta}\big] \geqslant c_1 \frac{K}{\varepsilon^2} \log \frac{c_2}{\delta}. \tag{2.26}$$

Algorithm 3: MedianElimination(ε, δ)

$S_0 \leftarrow \mathcal{X}$; $\varepsilon_0 \leftarrow \epsilon/4$; $\delta_0 \leftarrow \delta/2$
for $l = 0, 1, \dots$ **do**
  for $a \in S_l$ **do**
    for $i = 1, \dots, (\varepsilon_l/2)^{-2} \log(3/\delta_l)$ **do**
      $y_{l,a,i} \leftarrow$ **Query**(a)
      Compute $\widehat{\mu}_a$
    end
  end
  $m \leftarrow$ **Median**$\big(\{\widehat{\mu}_a : a \in S_l\}\big)$
  $S_{l+1} \leftarrow S_l \backslash \{a : \widehat{\mu}_a < m\}$
  $\varepsilon_{l+1} \leftarrow 3\varepsilon_l/4$; $\delta_{l+1} \leftarrow \delta_l/2$
  if $|S_{l+1}| = 1$ **then**
    return S_{l+1}
  end
end

Since this framework gives strictly more information to the algorithm, the lower bound is also valid for the simple regret and we deduce that:

$$\forall n \geqslant 1, \ \mathbb{P}\left[S_n \geqslant \sqrt{c_1 \frac{K}{n} \log \frac{c_2}{\delta}} \right] \geqslant 1 - \delta.$$

Another setting, the best arm identification problem, consists in evaluating the probability that an algorithm find the exact optimum. We remark that for Bernoulli observations, the simple regret is always smaller than this probability. In that case Audibert et al. (2010) prove that there exists $c_1 \in \mathbb{R}$ such that,

$$\mathbb{E}[S_n] \geqslant \exp\left(-c_1 \frac{n}{H \log K} \right), \tag{2.27}$$

where $H \triangleq \sum_{a:\Delta_a > 0} \Delta_a^{-2}$.

Optimal Algorithms for Pure Exploration Problems

The MedianElimination algorithm, shown in Algorithm 3, is (ε, δ)-PAC with near optimal number of queries (Even-Dar et al., 2002), that is it matches the lower bound from Eq. 2.26 up to constant multiplicative factors. This algorithm requires the knowledge of ε and δ, and cannot be considered in our optimization framework. Other closely related procedure like the Successive Reject algorithm (Audibert et al., 2010) have expected simple regret that match the lower bound from Eq. 2.27 up to logarithmic terms. We refer to Karnin et al. (2013), Jamieson et al. (2013), Kaufmann et al. (2016) and Garivier and Kaufmann (2016) for further upper and lower bounds on best arm identification problems. These algorithms are not suitable for optimization on continuous domain, since they heavily rely on the gaps between the optimum of the values of the other arms.

2.2.2 Stochastic Linear Bandits

As a response to the previous remark a continuous bandit model has been proposed where $\mathcal{X} \subset \mathbb{R}^d$ is compact and f is in $\mathcal{C}$ a set of linear functions (Auer et al., 2007). Interesting upper and lower bounds follow from Dani et al. (2008).

Algorithms and Logarithmic Upper Bounds on the Regret

Since f is linear, the candidate locations for the optimum are on $\mathcal{E}$ the extreme points of $\mathcal{X}$, that is the set of points that are not a convex combination of other points in $\mathcal{X}$. Then, we can extend the definition of the gap by looking at sub-optimal points in $\mathcal{E}$:

$$\Delta \triangleq \inf \Big\{ \sup_{x^\star \in \mathcal{X}} f(x^\star) - f(x) : x \in \mathcal{E},\ \sup_{x^\star \in \mathcal{X}} f(x^\star) > f(x) \Big\}.$$

When $\Delta > 0$, for example if $\mathcal{X}$ is a polytope, then the linear bandits problem is equivalent to classical multi-armed bandits and one can design algorithm with poly-logarithmic cumulative regret. Precisely for a constant $c \in \mathbb{R}$ depending on $\mathcal{X}$ and $\mathcal{C}$:

$$\mathbb{P}\Big[\forall n \geqslant 1,\ R_n \leqslant c\eta^2 \Delta^{-1} d^2 (u + \log^3 n)\Big] \geqslant 1 - e^{-u}. \tag{2.28}$$

The ConfidenceBall or LinRel algorithm (Auer, 2002; Dani et al., 2008) attains this regret by computing $\widehat{f}_n$ a least-square estimate of f and considering an ellipsoid centered in $\widehat{f}_n$ with small square loss $\mathcal{L}_n$:

$$\begin{aligned} \mathcal{L}_n(g) &\triangleq \sum_{i=1}^{n} \big(g^\top x_i - y_i\big)^2, \\ \widehat{f}_n &\triangleq \operatorname*{argmin}_{g \in \mathcal{C}} \mathcal{L}_n(g), \\ C_n &\triangleq \Big\{ g \in \mathcal{C} : \mathcal{L}_n(g) \leqslant \mathcal{L}_n(\widehat{f}_n) + \beta \Big\}, \end{aligned}$$

where $\beta^2 \triangleq \mathcal{O}\big(\eta^2 d \log n (u + \log n)\big)$ so that the following holds:

$$\mathbb{P}\Big[\forall n \geqslant 1,\ f \in C_n\Big] \geqslant 1 - e^{-u}.$$

The query is then selected to maximize the upper confidence bound:

$$x_{n+1} \triangleq \operatorname*{argmax}_{x \in \mathcal{X}} \max_{g \in C_n} g^\top x.$$

The OFUL algorithm (Abbasi-Yadkori et al., 2011) improves the regret bound by computing a regularized least-square and tight confidence ellipsoids. It is presented in Algorithm 4, where we used $\mathcal{L}_{n,\lambda}$ the regularized square loss and $\mathbf{C}_{n,\lambda}$ a re-scaled design matrix of the queries:

$$\begin{aligned} \mathcal{L}_{n,\lambda}(g) &\triangleq \mathcal{L}_n(g) + \lambda \|g\|_2\,, \\ \mathbf{C}_{n,\lambda} &\triangleq \lambda^{-1} X_n^\top X_n + \mathbf{I} \text{ where } X_n \triangleq \big[x_i\big]_{i \leqslant n}. \end{aligned}$$

The confidence ellipsoid C_n computed by the OFUL enjoys the same crucial property, that is

Algorithm 4: OFUL$(\mathcal{C}, \lambda, \eta, u)$

for $n = 0, 1, \dots$ **do**

$\widehat{g} \leftarrow \operatorname{argmin}_{g \in \mathcal{C}} \mathcal{L}_{n,\lambda}(g)$

$\beta^2 \leftarrow 2\eta^2(u + \log\det \mathbf{C}_{n,\lambda})$

$C_n \leftarrow \left\{ g \in \mathcal{C} : \mathcal{L}_{n,\lambda}(g) \leqslant \mathcal{L}_{n,\lambda}(\widehat{g}_n) + \beta + \lambda^{1/2} \sup_{g \in \mathcal{C}} \|g\|_2 \right\}$

$x_{n+1} \leftarrow \operatorname{argmax}_{x \in \mathcal{X}} \max_{g \in C_n} g^\top x$

$y_{n+1} \leftarrow$ **Query**(x_{n+1})

end

the unknown function f always lies in C_n with probability at least $1 - e^{-u}$. Its regret bound strictly improves the one of Eq. 2.28.

Polynomial Regret and Lower Bounds

When the gap Δ is 0, such as for a spherical $\mathcal{X}$, the order of the cumulative regret leave the poly-logarithmic magnitude and fall in a far worse polynomial range. Indeed there exists a bounded space $\mathcal{X}$ and a class $\mathcal{C}$ such that for every algorithm the following lower bound holds true for any iteration n (Dani et al., 2008):

$$\mathbb{E}\big[R_n\big] \geqslant \Omega\big(d\sqrt{n}\big). \tag{2.29}$$

The cumulative regret of the OFUL algorithm matches this lower bound up to poly-logarithmic multiplicative factors, which makes this procedure almost optimal even when $\Delta = 0$.

Upper and Lower Bounds for the Simple Regret

Compared to the fertile research in multi-armed bandit, the study of pure exploration for stochastic linear bandits seems to raise limited interest. Recently, Soare et al. (2014) and Soare (2015) adapted the analysis from Abbasi-Yadkori et al. (2011) and from preliminary analysis of Kaufmann et al. (2016) for the $(0, \delta)$-PAC setting in linear bandit, that is one wants to find with probability $1 - \delta$ the exact optimum. They came up with both lower and upper bounds for the case where the noise is bounded and $\mathcal{X}$ is finite. They show that the expected number τ of queries required to be $(0, \delta)$-PAC is lower bounded by:

$$\mathbb{E}[\tau] \geqslant \Omega\left(u \max_{\substack{x \in \mathcal{X} \\ \Delta_x > 0}} \frac{\|x^\star - x\|^2_{\mathbf{K}_\tau^{-1}}}{\Delta_x^2} \right),$$

where $u \triangleq \log \delta^{-1}$, $f(x^\star) = \max_{x \in \mathcal{X}} f(x)$ and $\mathbf{K}_\tau \triangleq \sum_{x \in \mathcal{X}} \mathbb{E}[N_\tau(x)/\tau] x x^\top$. The upper bound for the algorithms they propose does not match this lower bound, since additional quantities which may be leading terms are involved. Up to our knowledge, no result has been given for simple regret on non finite $\mathcal{X}$.

2.2.3 Lipschitzian Optimization

The linearity assumption on f is extremely restrictive and may limit the practicability of the former algorithms for real problems. Recently many authors proposed optimization algorithms that only require the function to be Lipschitz-continuous under the setting from

Section 2.1.3. In this view, the unknown function is assumed to satisfy Eq. 2.4 with known norm $\|f\|$. One cannot hope for cumulative regret better than $\mathcal{O}\big(d\sqrt{n}\big)$ in expectation since the lower bound from Eq. 2.29 still holds.

One-Dimensional Spaces

The first analysis of the cumulative regret for one-dimensional Lipschitz-continuous functions goes back to Agrawal (1995). In this work the authors exhibit an algorithm whose expected cumulative regret is at most $\mathcal{O}\big(n^{3/4}\big)$. Kleinberg (2004) refined this bound and proved almost matching lower bound. They show that under Gaussian noise, any algorithm incurs a regret at least:

$$\mathbb{E}\big[R_n\big] \geqslant \Omega\big(n^{2/3}\big).$$

They propose an algorithm for which the exponent is optimal, but with an additional logarithmic term,

$$\mathbb{E}\big[R_n\big] \leqslant \mathcal{O}\big(n^{2/3}\log^{1/3} n\big).$$

This lower bound can be overcome under additional restrictions on f. For instance if it possesses continuous second derivative, and the horizon n is fixed and known, then an extension of the UCB algorithm on discretized intervals obtain regret in $\mathcal{O}\big(\sqrt{n\log n}\big)$, and this is the best possible regret for this case (Auer et al., 2007). In the same respect, an extension of the KL-UCB algorithm is shown to obtain optimal cumulative regret with rates that explicitly describe distribution-dependent KL-divergences (Magureanu et al., 2014).

Multi-dimensional Spaces

The d-dimensional optimization of Lipschitz function is challenging, since the volume of the search space increases exponentially in d. We describe here several algorithms and associated regret bounds. In the above-mentioned article (Kleinberg, 2004), the authors also demonstrate that if f is convex, the exponential blowup can be avoided and replaced by polynomial growth. They obtain a cumulative regret of $\mathcal{O}\big(d^3n^{3/4}\big)$. Unfortunately, the convex hypothesis is too stringent for global optimization problems. For related assumptions such as unimodality of f, we refer to (Combes and Proutiere, 2014). For reasons explained in the beginning of this chapter, modern approaches introduce the nonparametric similarity measure ℓ for which f is assumed to be Lipschitz. In order to measure the size of such a search space, the most relevant quantity has shown to be the near-optimality dimension $\dim_\rho$ as defined in Eq. 2.10, or similar dimension such as the Zooming dimension (Kleinberg et al., 2008). Indeed in this article, the authors come up with a lower bound on the expected cumulative regret of:

$$\mathbb{E}\big[R_n\big] \geqslant \Omega\left(n^{\frac{d+1}{d+2}}\right), \tag{2.30}$$

for all $d > \dim_\rho(\mathcal{X}, \ell)$ (and $d = \dim_\rho(\mathcal{X}, \ell)$ if the dimension is attained), and a strategy called the Zooming algorithm with almost optimal regret when the horizon n is fixed and known:

$$\mathbb{E}\big[R_n\big] \leqslant \mathcal{O}\left(n^{\frac{d+1}{d+2}}\big(\log n\big)^{\frac{1}{d+2}}\right).$$

The fact that the upper and lower bounds differ only by a sub-logarithmic term indicates that the near-optimality dimension is the right measure of complexity. Note that in d-dimensional

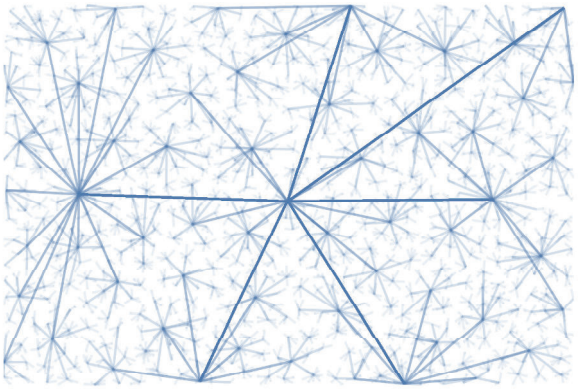


Figure 2.5. – A partitioning tree of a rectangle for the euclidean metric

euclidean $\mathcal{X}$ this dimension can be much smaller that d. As explained before, common scenarios enjoy $d = 0$, leading to $\mathbb{E}[R_n] \leqslant \mathcal{O}(\sqrt{n})$. Another strategy, called the HOO algorithm (Bubeck et al., 2008, 2011), refines and extends this work by constraining only the local behavior of f around the optimum (Eq. 2.5), and suppressing the knowledge of the horizon. This approach leads to equivalent upper and lower bound, but requires to know a hierarchical discretization of $\mathcal{X}$ which could be difficult to build if the metric space $(\mathcal{X}, \ell)$ is not parametric. When the learner observes the function without noise the HOO algorithm simplifies to the DOO algorithm (Munos, 2011), a tree-based search designed to obtain fast converging simple regret. The procedure takes as input a tree of partitions of $\mathcal{X}$ adapted to ℓ.

Partitioning Trees

First, let us formally define a discretization tree of $\mathcal{X}$. These definition are more general than what is typically found in the literature and will be reused in the next chapter.

Definition 2.6 (Discretization Tree)**.** *A sequence $\mathcal{T} \triangleq (\mathcal{T}_h)_{h \geqslant 0}$ with parent relation $p : \mathcal{X} \to \mathcal{X}$ is said to be a discretization tree of $\mathcal{X}$ when the following holds for all integer $h \geqslant 0$:*

1. $|\mathcal{T}_0| = 1$ *and* $|\mathcal{T}_h| < \infty$,
2. $\forall x \in \mathcal{T}_{h+1},\ p(x) \in \mathcal{T}_h$,
3. $\mathcal{T}_h \subset \mathcal{T}_{h+1}$,
4. $\lim_{h \to \infty} \mathcal{T}_h = \mathcal{X}$.

For $\big((\mathcal{T}_h)_{h \geqslant 0}, p\big)$ a discretization tree of $\mathcal{X}$, we use the notation $\mathrm{Children} \equiv \mathrm{Children}_{\mathcal{T}}$ for the set of children:

$$\forall h \geqslant 0, \forall s \in \mathcal{T}_h,\ \mathrm{Children}(s) \triangleq \Big\{ x \in \mathcal{T}_{h+1} : p(x) = s \Big\}.$$

In the sequel, we will use an abuse of notation and denote a discretization tree simply by $\mathcal{T}$. For all $h \geqslant 0$, we define the set of nodes at depth lower than h as follows:

$$\mathcal{T}_{\leqslant h} \triangleq \bigcup_{h' \leqslant h} \mathcal{T}_{h'}.$$

Algorithm 5: DOO with partitioning tree $\mathcal{T}$

$L \leftarrow \mathcal{T}_0$
for $r = 0, 1, \dots$ **do**
  $\forall x \in L,\ U(x) \leftarrow f(x) + \Delta(x)$
  $s \leftarrow \operatorname{argmax}_{x \in L} U(x)$
  for $x \in \mathrm{Children}(s)$ **do**
    $f(x) \leftarrow$ **Query**(x)
    $L \leftarrow L \cup \{x\}$
  end
  $L \leftarrow L \backslash \{s\}$
end

Finally, for all $h \geqslant 0$ and $a > 0$, we define the successor relation $(x \succ s) \equiv (x \succ_{\mathcal{T}} s)$ between $s \in \mathcal{T}_h$ and $x \in \mathcal{T}_{h+a}$ by:

$$x \succ s \text{ iff } \exists x_1 \in \mathrm{Children}(s), x_2 \in \mathrm{Children}(x_1), \dots \text{ s.t. } x \in \mathrm{Children}(x_{h+a-1}).$$

Such discretization trees permit to define hierarchical partitioning of $\mathcal{X}$ along the successor relation. By denoting $\mathrm{Cell} \equiv \mathrm{Cell}_{\mathcal{T}}$ the set of all the successor nodes:

$$\forall s \in \mathcal{X},\ \mathrm{Cell}(s) \triangleq \big\{x \in \mathcal{X} : x \succcurlyeq s\big\},$$

where the relation $\succcurlyeq$ extends the relation $\succ$ with the identity, we see that the set of cells at a given depth forms a partition:

$$\forall h \geqslant 0,\ \bigcup_{s \in \mathcal{T}_h} \mathrm{Cell}(s) = \mathcal{X},$$
$$\forall s, s' \in \mathcal{T}_h,\ s \neq s' \implies \mathrm{Cell}(s) \cap \mathrm{Cell}(s') = \varnothing.$$

We now define a partitioning tree of $\mathcal{X}$, for which the cells are nested between a larger and a smaller ℓ-balls of given radii.

Definition 2.7 (Partitioning Tree)**.** *Let $\mathcal{T}$ be a discretization tree of $\mathcal{X}$. We say that $\mathcal{T}$ is a (ℓ, δ, ρ)-partitioning tree of $\mathcal{X}$ for a function $\delta : \mathbb{N} \to \mathbb{R}$ and scalar $\rho > 0$ when the following holds for all $h \geqslant 0$ and $s \in \mathcal{T}_h$:*

1. $\mathrm{Cell}(s) \subseteq \mathcal{B}\big(s, \delta(h)\big)$,
2. $\mathcal{B}\big(s, \rho\delta(h)\big) \subseteq \mathrm{Cell}(s)$.

The ℓ-radius $\Delta(s) \equiv \Delta_{\mathcal{T}}(s)$ of the cell of a node is the largest distance between s and a point in the cell:

$$\forall s \in \mathcal{X},\ \Delta(s) \triangleq \sup_{x \succ s} \ell(s, x).$$

Let us assume that $\|f\| = 1$ without loss of generality. Knowing $\mathcal{T}$ a (ℓ, δ, ρ)-partitioning tree of $\mathcal{X}$, for a similarity ℓ satisfying the one-sided Lipschitz property from Eq. 2.6, the DOO

algorithm is described in Algorithm 5. It maintains a set of leafs of a finite sub-tree of $\mathcal{T}$. Thanks to the Lipschitz-continuity of f, the following property trivially holds:

$$\forall h \geqslant 0, \forall s \in \mathcal{T}_h, \quad \sup_{x \in \mathrm{Cell}(s)} f(x) - f(s) \leqslant \Delta(s) \leqslant \delta(h).$$

We deduce that the DOO algorithm only queries nodes with $U(x) \geqslant \sup_{x^\star \in \mathcal{X}} f(x^\star)$. Let I_h be the set of $\delta(h)$-optimal nodes at depth h:

$$I_h \triangleq \mathcal{T}_h \cap \mathcal{X}_{\delta(h)}.$$

As for the previous remark, the queries of DOO are focused in the children of $\bigcup_{h \geqslant 0} I_h$. When the round selects $s \in I_h$ such that $\sup_{x > s} f(x) = \sup_{x^\star \in \mathcal{X}} f(x^\star)$ the simple regret is at most $\delta(h)$, we can thus bound the regret of the DOO algorithm by the worst case which explores first all I_1 then I_2 and so on. We can now invoke the properties of partitioning trees and the near-optimality dimension of $\mathcal{X}$ to bound the number of nodes in I_h:

$$\forall d > \dim_\rho(\mathcal{X}, \ell), \exists c \in \mathbb{R}, \forall h \geqslant 0, \ |I_h| \leqslant c\delta(h)^{-d}.$$

We conclude for such d and c that the simple regret of DOO is bounded by:

$$S_n \leqslant \delta\big(h(n/K)\big),$$
$$\text{where } h(n) \triangleq \inf\left\{ h \in \mathbb{N} : c \sum_{h'=0}^{h} \delta(h')^{-d} \geqslant n \right\},$$
$$\text{and } K \triangleq \sup_{s \in \mathcal{X}} \big|\mathrm{Children}(s)\big|.$$

Finally, in the usual case where $-\log \delta(h) = \mathcal{O}(h)$ and $K = \mathcal{O}(1)$, we obtain $S_n \leqslant \mathcal{O}\big(e^{-n}\big)$ if $\dim_\rho(\mathcal{X}, \ell) = 0$, and $S_n \leqslant \mathcal{O}\big(n^{-1/d}\big)$ otherwise. The detailed proof can be found in Munos (2011).

Unknown Smoothness and Adaptive Algorithms

A compelling feature of this approach is that the previous algorithm can easily be adapted to the case where ℓ is not known, without paying a large cost for the regret. At each round, if we select simultaneously all the nodes having the greatest function value for their depth, then we are sure to select the maximizer of the unknown upper bound for the best possible δ. This modification is implemented in the SOO algorithm (Algorithm 6), which also takes as input a function $h_{\max} : \mathbb{N} \to \mathbb{N}$ preventing the sub-tree to grow linearly. By selecting $h_{\max}(r) \triangleq \sqrt{r}$, the SOO algorithm attains simple regrets close to the ones from DOO with optimal similarity ℓ. Note that the optimal similarity may depend on the function f itself, as discussed in Section 2.1.3. Let $d = \dim_\rho(\mathcal{X}, \ell)$ for the best possible ℓ and ρ. When $-\log \delta(h) = \mathcal{O}(h)$ and $K = \mathcal{O}(1)$, then SOO satisfies $S_n \leqslant \mathcal{O}\big(e^{-\sqrt{n}}\big)$ if $d = 0$, and $S_n \leqslant \mathcal{O}\big(n^{-1/2d}\big)$ otherwise. Finally, remark that despite the SOO algorithm is agnostic to the smoothness of the function, it requires a partitioning tree $\mathcal{T}$ adapted to ℓ. Computing a (ℓ, δ, ρ)-partitioning tree with optimal but unknown smoothness ℓ has no general solution. In Grill et al. (2015), the authors use

Algorithm 6: SOO with partitioning tree $\mathcal{T}$ and max height $h_{\max} : \mathbb{N} \to \mathbb{N}$

```
L ← {x_0}
for r = 0, 1, ... do
    v ← −∞
    for h = 0, ..., height(L) ∧ h_max(r) do
        s ← argmax_{x∈L∩T_h} f(x)
        if f(s) ⩾ v then
            for x ∈ Children(s) do
                f(x) ← Query(x)
                L ← L ∪ {x}
            end
            L ← L\{s}
            v ← f(s)
        end
    end
end
```

this technique together with UCB to face noisy observations without knowing the smoothness of the function. The expected simple regret they obtain is bounded by:

$$\mathbb{E}[S_n] \leqslant \mathcal{O}\left(n^{-\frac{1}{d+2}} (\log n)^{\frac{2}{d+2}}\right),$$

which is the almost optimal (up to logarithmic multiplicative factors) simple regret one can deduce from the lower bound 2.30. Finally, the independent work from Bull (2015) analyzes cumulative regret in a slightly modified assumption equivalent to functions with $\dim_\rho(\mathcal{X}, \ell) = 0$ and unknown smoothness. The obtain regret, $\mathsf{R}_n \approx \sqrt{n}$ up to logarithmic terms with high probability, is equivalent to the previous approaches.

2.2.4 Bayesian Optimization and Gaussian Processes

We conclude this chapter by providing known upper and lower bounds in the Bayesian optimization framework. As explained in Section 2.1.4, we do not consider here that f is a fixed function in a set of smooth functions, but we assume instead a probability distribution over functions. The randomness comes from the noise and the function itself. We refer to Bull (2011) and Srinivas et al. (2012) for analysis of the regrets incurred by a Bayesian algorithm on a fixed function in the corresponding RKHS. As discussed before functions in the RKHS are smoother than samples from the process.

Simple Regret for Gaussian Processes with Deterministic Observations

In the previous section we have seen that the optimization of a Lipschitz-continuous function without noise can be solved by algorithms with exponentially fast simple regret. When the function is a realization of a Gaussian process $\mathcal{GP}(0, k)$ with smooth but arbitrary kernel, such convergence rate are impossible to reach. In Grünewälder et al. (2010) the authors derive almost tight lower bounds when $\mathcal{X} \subset \mathbb{R}^d$ and the kernel k is Hölder-continuous with exponent

Algorithm 7: GP-UCB (k, η, u) on finite $\mathcal{X}$

for $n = 0, 1, \dots$ **do**
  Compute μ_n and σ_n^2 (Eq. 2.14, 2.16)
  $\beta \leftarrow 2 \log \left(|\mathcal{X}| n^2 \frac{\pi^2}{6} \right) + 2u$
  for $x \in \mathcal{X}$ **do**
    $U(x) \leftarrow \mu_n(x) + \sqrt{\beta \sigma_n^2(x)}$
  end
  $x_{n+1} \leftarrow \operatorname{argmax}_{x \in \mathcal{X}} U(x)$
  $y_{n+1} \leftarrow$ **Query**(x_{n+1})
end

α with respect to the supremum norm. They prove that it exists such Gaussian processes on which the expected simple regret of any algorithm is larger than:

$$\mathbb{E}[S_n] \geqslant \Omega\left(n^{-\frac{\alpha}{2d}} \log^{-\frac{1}{2}} n \right).$$

They show that the optimal algorithm when the horizon n is known can be computed by the impracticable resolutions of the nested integrals involved in the simple regret. Yet, they also demonstrate that a blind exploration of the search space obtains an almost optimal simple regret:

$$\mathbb{E}[S_n] \leqslant \mathcal{O}\left(n^{-\frac{\alpha}{2d}} \log^{\frac{1}{2}} n \right).$$

Under more restrictive hypothesis on the kernel, exponential rates are still possible. In this respect, de Freitas et al. (2012) analyze Gaussian process optimization on finite $\mathcal{X}$ where k is a stationary and four times differentiable kernel, and f is locally similar to $x \mapsto \sup_{x^\star} f(x^\star) - \|x^\star - x\|_2^2$ around its single global maximizer. They propose a branch-and-bound algorithm whose simple regret satisfies with high probability for $a > 0$:

$$S_n \leqslant \mathcal{O}\left(\log^{\frac{1}{2}} |\mathcal{X}| e^{-\frac{an}{\log^{d/4} n}} \right).$$

Finally, we refer to Vazquez and Bect (2010) and Bect et al. (2016) for convergence results on the Expected Improvement algorithm.

Simple and Cumulative Regrets with Noisy Observations

This section is dedicated to the most general optimization framework we consider in this dissertation, Bayesian optimization with noisy observations. As stated in the introduction, this setting is heavily used in many applications, yet the theoretical results are limited. Up to our knowledge, the only existing lower bound assumes Bayesian linear optimization, and all the upper bounds are restricted to smooth Gaussian processes.

Bayesian linear optimization with multi-variate normal prior is equivalent to Gaussian process optimization with linear kernel $k(x_1, x_2) = x_1^\top x_2$. In Rusmevichientong and Tsitsiklis (2010), it is shown that in this case for $\mathcal{X} \subset \mathbb{R}^d$, the expected cumulative regret of any algorithm agnostic to the time horizon is lower bounded by:

$$\mathbb{E}[R_n] \geqslant \Omega\left(d\sqrt{n} \right). \tag{2.31}$$

We note that this lower bound holds even when $\mathcal{X}$ is the unit sphere. Consequently there is no hope of getting better expected cumulative regret when the problem is not simpler than linear functions of the sphere. Srinivas et al. (2012) propose the GP-UCB algorithm which attains this lower bound with high probability in the linear case and extends to more general kernels, but require the knowledge of the probability of error e^{-u}. When the search space $\mathcal{X}$ is finite, the algorithm runs as the UCB Algorithm 1 where the U values are computed with Bayesian inference (Eq. 2.14 and Eq. 2.16), as displayed in Algorithm 7. The GP-UCB is guaranteed with probability at least $1 - e^{-u}$ to incur a cumulative regret lower than:

$$R_n \leqslant \mathcal{O}\left(\sqrt{n\gamma_n\big(u + \log(n|\mathcal{X}|)\big)}\right). \tag{2.32}$$

The quantity γ_n measures the cost of exploring the search space $\mathcal{X}$ with respect to the kernel k. It is formally defined as the maximum information gain on f obtainable by a set of n queries:

$$\gamma_n \equiv \gamma_n(\mathcal{X}, f, \epsilon) \triangleq \max_{\substack{X \subset \mathcal{X} \\ |X|=n}} I(X), \tag{2.33}$$

where I is the mutual information of f and the noisy observations $Y_X = \{f(x_i) + \epsilon_i\}_{x_i \in X}$:

$$I(X) \equiv I(X, f, \epsilon) \triangleq H\big(Y_X\big) - H\left(Y_X \,\middle|\, \big\{f(x_i)\big\}_{x_i \in X}\right),$$

with $H(Z) = \mathbb{E}[-\log p(Z)]$ the Shannon entropy of a random variable Z with distribution p, and $H(Z_1 \,|\, Z_2) = H((Z_1, Z_2)) - H(Z_2)$ the conditional entropy. For Gaussian processes, the Shannon entropy H has a simple form and the previous mutual information simplifies to:

$$I(X) = \frac{1}{2} \log\det\left(\mathbf{I} + \eta^{-2}\mathbf{K}_X\right), \tag{2.34}$$

where $\mathbf{K}_X \triangleq \big[k(x_1, x_2)\big]_{x_1, x_2 \in X}$ is the kernel matrix of the points in X and η^2 is the variance of the noise. In the worst case where the kernel k is a Kronecker delta, that is f is a Gaussian white noise process, $\gamma_n = \mathcal{O}(n)$ and the bound from Eq. 2.32 reflects the impossibility of optimizing such processes. For more usual kernels γ_n is sub-linear. Upper bounds on its values leads directly to upper bounds on the cumulative regret of the GP-UCB algorithm. When $\mathcal{X} \subset \mathbb{R}^d$, we have the following results with high probability, where the $\widetilde{\mathcal{O}}$ notation hides some logarithmic terms:

Linear kernel (Eq. 2.11):	$\gamma_n \leqslant \mathcal{O}\big(d \log n\big)$,	$R_n \leqslant \widetilde{\mathcal{O}}\big(\sqrt{dn}\big)$,
SE kernel (Eq. 2.12):	$\gamma_n \leqslant \mathcal{O}\Big(\log^{d+1} n\Big)$,	$R_n \leqslant \widetilde{\mathcal{O}}\Big(\sqrt{n \log^{d+1} n}\Big)$,
Matérn kernel with $\nu > 1$ (Eq. 2.13):	$\gamma_n \leqslant \mathcal{O}\Big(n^{\frac{d(d+1)}{2\nu + d(d+1)}}\Big)$,	$R_n \leqslant \widetilde{\mathcal{O}}\Big(n^{\frac{\nu + d(d+1)}{2\nu + d(d+1)}}\Big)$.

The regret in $\widetilde{\mathcal{O}}\big(\sqrt{dn}\big)$ for the linear case is actually better than the lower bound 2.31. This can be explained by the fact that this is a high probabilistic result and not an expectation, and the GP-UCB algorithm knows the error probability e^{-u}. Note also that the search space $\mathcal{X}$ is

finite. When $\mathcal{X}$ is not finite, the authors propose to adapt the algorithm by changing only the value of β. If the Lipschitz-norm of f has b-subgaussian tails, that is:

$$\exists a \in \mathbb{R}, \forall \lambda > 0,\ \mathbb{P}\Big[\ \|f\|_{\text{Lip}} > \lambda\Big] \leqslant a e^{-\frac{\lambda^2}{2b^2}},$$

and the values of a and b are known, then they obtain the following upper bound:

$$R_n \leqslant \mathcal{O}\Big(\sqrt{dn\gamma_n\big(u + \log(nb)\big)}\Big).$$

Knowing the exact value of b is not an easy task. In practice the authors perform complete cross-validations on β. Examples of Gaussian processes with b-subgaussian Lipschitz-norm are processes whose kernel is stationary and four times differentiable. As mentioned in Section 2.1.4, this does not holds for the Matérn kernel if $\nu \leqslant 1$ such as the Ornsetin-Uhlenbeck kernel. The high probabilistic upper bounds on the cumulative regret follow:

Linear kernel (Eq. 2.11):	$R_n \leqslant \widetilde{\mathcal{O}}\big(d\sqrt{n}\big),$
SE kernel (Eq. 2.12):	$R_n \leqslant \widetilde{\mathcal{O}}\Big(\sqrt{dn\log^{d+1} n}\Big),$
Matérn kernel with $\nu > 1$ (Eq. 2.13):	$R_n \leqslant \widetilde{\mathcal{O}}\Big(d^{\frac{1}{2}} n^{\frac{\nu+d(d+1)}{2\nu+d(d+1)}}\Big).$

The convergence rate of the simple regret one can deduce from these bounds remains good for the linear kernel. Otherwise, this degrades rapidly with the dimension. As an example for the squared exponential kernel with $d = 4$ the number of iterations needed to have $\sqrt{dn^{-1}\log^{d+1} n} < 0.1$ is $n > 10^9$, for the Matérn kernel with $\nu = 5/2$ we need $n > 10^{13}$. The direct analysis of the simple regret with noisy observations is fairly limited. Up to our knowledge, the only theoretical work not built on the GP-UCB algorithm comes from Hoffman et al. (2014). They analyze the case where f is linear and $\mathcal{X}$ is finite. By denoting $K \triangleq |\mathcal{X}|$, they propose an algorithm satisfying the following simple regret:

$$\mathbb{P}\left[S_n \leqslant \sqrt{\frac{2\eta^2 K\big(u + \log(nK)\big)}{n - K}}\right] \geqslant 1 - e^{-u}.$$

This previous article explores finer distribution-dependent regret bounds involving the sum of the squared inverse gaps over the arms, similarly to the classical multi-armed bandit framework as in Eq. 2.27.

Advances in Bayesian Optimization

3

This chapter describes novel advances in Bayesian optimization. We first consider in Section 3.1 a contribution on optimization using mini-batches of queries at each iteration. In Section 3.2, we then rigorously study Gaussian process optimization on continuous space via geometrical arguments. We finally examine in Section 3.3 Bayesian optimization with non-Gaussian stochastic processes. We show that our algorithms adapt easily to various more complex priors that are used in natural applications. A significant part of the work from this chapter has been published in Contal et al. (2013) and Contal et al. (2015).

Contents

3.1 Batch Sequential Optimization **63**
3.1.1 Problem Formulation and Objectives 63
Objectives 63
3.1.2 Parallel Optimization Procedure 64
Relevant Region 64
The GP-UCB-PE Algorithm 65
Numerical Complexity 66
3.1.3 Theoretical Analysis 66
Upper Bound on the Cumulative Regret 66
Discussion and Comparison with the Sequential Setting 66
Proofs of the Upper Bound on the Cumulative Regret 67
3.1.4 Experiments 70
Protocol 71
Description of Data Sets 71
Comparison of Algorithms 72
3.1.5 Conclusion and Discussion 72
3.2 Gaussian Processes in Metric Spaces **73**
3.2.1 Hierarchical Discretizations of the Search Space 73
The Canonical Pseudo-Metric of Gaussian Processes 73
Generic Chaining 74
Geometric Interpretation and Classical Chaining 75
Bounding with the Covering Dimension 76
3.2.2 Regret Bounds for Bandit Algorithms 77
UCB Computed with Classical Chaining 77
UCB on Adaptive Discretizations 80
Choice of the Discretization Depth 81
3.2.3 Efficient Algorithms 82
Greedy Cover 82
UCB on Greedily Grown Tree 82
Computations on Non-Finite Compact Spaces 82
3.2.4 Tightness Results on Discretization Trees 84
A High Probabilistic Lower Bound on the Supremum 84
Pruning the Discretization Tree to Obtain a Balanced Tree 84
Computing the Pruning Values and Anti-Concentration Inequalities 85
Gaussian Processes Indexed by Ellipsoids 85
3.2.5 Proof of the Generic Chaining Lower Bound 87

Probabilistic Tools for Gaussian Processes 87
Proof of the Lower Bound 88
3.2.6 Conclusion and Discussions 90
3.3 Beyond Gaussian Processes **90**
3.3.1 Generic Stochastic Processes 90
Generic Chaining for Stochastic Processes 91
Classical Chaining for (ℓ, ψ)-Processes and Sub-Gamma Processes 91
High Confidence Empirical Intervals 92
3.3.2 Quadratic Forms of Gaussian Processes 93
The Stochastic Smoothness of a Sum of Squared Gaussian Processes 94
Confidence Intervals for Squared Gaussian Processes 94
3.3.3 Conclusion and Discussions 96

3.1 Batch Sequential Optimization

In this section, we focus on the case where the unknown function can be evaluated in parallel with mini-batches of fixed size and analyze the benefits compared to the purely sequential procedure in terms of cumulative regret. We present the Gaussian Process Upper Confidence Bound and Pure Exploration algorithm (GP-UCB-PE), which combines the UCB strategy and Pure Exploration in the same batch of evaluations along the parallel iterations. We prove theoretical upper bounds on the regret with batches of size K for this procedure which reveals the improvement of the order of $\sqrt{K}$ for fixed iteration cost over purely sequential versions. Moreover, the multiplicative constants involved have the property of being dimension-free. We also confirm empirically the efficiency of GP-UCB-PE on real and synthetic problems compared to state-of-the-art competitors.

3.1.1 Problem Formulation and Objectives

In some optimization scenarios, it is possible to evaluate the function in parallel with batches of K queries with no increase in cost. This is typically the case in the sensors location problem if K sensors are available at each iteration, or in the numerical optimization problem on a cluster of K machines, or recommendation systems with batches of K customers. Parallel strategies have been developed recently in Azimi et al. (2010) or Desautels et al. (2012). We propose to explore further the potential of parallel strategies for noisy function optimization with unknown horizon aiming simultaneously at practical efficiency and plausible theoretical results. The novel algorithm we introduce, called GP-UCB-PE, combines the benefits of the UCB policy with Pure Exploration queries in the same batch of K evaluations of f. The Pure Exploration component helps to reduce the entropy of f around the maximum in order to support the UCB policy in finding the location of the maximum, and therefore in increasing the decay of the regret R_n at every iteration n. In comparison to other algorithms based on Gaussian processes and UCB such as GP-BUCB (Desautels et al., 2012), the new algorithm discards the need for the initialization phase and offers a tighter control on the uncertainty parameter which monitors overconfidence. As a result, the derived regret bounds are more robust against the curse of dimensionality since the multiplicative constants obtained are dimension free in contrast with the doubly exponential dependence observed in previous work. We also mention that Monte-Carlo simulations can be proposed as an alternative and this idea has been implemented in the SimulationMatching algorithm with UCB policy (SM-UCB) (Azimi et al., 2010) which we also consider for comparison in the present document. Unlike GP-BUCB, no theoretical guarantees for the SM-UCB algorithm are known for the bounds on the number of iterations needed to get close enough to the maximum, therefore the discussion will be reduced to empirical comparisons over several benchmark problems.

Objectives

At each iteration n, we choose a batch of K points in $\mathcal{X}$ called the queries $\{x_{n,k}\}_{0 \leqslant k < K}$, and then observe simultaneously the noisy values taken by f at these points, $y_{n,k} \triangleq f(x_{n,k}) + \epsilon_{n,k}$. We denote by $r_{n,k}$ instantaneous regret, that is the difference between the optimum of f and the point queried $x_{n,k}$,

$$r_{n,k} \triangleq \sup_{x \in \mathcal{X}} f(x) - f(x_{n,k}).$$

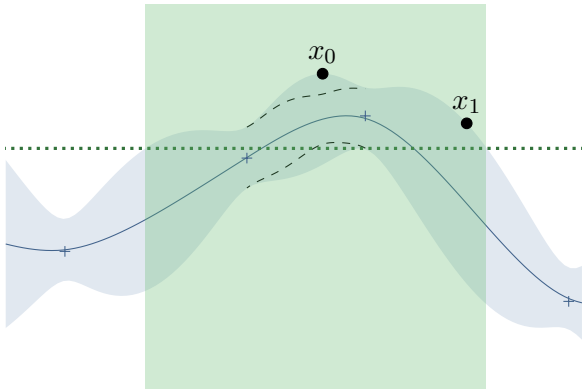


Figure 3.1. – The first two queries of GP-UCB-PE in a batch. x_0 maximize $U_n(\cdot)$, the horizontal dotted line is at $\widehat{y}_n$, the relevant region $\mathfrak{R}_n$ (green area) contains all the points such that $U_n(\cdot) \geqslant \widehat{y}_n$, the dashed lines illustrates the updated deviation $\sigma_{n,1}(\cdot)$ after choosing x_0, and x_1 maximizes $\sigma_{n,1}(\cdot)$ in $\mathfrak{R}_n$.

We aim at minimizing either the full cumulative regret:

$$R_{n,K} \triangleq \sum_{i \leqslant n} \sum_{k < K} r_{i,k},$$

which accounts for the case where all the queries in a batch should have a low regret; or the batch cumulative regret:

$$\widetilde{R}_{n,K} \triangleq \sum_{i \leqslant n} \min_{k < K} r_{i,k},$$

for the case where the cost for a batch of evaluations is fixed. An upper bound on $\widetilde{R}_{n,K}$ gives an upper bound of $n^{-1}\widetilde{R}_{n,K}$ on the simple regret, the minimum gap between the best point found so far and the true maximum. We restrict our analysis to the case where $\mathcal{X}$ is finite. Note that it is easily adaptable for a compact and convex $\mathcal{X}$ when the Lipschitz norm of the Gaussian process has subgaussian tail with known variance, and one contribution of the thesis in the next section will be to remove this hypothesis.

3.1.2 Parallel Optimization Procedure

We describe here the GP-UCB-PE algorithm, which will be shown later to satisfy theoretical guarantees on the three notions of regrets: full cumulative, batch cumulative and simple regret. Our idea is to focus all the queries in a region of the search space called the relevant region, where the maximizer will lie with high probability.

Relevant Region

We first recall $\mu_n(\cdot)$ and $\sigma_n^2(\cdot)$ the posterior expectation and variance computed with the observations obtained after n iterations with Eq. 2.14, 2.16. They allow to define the high confidence upper and lower bounds $U_n(\cdot)$ and $L_n(\cdot)$ with Eq. 2.19, 2.20:

$$U_n(x) \triangleq \mu_n(x) + \sqrt{\beta_n \sigma_n^2(x)},$$
$$L_n(x) \triangleq \mu_n(x) - \sqrt{\beta_n \sigma_n^2(x)}.$$

Algorithm 8: GP-UCB-PE (k, η, u)

$\mathfrak{R}_0 \leftarrow \mathcal{X}$
for $n = 0, 1, \ldots$ **do**
  Compute μ_n, σ_n^2, U_n and L_n (Eq. 2.14, 2.16, 2.19, 2.20)
  $x_{n+1,0} \leftarrow \operatorname{argmax}_{x \in \mathcal{X}} U_n(x)$
  $\mathfrak{R}_n \leftarrow \mathfrak{R}_{n-1} \backslash \left\{ x \in \mathfrak{R}_{n-1} : U_n(x) < \sup_{x \in \mathcal{X}} L_n(x) \right\}$
  for $k = 1, \ldots, K-1$ **do**
    Compute $\sigma_{n,k}^2$
    $x_{n+1,k} \leftarrow \operatorname{argmax}_{x \in \mathfrak{R}_n} \sigma_{n,k}^2(x)$
  end
  $\left\{ y_{n,k} \right\}_{k<K} \leftarrow$ **Query**$\left(\left\{ x_{n,k} \right\}_{k<K} \right)$
end

We define the relevant region $\mathfrak{R}_n$ being the region which contains the maximizer(s) of f with high probability. Let $\widehat{y}_n$ be our lower confidence bound on the maximum:

$$\widehat{y}_n \triangleq \sup_{x \in \mathcal{X}} L_n(x). \tag{3.1}$$

The value of $\widehat{y}_n$ is represented by the horizontal dotted green line on Figure 3.1. The relevant region is defined as the set of points whose UCB is larger than the best LCB:

$$\mathfrak{R}_n \triangleq \left\{ x \in \mathcal{X} : U_n(x) \geqslant \widehat{y}_n \right\}.$$

This region discards the locations where the maximizer does not belong with high probability. It is represented in green on Figure 3.1. We refer to de Freitas et al. (2012) for related use of relevant regions in the special case of deterministic Gaussian Process bandits.

The GP-UCB-PE Algorithm

We present here the Gaussian Process Upper Confidence Bound with Pure Exploration algorithm, GP-UCB-PE, a novel algorithm combining two strategies to determine the queries $\{x_{n,k}\}_{k<K}$ for batches of size K. The first location is chosen according to the GP-UCB rule,

$$x_{n+1,0} \in \operatorname*{argmax}_{x \in \mathcal{X}} U_n(x). \tag{3.2}$$

As described in the previous chapter, this single rule is enough to tackle the exploration-exploitation tradeoff. The value of β_n, fixed in Eq. 2.21, governs the trade-off between exploring uncertain regions (high posterior variance σ_n^2) and focusing on the supposed location of the maximum (high posterior mean μ_n). This policy is illustrated with the point x_0 on Figure 3.1. The $K-1$ remaining locations are selected via Pure Exploration restricted to the relevant region $\mathfrak{R}_n$. We aim to maximize $I_n(X_{n+1,K-1})$, the information gain on f by the observations at locations $X_{n+1,K-1} = \left\{ x_{n+1,k} \right\}_{1 \leqslant k < K}$ as defined in Eq. 2.34, conditioned on the observations so far $\mathbf{Y}_n$ at X_n:

$$I_n(X) \triangleq H\big(f \,|\, \mathbf{Y}_n, X_n\big) - H\Big(f \,|\, \mathbf{Y}_n, X_n, \left\{ f(x_{i,k}) + \epsilon_{i,k} \right\}_{x_{i,k} \in X}\Big). \tag{3.3}$$

Finding the $K-1$ points that maximize I_n for any integer K is known to be NP-complete (Ko et al., 1995). However, due to the submodularity of I_n (Guestrin et al., 2005), it can be efficiently approximated by the greedy procedure which selects the points one by one and never backtracks. The location of the single point that maximizes the information gain is easily computed by maximizing the posterior variance. For all $1 \leqslant k < K$ our greedy strategy selects the following points one by one:

$$x_{n+1,k} \in \operatorname*{argmax}_{x \in \mathfrak{R}_n} \sigma^2_{n,k}(x), \tag{3.4}$$

where $\sigma^2_{n,k}$ is the updated variance after choosing $\{x_{n+1,k'}\}_{k'<k}$. We use here the fact that the posterior variance does not depend on the values $y_{n+1,k}$ of the observations, but only on their position $x_{n+1,k}$. One such point is illustrated with x_1 on Figure 3.1. These $K-1$ locations reduce the uncertainty about f, improving the guesses of the UCB procedure by $x_{n+1,0}$. The overall procedure is shown in Algorithm 8.

Numerical Complexity

Even if the numerical cost of GP-UCB-PE is often not significant in practice compared to the cost of the evaluation of f, the complexity of the exact update of the variances (Eq.2.16) is in $\mathcal{O}\big((nK)^2\big)$, as discussed in Chapter 5, and might by prohibitive for large nK. One can reduce drastically the computation time by means of Lazy Variance Calculation (Desautels et al., 2012), built on the fact that $\sigma^2_n(x)$ always decreases when n increases for all $x \in \mathcal{X}$.

3.1.3 Theoretical Analysis

The main theoretical result of this section is the upper bound on the regret formulated in Theorem 3.1.

Upper Bound on the Cumulative Regret

The regret bound are expressed in term of γ_{nK}, the maximum information gain from Eq. 2.33 obtainable by a sequence of nK queries. Under these assumptions, we obtain the following result.

Theorem 3.1 (Regret Bound for GP-UCB-PE). *Fix $u > 0$ and consider the calibration of β_n defined before (Eq. 2.21), assuming $f \sim \mathcal{GP}(0,k)$ with bounded variance, $\forall x \in \mathcal{X},\ k(x,x) \leqslant 1$, then the full cumulative regret $R_{n,K}$ incurred by GP-UCB-PE on f is bounded by $\mathcal{O}\big(\sqrt{nK\beta_n\gamma_{nK}}\big)$. More precisely with $c_\eta \triangleq \frac{2}{\log(1+\eta^{-2})}$ where η^2 is the variance of the noise, we have,*

$$\mathbb{P}\left[\forall n \geqslant 1,\ R_{n,K} \leqslant 4\sqrt{c_\eta(n-1)K\beta_n\gamma_{nK} + K\beta_0}\right] \geqslant 1 - e^{-u}.$$

For the batch cumulative regret we obtain similar bounds,

$$\mathbb{P}\left[\forall n \geqslant 1,\ \widetilde{R}_{n,K} \leqslant 2\sqrt{c_\eta \frac{n}{K}\beta_n\gamma_{nK}}\right] \geqslant 1 - e^{-u}.$$

Discussion and Comparison with the Sequential Setting

When $K \ll n$ and $\gamma_n \ll n$, the upper bound for $\widetilde{R}_{n,K}$ is better than R_n for sequential GP-UCB by an order of $\sqrt{K}$. For the simple regret, the upper bound we derive differs only by a

	GP-UCB-PE	GP-BUCB
$R_{n,K}$	$\sqrt{nK\gamma_{nK}\log n}$	$c\sqrt{nK\gamma_{nK}\log nK}$
$\widetilde{R}_{n,K}$	$\sqrt{\frac{n}{K}\gamma_{nK}\log n}$	$c\sqrt{\frac{n}{K}\gamma_{nK}\log nK}$

Kernel	Linear	SquaredExp	Matérn
γ_{nK}	$d\log nK$	$\log^{d+1} nK$	$(nK)^{\alpha}\log nK$
c	$\exp(\frac{2}{e})$	$\exp((\frac{2d}{e})^d)$	e

Table 3.1. – General forms of regret bounds for GP-UCB-PE and GP-BUCB

universal multiplicative constant from the simple regret obtained when the observations are not delayed. Likewise when the regrets for all the points in the batch matter, the full cumulative regret $R_{n,K}$ is equivalent up to universal multiplicative constant from the upper bound on R_{nK} for GP-UCB.

Compared to Desautels et al. (2012), we remove the need of the initialization phase. Furthermore GP-UCB-PE does not need to multiply the uncertainty parameter β_n by $\exp(\gamma_{nK}^{\text{init}})$ where $\gamma_{nK}^{\text{init}}$ is equal to the maximum information gain obtainable by a sequence of nK queries after the initialization phase. The improvement can be doubly exponential in the dimension d in the case of squared exponential kernels. The values of γ_{nK} for different common kernels are reported in Table 3.1, where d is the dimension of the space considered and $\alpha \triangleq \frac{d(d+1)}{2\nu+d(d+1)} \leqslant 1$, ν being the Matérn parameter. We also compare on Table 3.1 the general forms of the bounds for the regret obtained by GP-UCB-PE and GP-BUCB up to constant terms. The cumulative regret we obtained with squared exponential kernel is of the form $\widetilde{\mathcal{O}}\left(\sqrt{\frac{n}{K}\log^d nK}\right)$ against $\widetilde{\mathcal{O}}\left(\exp((\frac{2d}{e})^d)\sqrt{\frac{n}{K}\log^d nK}\right)$ for GP-BUCB.

Proofs of the Upper Bound on the Cumulative Regret

In this section, we analyze theoretically the regret bounds for the GP-UCB-PE algorithm. We provide here the main steps for the proof of Theorem 3.1. On one side the UCB rule of the algorithm provides a regret bounded by the information we have on f conditioned on the values observed so far. On the other side, the Pure Exploration part gathers information and therefore accelerates the decrease in uncertainty. We refer to Desautels et al. (2012) for the proofs of the bounds for GP-BUCB. Thanks to the relevant region, we express the regret incurred by the queries $x_{n,k}$ in terms of the query $x_{n,0}$ chosen by the UCB rule. This enables to adapt the proof of the GP-UCB algorithm to this batch algorithm. Let $u > 0$ be fixed. In what follows, *with high probability* means *with probability at least* $1 - e^{-u}$, and we use the notations:

$$s_n \equiv s_{n,0} \triangleq \sqrt{\sigma^2_{n-1}(x_{n,0})},$$
$$\text{and } s_{n,k} \triangleq \sqrt{\sigma^2_{n-1,k}(x_{n,k})} \text{ for } 1 \leqslant k < K.$$

Lemma 3.1 (Regret Bound for UCB query)**.** *With β_n calibrated as in Eq. 2.21, with high probability,*

$$r_{n,0} \leqslant 2\sqrt{\beta_{n-1}}s_n.$$

Proof. As seen in the previous chapter in Eq. 2.22, this calibration of β_n produces the following inequality with high probability:

$$\forall x \in \mathcal{X}, \forall n \geqslant 1, \ |f(x) - \mu_n(x)| \leqslant \sqrt{\beta_n \sigma_n^2(x)}.$$

Under this event, since $x_{n+1,0} \in \operatorname{argmax}_{x \in \mathcal{X}} U_n(x)$, we directly obtain:

$$\sup_{x^\star \in \mathcal{X}} f(x^\star) - f(x_{n+1}) \leqslant U_n(x_{n+1}) - L_n(x_{n+1}) \leqslant 2\sqrt{\beta_n} s_{n+1}.$$

□

We first show an intermediate result bounding the posterior variance at the points $x_{n+1,0}$ by the one at the points $x_{n,K-1}$.

Lemma 3.2 (Decrease of Posterior Variances)**.** *The posterior variance of the point selected by the UCB policy satisfies the following inequality with high probability:*

$$\forall n \geqslant 1, \ s_{n+1,0} \leqslant s_{n,K-1}.$$

Proof. By the definitions of $x_{n+1,0}$ from Eq. 3.2 and $\widehat{y}_n$ from Eq. 3.1, we have,

$$U_n(x_{n+1,0}) \geqslant \widehat{y}_n,$$

thus $x_{n+1,0} \in \mathfrak{R}_n \subseteq \mathfrak{R}_{n-1}$. We have as a result of the definition of $x_{n,K-1}$ from Eq. 3.4 that,

$$\sigma_{n-1,K-1}(x_{n+1,0}) \leqslant s_{n,K-1}.$$

Using the "information never hurts" principle (Krause and Guestrin, 2005), we know that the entropy of $f(x)$ for all location x decreases while we observe f at points $x_{n,k}$. For Gaussian processes, the entropy is also a non-decreasing function of the variance, so that:

$$\forall x \in \mathcal{X}, \ \sigma_{n,0}(x) \leqslant \sigma_{n-1,K-1}(x).$$

We thus prove $s_{n+1,0} \leqslant s_{n,K-1}$. □

We now use this lemma to prove an inequality between the sum of the posterior deviations.

Lemma 3.3 (Sum of Posterior Variances Comparison)**.** *The sum of the posterior deviations of the points selected by the UCB policy are bounded by the sum of the average of the ones for all the selected points. With high probability, for all $n \geqslant 1$,*

$$\sum_{i=1}^{n} s_{i,0} \leqslant K^{-1} \sum_{i=1}^{n} \sum_{k<K} s_{i,k}.$$

Proof. Using Lemma 3.2 and the definitions of $x_{i,k}$, we have that $s_{i+1,0} \leqslant s_{i,k}$ for all $k \geqslant 1$. Summing over k, we get for all $i \geqslant 1$, $(K-1)s_{i+1,0} \leqslant \sum_{k=1}^{K-1} s_{i,k}$. Now, summing over i we obtain the desired result. □

Next, we can bound the sum of all posterior variances via the maximum information gain for a sequence of nK locations.

Lemma 3.4 (Maximum Information Gain). *The sum of the posterior variances of the selected points are bounded by a constant factor times γ_{nK}. With $c_\eta \triangleq \frac{2}{\log(1+\eta^{-2})}$,*

$$\sum_{i=1}^{n} \sum_{k<K} s_{t,k}^2 \leqslant c_\eta \gamma_{nK},$$

where γ_{nK} is the informational quantity defined in Eq. 2.33.

Proof. We know that the information gain for a sequence of n locations x_i can be expressed in terms of the posterior variances $\sigma_i^2(x_i)$. The deviations $s_{i,k}$ being independent of the observations $y_{i,k}$, the same equality holds for the updated posterior variances $s_{i,k}^2$. See Lemmas 5.3 and 5.4 in Srinivas et al. (2012) for the detailed proof. □

We are now ready to give an upper on the full and batch cumulative regret of our algorithm.

Lemma 3.5 (Batch Cumulative Regret). *The batch cumulative regret incurred by the GP-UCB-PE can be bounded in terms of the maximum information gain with high probability,*

$$\widetilde{R}_{n,K} \leqslant 2\sqrt{c_\eta \frac{n}{K} \beta_n \gamma_{nK}}.$$

Proof. Using the previous lemmas and the fact that $\beta_i \leqslant \beta_n$ for all $i \leqslant n$, we have with high probability,

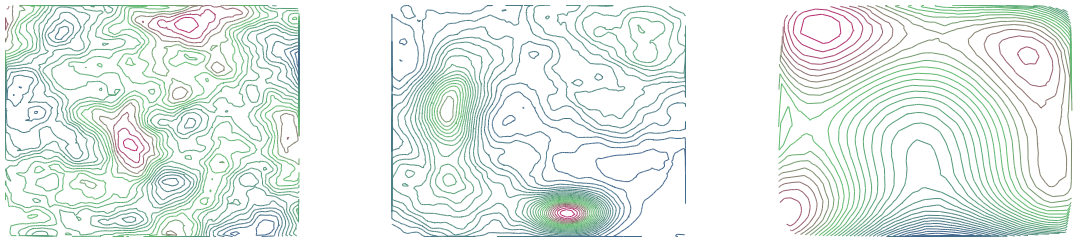
$$\begin{aligned}
\widetilde{R}_{n,K} = \sum_{i \leqslant n} \min_{k<K} r_{i,k} &\leqslant \sum_{i \leqslant n} r_{i,0} \\
&\leqslant \sum_{i \leqslant n} 2\sqrt{\beta_{n-1}} s_i, \text{ by Lemma 3.1} \\
&\leqslant 2\sqrt{\beta_n} K^{-1} \sum_{i \leqslant n} \sum_{k<K} s_{i,k}, \text{ by Lemma 3.3} \\
&\leqslant 2\sqrt{\beta_n} K^{-1} \sqrt{nK \sum_{i \leqslant n} \sum_{k<K} s_{i,k}^2}, \text{ by Cauchy-Schwarz} \\
&\leqslant 2\sqrt{\beta_n} K^{-1} \sqrt{nK c_\eta \gamma_{nK}}, \text{ by Lemma 3.4} \\
&\leqslant 2\sqrt{c_\eta \frac{n}{K} \beta_n \gamma_{nK}}.
\end{aligned}$$

□

Lemma 3.5 concludes the proof of Theorem 3.1 for the regret $\widetilde{R}_{n,K}$. The analysis for $R_{n,K}$ is similar, using the same steps to bound the regret for the Pure Exploration queries.

Lemma 3.6 (Full Cumulative Regret). *The regret for the queries $x_{t,k}$ selected by Pure Exploration in $\mathfrak{R}_n$ are bounded with high probability by,*

$$r_{n,k} \leqslant 2\sqrt{\beta_{n-1}} \big(s_n + \sigma_{n-1}(x_{n,k})\big).$$



Gaussian process | Gaussian mixture | Himmelblau

Figure 3.2. – Visualization of the synthetic functions used for assessment

Proof. Under the event of Eq. 2.22, which holds with high probability, the values of f are included in their confidence intervals $\mu_n(\cdot) \pm \sqrt{\beta_n}\sigma_n(\cdot)$. Therefore for all $n \geqslant 1$ and $k \leqslant K$,

$$\begin{aligned}
r_{n,k} &\leqslant \sup_{x^\star \in \mathcal{X}} \mu_{n-1}(x^\star) + \sqrt{\beta_{n-1}}\sigma_{n-1}(x^\star) - \mu_{n-1}(x_{n,k}) + \sqrt{\beta_{n-1}}\sigma_{n-1}(x_{n,k}) \\
&\leqslant \mu_{n-1}(x_{n,0}) + \sqrt{\beta_{n-1}}s_n - \mu_{n-1}(x_{n,k}) + \sqrt{\beta_{n-1}}\sigma_{n-1}(x_{n,k}) \\
&\leqslant \widehat{y}_{n-1} + 2\sqrt{\beta_{n-1}}s_n - \mu_{n-1}(x_{n,k}) + \sqrt{\beta_{n-1}}\sigma_{n-1}(x_{n,k}) \\
&\leqslant \mu_{n-1}(x_{n,k}) + \sqrt{\beta_{n-1}}\sigma_{n-1}(x_{n,k}) + 2\sqrt{\beta_{n-1}}s_n - \mu_{n-1}(x_{n,k}) + \sqrt{\beta_{n-1}}\sigma_{n-1}(x_{n,k}) \\
&\leqslant 2\sqrt{\beta_{n-1}}\sigma_{n-1}(x_{n,k}) + 2\sqrt{\beta_{n-1}}s_n \\
&\leqslant 2\sqrt{\beta_{n-1}}\left(s_n + \sigma_{n-1}(x_{n,k})\right),
\end{aligned}$$

where used only the definitions of $x_{n,0}$ and the properties of L_{n-1} and $\widehat{y}_{n-1}$. □

Now as before, we have that $\sigma_{n-1}(x_{n,k}) \leqslant s_{n-1,K-1}$ that is for $n > 1$,

$$r_{n,k} \leqslant 2\sqrt{\beta_{n-1}}\big(s_n + s_{n-1,k}\big).$$

Therefore, summing over k,

$$\sum_{k<K} r_{n,k} \leqslant 4\sqrt{\beta_n} \sum_{k<K} s_{n-1,k}.$$

To conclude the analysis of R_{TK} and prove Theorem 3.1, it suffices to use the last four steps of Lemma 3.5, with an additional constant term coming from the shifting of the indexes.

3.1.4 Experiments

We compare the empirical performances of our algorithm against two global optimization algorithms by batches, GP-BUCB (Desautels et al., 2012) and SM-UCB (Azimi et al., 2010). The GP-BUCB algorithm selects the queries by pure exploration for the first n_0 iterations, where n_0 is a predefined number depending on K, η and the dimension d, and then follows the UCB rule with updated variance. The SM-UCB algorithm attempts to match the expected queries of the UCB rule approximated by repeated simulations. The batch of queries is then selected with the K-medoids algorithm to optimize a matching cost.

Protocol

The tasks used for assessment come from three real applications and three synthetic problems described here. The results are shown in Figure 3.3. For all data sets and algorithms, the size of the batches K was set to 10 and the learners were initialized with a random subset of 20 observations $\{(x_i, y_i)\}$. The curves on Figure 3.3 show the evolution of the simple regret S_n in term of iteration n. We report the average value with the confidence interval over 64 experiments. The parameters for the prior distribution, like the bandwidth of the kernel, were chosen through the maximization of the marginal likelihood.

Description of Data Sets

Gaussian processes. This assessment corresponds to functions randomly generated by a Gaussian process with Matérn kernel of parameter $\nu = 3/2$ (Eq. 2.13). The search space was set to $\mathcal{X} = [0, 4]^2$ and the noise variance η^2 set to 10^{-2}.

Gaussian Mixture. This synthetic function comes from the addition of three two-dimensional Gaussian functions, in $\mathcal{X} = [0, 1]^2$, at $(0.2, 0.5)$, $(0.9, 0.9)$, and the maximum at $(0.6, 0.1)$. We then perturb these Gaussian functions with smooth variations generated from a Gaussian Process with Matérn kernel $\nu = 3/2$ and very few noise, $\eta^2 = 10^{-2}$. It is shown on Figure 3.2 (middle). The highest peak being thin, the sequential search for the maximum of this function is quite challenging.

Himmelblau Function. The Himmelblau function is another synthetic function in dimension two. We compute a slightly tilted version of the Himmelblau's function, and take the opposite to match the challenge of finding its maximum. This function presents four peaks but only one global maximum. It gives a practical way to test the ability of a strategy to manage exploration/exploitation tradeoffs. It is represented in Figure 3.2 (right).

Mackey-Glass Function. The Mackey-Glass function is the solution of a delay-differential equation which describes a chaotic system in dimension six, but without noise. It models real feedback systems and is used in physiological domains such as hematology, cardiology, neurology, and psychiatry. The highly chaotic behavior of this function makes it an exceptionally difficult optimization problem. It has been used as a benchmark for example by Flake and Lawrence (2002).

Tsunamis. This data set is a real optimization challenge coming from Stefanakis et al. (2012) and Stefanakis et al. (2014), presented in more details in Section 5.2.1. The goal is to optimize the five physical parameters of an island and a sloping beach to find the maximal amplification of a tsunami. Since this problem is too complex to be addressed analytically, the amplification is computed by numerical solution of the nonlinear shallow water equations.

Abalone. The challenge of the Abalone data set is to predict the age of a specie of sea snails from physical measurements. It comes from the study by Nash et al. (1994) and it is provided by the UCI Machine Learning Repository [1]. We use it as a maximization problem in dimension eight.

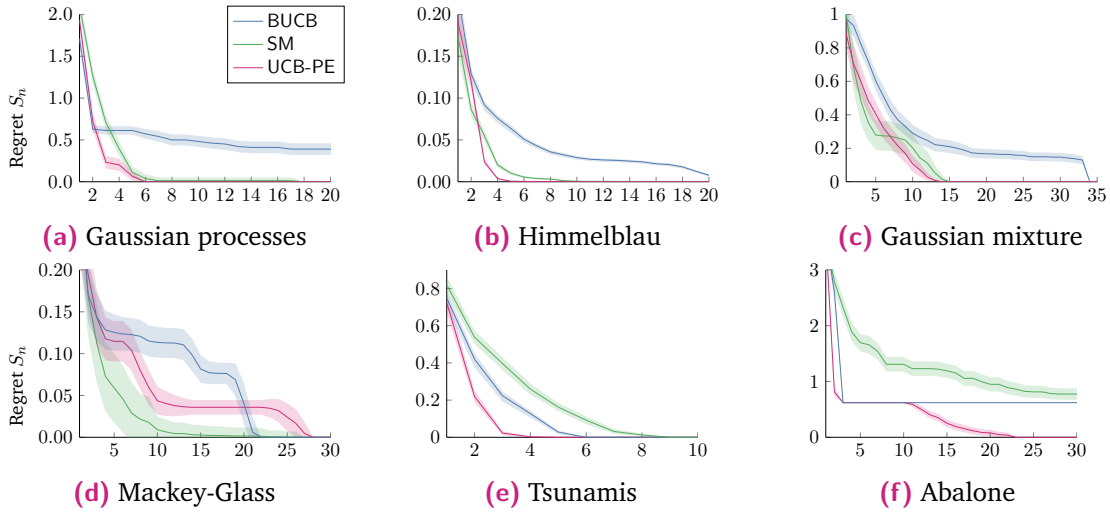


Figure 3.3. – Experiments on several real and synthetics tasks. The curves show the decay of the mean of the simple regret S_n with respect to the iteration n, over 64 experiments, with batch size $K = 10$. We show with the translucent area the confidence intervals.

Comparison of Algorithms

The Simulation Matching algorithm described in Azimi et al. (2010), with UCB base policy, has shown similar results to GP-UCB-PE on synthetic functions (Figures 3.3a, 3.3b, 3.3c) and even better results on chaotic problem without noise (Figure 3.3d), but performs worse on real noisy data (Figures 3.3e, 3.3f). On the contrary, the initialization phase of GP-BUCB leads to good regret on difficult real tasks (Figure 3.3e), but shows slower convergence on synthetic Gaussian or polynomial ones (Figures 3.3a, 3.3b, 3.3c). The number of dimensions of the Abalone task is already a limitation for GP-BUCB, making the initialization phase time-consuming. The mean regret for GP-BUCB converges to zero abruptly after the initialization phase at iteration 55, and is therefore not visible on Figure 3.3f, as for 3.3c where its regret decays at iteration 34. GP-UCB-PE achieves good performances on both sides. We obtained better regret on synthetic data as well as on real problems from the domains of physics and biology. Moreover, the computation time of SM-UCB was two order of magnitude longer than the others.

3.1.5 Conclusion and Discussion

We presented the GP-UCB-PE algorithm which addresses the problem of finding in few iterations the maximum of an unknown arbitrary function observed via batches of K noisy evaluations. We provide theoretical bounds for the cumulative regret obtained by GP-UCB-PE in the Gaussian process setting. Through parallelization, these bounds improve the ones for the state-of-the-art of sequential Bayesian optimization by a ratio of $\sqrt{K}$, and are strictly better than the ones for GP-BUCB, a concurrent algorithm for parallel Bayesian optimization. We have compared experimentally our method to GP-BUCB and SM-UCB, another approach for parallel Bayesian optimization lacking of theoretical guarantees. These empirical results have confirmed the efficiency of GP-UCB-PE on several applications. The strategy of combining in the same batch some queries selected via Pure Exploration is an intuitive idea that can

[1] http://archive.ics.uci.edu/ml/

be applied in many other methods. We expect for example to obtain similar results with the Maximum Expected Improvement policy (EI algorithm). Any proof of regret bound that relies on the fact that the uncertainty decreases with the exploration should be easily adapted to a paralleled extension with Pure Exploration. On the other hand, we observed in practice that the strategies which focus more on exploitation often lead to faster decrease of the regret, for example the strategy that uses K times the GP-UCB criterion with updated variance. We formulate the conjecture that the regret for this strategy is unbounded for general Gaussian processes, justifying the need for the initialization phase of GP-BUCB. However, it would be relevant to specify formally the assumptions needed by this greedy strategy to guarantee competitive performance.

3.2 Gaussian Processes in Metric Spaces

The previous algorithm suffers from the same drawbacks of the GP-UCB algorithm. The upper confidence bound, at the heart of the theoretical properties, is computed via a union bound over $\mathcal{X}$. As a result its performance decreases when $|\mathcal{X}|$ increases, making this approach unsuitable for continuous search spaces. In Srinivas et al. (2012) the authors show that when the unknown function is sufficiently smooth and $\mathcal{X}$ compact and convex it is possible to run the algorithm on a finite subset of $\mathcal{X}$ so that the discretization error is controlled. Unfortunately, it is necessary to know the distribution of supremum of the gradient of f to compute such a discretization. In usual settings, finding this number is strictly harder than finding the supremum of f which limits the practicability of such technique. In this section, we introduce a novel approach to compute upper confidence bound in an adaptive manner for arbitrary search spaces without additional smoothness assumptions.

3.2.1 Hierarchical Discretizations of the Search Space

Our strategy is to build a partitioning tree of $\mathcal{X}$ in the spirit of Section 2.2.3. In this respect, nodes at small depths in the tree form coarse discretizations, and nodes at bigger depths form finer discretizations. The obstacle here is that the Lipschitz-norm of f is a complicated random variable. When two points $x_1, x_2 \in \mathcal{X}$ are fixed it is easy to obtain tight high probabilistic bounds on $f(x_1) - f(x_2)$, as seen in Section 2.1.4. However bounds on $\sup_{x_1,x_2\in\mathcal{X}} \big(f(x_1) - f(x_2)\big)$ are not trivial. For continuous $\mathcal{X}$ it is necessary to consider the spatial correlations of the process to obtain meaningful bounds.

The Canonical Pseudo-Metric of Gaussian Processes

For a Gaussian process $f \sim \mathcal{GP}(0, k)$ on $\mathcal{X}$ and two points $x_1, x_2 \in \mathcal{X}$ the distribution of the difference in value is distributed as a Gaussian:

$$\begin{aligned} &f(x_1) - f(x_2) \sim \mathcal{N}\big(0, \ell^2(x_1, x_2)\big), \\ &\text{where } \ell^2(x_1, x_2) \triangleq k(x_1, x_1) + k(x_2, x_2) - 2k(x_1, x_2). \end{aligned} \tag{3.5}$$

Since the kernel k is non-negative definite, the function ℓ is positive and satisfies the triangle inequality, we call it the canonical pseudo-metric of the process. For all $u > 0$ we have the following high probabilistic version of Eq. 2.4:

$$\mathbb{P}\Big[f(x_1) - f(x_2) > \sqrt{2u}\ell(x_1, x_2)\Big] < e^{-u}. \tag{3.6}$$

Therefore for a finite $\mathcal{X}$ a union bound gives, with $\Delta(\mathcal{X})$ its ℓ-diameter:

$$\mathbb{P}\left[\sup_{x_1,x_2\in\mathcal{X}} f(x_1) - f(x_2) > \sqrt{2u + 4\log|\mathcal{X}|}\Delta(\mathcal{X})\right] < e^{-u}.$$

In the sequel, we are looking for upper bounds that do not involve $|\mathcal{X}|$ but a finer notion of geometrical size of $\mathcal{X}$ with respect to ℓ. We will compute successive ε-nets and exhibit links with the covering dimension.

Generic Chaining

Let $\mathcal{T} = (\mathcal{T}_h)_{h\geqslant 0}$ with parent map p be a discretization tree of $\mathcal{X}$ in the sense of Definition 2.6. We recall the successor relation $x \succ s$ when $x \in \mathcal{X}$ is a descendant of $s \in \mathcal{X}$. For all $h \in \mathbb{N}$, we introduce the notation p_h denoting the parent at depth h:

$$\forall s \in \mathcal{T}_h, \forall x \succcurlyeq s,\ p_h(x) \triangleq s.$$

The generic chaining (Talagrand, 2014) permits to obtain the following inequality, bounding the supremum of the difference of the values between a node and any of its descendant in $\mathcal{T}$:

Theorem 3.2 (Generic Chaining). *Fix any $u > 0$ and $a > 1$, and $(n_h)_{h\in\mathbb{N}}$ an increasing sequence of integers. Set $u_i \triangleq u + n_i + \log\big(i^a\zeta(a)\big)$ where ζ is the Riemann zeta function. Then for any tree such that $|\mathcal{T}_h| \leqslant e^{n_h}$ we have that,*

$$\forall h \geqslant 0, \forall s \in \mathcal{T}_h,\ \sup_{x\succ s} f(x) - f(s) \leqslant \omega(s),$$

holds with probability at least $1 - e^{-u}$, where for $s \in \mathcal{T}_h$,

$$\omega(s) \equiv \omega(s, \mathcal{X}, \mathcal{T}, \ell) \triangleq \sup_{x\succ s} \sum_{i>h} \sqrt{2u_i}\ell\big(p_i(x), p_{i-1}(x)\big).$$

Proof. For any $s \in \mathcal{T}_h$ and any $x \succ s$, we have $p_h(x) = s$ and $\lim_{i\to\infty} p_i(x) = x$, therefore the following sum collapses:

$$f(x) - f(s) = \sum_{i>j}\Big(f\big(p_i(x)\big) - f\big(p_{i-1}(x)\big)\Big). \tag{3.7}$$

By the properties of ℓ, we have:

$$\mathbb{P}\Big[f\big(p_i(x)\big) - f\big(p_{i-1}(x)\big) \geqslant \sqrt{2u_i}\ell\big(p_i(x), p_{i-1}(x)\big)\Big] < e^{-u_i}.$$

Algorithm 9: DiscretizationTree$(\mathcal{X}, \ell)$

$h \leftarrow 0$
$\mathcal{T} \leftarrow \{x_0\}$ for any $x_0 \in \operatorname{argmin}_{x_0 \in \mathcal{X}} \sup_{x \in \mathcal{X}} \ell(x_0, x)$
while $\mathcal{T} \neq \mathcal{X}$ **do**
  $h \leftarrow h + 1$
  $\epsilon_h \leftarrow 2^{-h-1}\Delta(\mathcal{X})$
  $\mathcal{T}_h \leftarrow \mathsf{Cover}\Big(\epsilon_h, \mathcal{X} \backslash \bigcup_{t \in \mathcal{T}} \mathcal{B}(t, \epsilon_h)\Big)$
  $\forall t \in \mathcal{T}_h,\ p(t) \leftarrow \operatorname{argmin}_{s \in \mathcal{T}} \ell(t, s)$
  $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{T}_h$
end
return $\mathcal{T}, p$

Thanks to the fact that the process f is assumed to be separable, we can consider that it exists h_0 such that $\mathcal{T}_{\leqslant h_0} = \mathcal{X}$. Now using the tree structure, the number of pairs $\big(p_i(\cdot), p_{i-1}(\cdot)\big)$ is upper bounded by $|\mathcal{T}_i|$:

$$\Big|\Big\{\big(p_i(x), p_{i-1}(x)\big) : x \in \mathcal{X}\Big\}\Big| \leqslant e^{n_i}.$$

By a union bound on the pairs we obtain:

$$\mathbb{P}\Big[\exists x \in \mathcal{X},\ f\big(p_i(x)\big) - f\big(p_{i-1}(x)\big) \geqslant \sqrt{2u_i}\ell\big(p_i(x), p_{i-1}(x)\big)\Big] < e^{n_i}e^{-u_i}.$$

With a second union bound over $i > 0$, if we denote by E^c the following event:

$$E^c \triangleq \Big\{\exists i > 0, \exists x \in \mathcal{X},\ f\big(p_i(x)\big) - f\big(p_{i-1}(x)\big) > \sqrt{2u_i}\ell\big(p_i(x), p_{i-1}(x)\big)\Big\},$$

we have $\mathbb{P}\big[E^c\big] < \sum_{i>0} e^{n_i - u_i}$. By setting $u_i = u + n_i + \log\big(i^a\zeta(a)\big)$ for $a > 1$ this simplifies to $\mathbb{P}\big[E^c\big] < e^{-u}$, that is,

$$\mathbb{P}\left[\exists x \in \mathcal{X},\ \sum_{i>h}\Big(f\big(p_i(x)\big) - f\big(p_{i-1}(x)\big)\Big) > \sum_{i>h}\sqrt{2u_i}\ell\big(p_i(x), p_{i-1}(x)\big)\right] < e^{-u}.$$

□

Theorem 3.2 can be read in terms of discretization error of $\mathcal{T}_h$. First, let us write,

$$\omega_h \triangleq \max_{s \in \mathcal{T}_h} \omega(s).$$

For $h \in \mathbb{N}$, the map p_h associates for any $x \in \mathcal{X}$ a point in $\mathcal{T}_h$ and we have with probability at least $1 - e^{-u}$:

$$f(x) - f\big(p_h(x)\big) \leqslant \omega_h.$$

Geometric Interpretation and Classical Chaining

The previous inequality suggests that to obtain a good upper bound on the discretization error, one should take $\mathcal{T}$ such that $\ell\big(p_i(x), p_{i-1}(x)\big)$ is as small as possible for every $i > 0$ and

$x \in \mathcal{X}$. As before we denote by $\Delta(s)$ the ℓ-radius of $\mathrm{Cell}(s)$. We extend this notation to the radius of the cell at depth i of any $x \in \mathcal{X}$:

$$\Delta_i(x) \triangleq \Delta\big(p_i(x)\big).$$

We have directly that the following property is satisfied for every $i \geqslant 1$,

$$\ell\big(p_i(x), p_{i-1}(x)\big) \leqslant \Delta_{i-1}(x).$$

Therefore Theorem 3.2 can be rewritten in terms of the radius:

$$\forall h \geqslant 0, \forall s \in \mathcal{T}_h,\ \sup_{x \succ s} f(x) - f(s) \leqslant \sup_{x \succ s} \sum_{i>h} \sqrt{2u_i}\Delta_{i-1}(x), \tag{3.8}$$

holds with probability at least $1 - e^{-u}$. In order to make this bound as small as possible, one should spread the points of $\mathcal{T}_h$ in $\mathcal{X}$ so that $\Delta_i(x)$ are uniformly small, while satisfying the requirement $|\mathcal{T}_i| \leqslant e^{n_i}$. In this view, we will use ε-nets to cover $\mathcal{X}$. Let ε_i decreases geometrically:

$$\varepsilon_i \triangleq \Delta(\mathcal{X})2^{-i}. \tag{3.9}$$

If one takes $n_i = H(\varepsilon_i)$, the metric entropy of $\mathcal{X}$ from Eq. 2.8, there exists a tree such that with probability at least $1 - e^{-u}$, for all $h \geqslant 0$ and $s \in \mathcal{T}_h$:

$$\sup_{x \succ s} f(x) - f(s) \leqslant \sum_{i>h} \sqrt{2u_i}\varepsilon_{i-1}, \tag{3.10}$$

where $u_i = u + H(\varepsilon_i) + \log(i^a\zeta(a))$. The tree achieving this bound can be constructed using Algorithm 9. It consists in computing a minimal ε_i-net at each depth and assigning parents to the nearest points in the upper ε_{i-1}-net, leading to $\Delta_i(x) \leqslant \varepsilon_i$ for all $i \geqslant 0$ and $x \in \mathcal{X}$. This technique is often called classical chaining (Dudley, 1967), and can be re-written in terms of the metric entropy integral:

$$\sup_{x \succ s} f(x) - f(x) \leqslant c \int_{\varepsilon=0}^{\Delta(s)} \sqrt{H(\varepsilon)}\,\mathrm{d}\varepsilon, \tag{3.11}$$

up to a constant $c \in \mathbb{R}$ thanks to the geometric decay of ε_i. We remark that the tree obtained by this method is almost a $(\ell, \delta, \frac{1}{2})$-partitioning tree of $\mathcal{X}$ in the sense of Definition 2.7 with $\delta(h) = \varepsilon_h$. However we will see later that the upper bound from Eq. 3.8 is tight but not the one from Eq. 3.10, as for instance with a Gaussian process indexed by an ellipsoid. In general, computing a minimal ε-net is NP-complete. The greedy heuristic from Algorithm 10 exhibits an approximation ratio of $\max_{x \in \mathcal{X}} \sqrt{\log\log|\mathcal{B}(x, \epsilon)|}$ for finite $\mathcal{X}$, as discussed in Section 3.2.3. We will present in Section 3.2.4 an algorithm to compute a tree in quadratic time in $|\mathcal{X}|$ leading to both a lower and upper bound on $\sup_{x \succ s} f(x) - f(s)$.

Bounding with the Covering Dimension

The upper bound from Eq. 3.10 is particularly convenient when we know a bound on $\dim(\mathcal{X}, \ell)$ the covering dimension (Definition 2.1), as stated in the following theorem.

Theorem 3.3 (Classical Chaining with Covering Dimension)**.** *For all $d > \dim(\mathcal{X}, \ell)$ and $d = \dim(\mathcal{X}, \ell)$ if the dimension is attained, with probability at least $1 - e^{-u}$:*

$$\forall h \geqslant 0, \forall s \in \mathcal{T}_h, \ \sup_{x > s} f(x) - f(s) = \mathcal{O}\Big(\sqrt{u + dh} 2^{-h}\Big).$$

Proof. For all $d > \dim(\mathcal{X}, \ell)$ and $d = \dim(\mathcal{X}, \ell)$ if the dimension is attained, it exists a constant $c \in \mathbb{R}$ such that we can bound from above the metric entropy by:

$$H(\varepsilon_i) \leqslant c - d \log \varepsilon_i.$$

We obtain $u_i = u + 2c - 2d \log \varepsilon_i + \log\big(i^a \zeta(a)\big)$. With $\varepsilon_i = \Delta(\mathcal{X}) 2^{-i}$, this leads to:

$$u_i = \mathcal{O}\big(u + di\big).$$

With Eq. 3.10, knowing that $\sum_{i=h}^{\infty} \sqrt{i} 2^{-i} = \mathcal{O}\big(\sqrt{h} 2^{-h}\big)$, we get:

$$\omega_h = \mathcal{O}\Big(\sqrt{u + dh} 2^{-h}\Big).$$

□

Theorem 3.3 shows the difference between computing a discretization for a Lipschitz function and for a Gaussian process. For a function with bounded Lipschitz norm with respect to ℓ, there exists a discretization with $e^{H(2^{-h})}$ points having error $\mathcal{O}\big(2^{-h}\big)$. Here we pay an extra price of $\sqrt{u + dh}$ because the process is stochastic. Note that for Gaussian processes, we have that the covering dimension attains $\dim(\mathcal{X}, \ell) = \alpha d$ as soon as $\mathcal{X} \subset [0, R]^d$ and there exists a constant $c > 0$ such that $\ell(x_1, x_2) \leqslant c \|x_1 - x_2\|_2^{1/\alpha}$ for all $x_1, x_2 \in \mathcal{X}$. This condition is fulfilled with $\alpha = 1$ for all the kernels presented in Section 2.1.4, including the Matérn kernel with parameter $\nu > 1$. For the Matérn kernel with parameter $\nu = 1/2$ (Ornstein-Uhlenbeck process), we have $\ell(x_1, x_2) \leqslant \|x_1 - x_2\|_2^{1/2}$, leading to $\dim(\mathcal{X}, \ell) = 2d$.

3.2.2 Regret Bounds for Bandit Algorithms

Now we have a tool to discretize $\mathcal{X}$ at a certain accuracy, we show here how to adapt the UCB strategy on the metric space $(\mathcal{X}, \ell)$. Our first idea is to compute the upper confidence bound directly using the chaining method. This is a joint work with Cédric Malherbe and has been published in Contal et al. (2015). Since the posterior process given n observations is a Gaussian process $\mathcal{GP}(\mu_n, k_n)$, with mean μ_n and kernel k_n as defined in Eq. 2.14 and Eq. 2.15, we can perform the above steps for the canonical pseudo-distance ℓ_n of this posterior process to get the UCB at each iteration. The second method we present lightens the computational burden by greedily growing the discretization tree along the iterations.

UCB Computed with Classical Chaining

In this section, we adapt the chaining technique to get upper bounds on:

$$\sup_{x^\star \in \mathcal{X}} f(x^\star) - f(x),$$

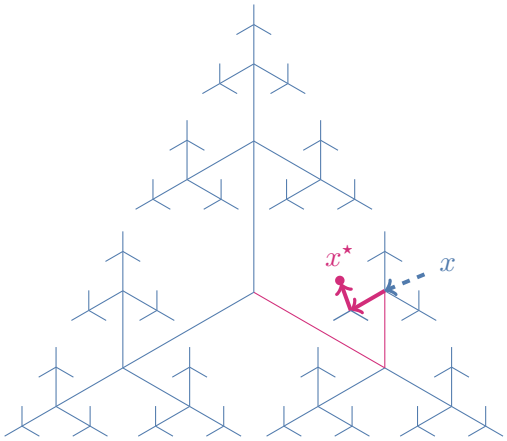


Figure 3.4. – Illustration of the path $\pi_i(x, x^\star)$. The red edges connect the branch $\{p_i(x^\star)\}_{i\leqslant 4}$ and the thick arrows show the points $\{\pi_i(x, x^\star)\}_{i\leqslant 4}$. Note that the positions of the nodes are arbitrary and do not form an ε-net with respect to the Euclidean distance.

for every $x \in \mathcal{X}$, given n observations. Let ε_h be such as Eq. 3.9 and $\mathcal{T}$ be a discretization tree of $\mathcal{X}$ with $\Delta_h(x) \leqslant \varepsilon_h$ for all $h \geqslant 0$ and $x \in \mathcal{X}$. We first define for all $x, x^\star \in \mathcal{X}$ a sequence $\pi_h(x, x^\star) \equiv \pi_{h,n,\mathcal{T}}(x, x^\star)$ of mappings from x to $\mathcal{T}_h$, converging to $x^\star$:

$$\pi_h(x, x^\star) \triangleq \begin{cases} p_h(x^\star) & \text{if } \varepsilon_h < \ell_n(x, x^\star), \\ x & \text{otherwise.} \end{cases}$$

The sequence $\pi_h(x, x^\star)$ is illustrated on Figure 3.4. We directly obtain that:

$$\begin{aligned} \pi_0(x, x^\star) &= x, \\ \pi_i(x, x^\star) &\to_{i\to\infty} x^\star. \end{aligned}$$

Therefore, we can replace $p_i(\cdot)$ by $\pi_i(x, \cdot)$ in Equation 3.7, and the rest of the proof of Theorem 3.2 still holds, conditionally on the observations, for the centered posterior process $f(\cdot) - \mu_n(\cdot)$. Since $\pi_h(x, x^\star) = x$ while $\varepsilon_h \geqslant \ell_n(x, x^\star)$, we can rewrite Equation 3.10 with a truncated sum:

$$\sup_{x^\star\in\mathcal{X}} \Big\{ f(x^\star) - f(x) - \big(\mu_n(x^\star) - \mu_n(x)\big) \Big\} \leqslant \sup_{x^\star\in\mathcal{X}} \sum_{h:\varepsilon_h<\ell_n(x,x^\star)} \sqrt{2u_h}\varepsilon_{h-1}.$$

We now decompose the indexes of the sum in two sets and take the supremum on both:

$$\sup_{x^\star\in\mathcal{X}} \sum_{h:\varepsilon_h<\ell_n(x,x^\star)} \sqrt{2u_h}\varepsilon_{h-1} \leqslant \sup_{x^\star\in\mathcal{X}} \sum_{h:\varepsilon_h<\sigma_n(x^\star)} \sqrt{2u_h}\varepsilon_{h-1} + \sup_{x^\star\in\mathcal{X}} \sum_{h:\sigma_n(x^\star)\leqslant\varepsilon_h<\ell_n(x,x^\star)} \sqrt{2u_h}\varepsilon_{h-1}.$$

This suggests the following query:

$$x_{n+1} \in \operatorname*{argmax}_{x\in\mathcal{X}} \left\{ \mu_n(x) + \sum_{h:\varepsilon_h<\sigma_n(x)} \sqrt{2u_h}\varepsilon_{h-1} \right\}. \tag{3.12}$$

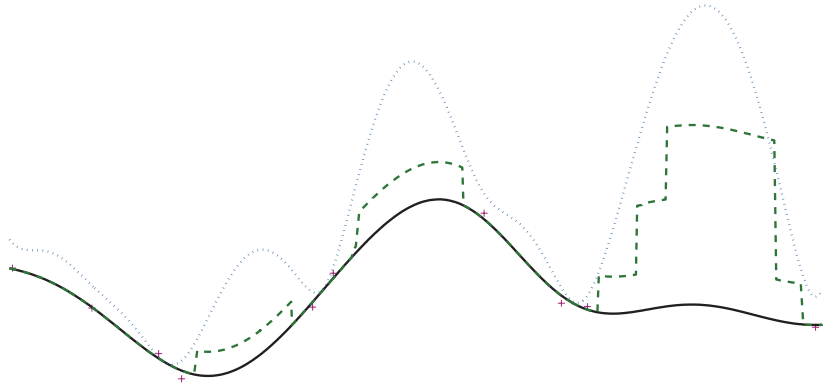


Figure 3.5. – Illustration of the UCB used in Eq. 3.12. The plain black line is the target function f. The red crosses are the noisy observations. The dashed green line is the UCB used by the Chaining-UCB algorithm. The rectangular form of the UCB is explained by the discrete sum. The next query selected by the algorithm is the maximum of the UCB. The dotted blue line is the target used by the GP-UCB algorithm.

The principle of this rule is compared intuitively to the classical UCB rule with a union bound on Figure 3.5, where the union bound in the UCB rule is calibrated for $|\mathcal{X}| = 10^4$ points. By using the discrete sum instead of the integral from Eq. 3.11, we limit the number of ε-nets which need to be computed to only the ε_h satisfying $\varepsilon_h > \min_{x\in\mathcal{X}} \sigma_n(x)$. Indeed after this level the summands are shared for all x and can be removed for the maximization in Eq. 3.12. Thanks to the geometrical decay of ε_h, the number of levels we need to compute remains low in practice. This choice of query leads to, with probability at least $1 - e^{-u}$:

$$\sup_{x^\star\in\mathcal{X}} f(x^\star) - f(x_{n+1}) \leqslant \sum_{h:\varepsilon_h<\sigma_n(x_{n+1})} \sqrt{2u_h}\varepsilon_{h-1} + \sup_{x^\star\in\mathcal{X}} \sum_{\substack{h:\sigma_n(x^\star)\leqslant\varepsilon_h \\ \varepsilon_h<\ell_n(x_{n+1},x^\star)}} \sqrt{2u_h}\varepsilon_{h-1}.$$

Using that $\ell_n(x, x^\star) \leqslant \sigma_n(x) + \sigma_n(x^\star)$ and the geometric decay of ε_h, we know by elementary calculations that the right sum is smaller than twice the first sum (see details in Contal et al. (2015)). We get:

$$\sup_{x^\star\in\mathcal{X}} f(x^\star) - f(x_{n+1}) \leqslant 3 \sum_{h:\varepsilon_h<\sigma_n(x_{n+1})} \sqrt{2u_h}\varepsilon_{h-1}.$$

Finally, we can bound from above the metric entropy of $(\mathcal{X}, \ell_n)$ appearing in u_h by the metric entropy of $(\mathcal{X}, \ell)$, since $\ell_n \leqslant \ell$. Therefore, when the covering dimension of $(\mathcal{X}, \ell)$ is finite we can translate results from Theorem 3.3 to obtain regret bound for this method. Precisely, for all $d > \dim(\mathcal{X}, \ell)$ and $d = \dim(\mathcal{X}, \ell)$ when the dimension is attained, denoting $s_n \triangleq \sigma_{n-1}(x_n)$,

$$\sup_{x^\star\in\mathcal{X}} f(x^\star) - f(x_n) \lesssim \sqrt{d}\big(s_n - s_n \log s_n\big),$$

where we used that for all $s \in (0, 1)$, $\sum_{h:2^{-h}<s} 2^{-h}\sqrt{\log h} < s - s\log s$ and $\sum_{h:2^{-h}<s} 2^{-h} < 2s$, and the $\lesssim$ notation hides universal constants. Concrete regret bounds are obtainable by bounding from above $\sum_{i=1}^n (s_n - s_n \log s_n)$ for particular kernel. As an example for squared

exponential kernel we have seen that $\sum_{i=1}^{n} s_n^2 \leqslant \mathcal{O}(\log^{d+1} n)$, thus by maximizing the previous sum under this constraint with Lagrange multipliers,

$$R_n \leqslant \mathcal{O}\Big(\sqrt{n \log^{d+2} n}\Big).$$

UCB on Adaptive Discretizations

The major drawback of the above approach is that it requires to compute nested ε-nets at each iteration of the optimization procedure. Even if we propose efficient way to compute almost optimal ε-nets, the computational cost may be prohibitive in some cases. In this section, we introduce another way to build a sound UCB rule with chaining. Here we consider a single tree $\mathcal{T}$ adapted for ℓ only (instead of ℓ_n), which will be built greedily. At each iteration n, we select a depth $h(n) \in \mathbb{N}$ and perform the UCB algorithm on $\mathcal{T}_{h(n)}$, that is for $a > 1$, $u > 0$ and $|\mathcal{T}_h| \leqslant e^{n_h}$ for all $h \geqslant 0$:

$$\begin{aligned} x_{n+1} &\in \operatorname*{argmax}_{x \in \mathcal{T}_{h(n)}} U_n(x), \\ \text{where } U_n(x) &\triangleq \mu_n(x) + \sqrt{2u_n \sigma_n^2(x)}, \\ \text{and } u_n &\triangleq u + n_{h(n)} + \log\left(n^a \zeta(a)\right). \end{aligned}$$

The error of the obtained algorithm can be decomposed in two terms, the discretization error and the optimization error. This is rendered explicit in the following theorem.

Theorem 3.4 (Regret Bound on Metric Spaces)**.** *Fix any $u > 0$ and $a > 1$. For any discretization tree such that $|\mathcal{T}_h| \leqslant e^{n_h}$ and any sequence $h(n) \in \mathbb{N}$, when $k(\cdot,\cdot) \leqslant 1$, the UCB algorithm described above has a regret upper bounded with probability at least $1 - 2e^{-u}$ by:*

$$R_n \leqslant 2\sqrt{2c_\eta u_n n \gamma_n} + \sum_{i=1}^{n} \omega_{h(i)},$$

where $c_\eta \triangleq \frac{2}{\log(1+\eta^{-2})}$, η^2 is the variance of the noise, and γ_n is the maximum information gain of f attainable by n queries (Eq. 2.33).

Proof. Like previously we have that the values of f at $\mathcal{T}_{h(n)}$ will lie in the upper and lower confidence bounds with high probability:

$$\mathbb{P}\Big[\exists n \geqslant 1, \exists x \in \mathcal{T}_{h(n)},\ |f(x) - \mu_n(x)| > \sqrt{2u_n \sigma_n^2(x)}\Big] < e^{-u}.$$

Using Theorem 3.2, we have that,

$$\forall h \geqslant 0,\ \sup_{x^\star \in \mathcal{X}} f(x^\star) \leqslant \omega_h + \sup_{x^\star \in \mathcal{X}} f\big(p_h(x^\star)\big),$$

holds with probability at least $1 - e^{-u}$. Since $p_{h(n)}(x^\star) \in \mathcal{T}_{h(n)}$ for all $x^\star \in \mathcal{X}$, we can combine both inequalities and obtain for all $x \in \mathcal{T}_{h(n)}$:

$$\forall n \geqslant 1,\ \sup_{x^\star \in \mathcal{X}} f(x^\star) - f(x) \leqslant \omega_{h(n)} + \sup_{x^\star \in \mathcal{T}_{h(n)}} \Big(\mu_n(x^\star) + \sqrt{2u_n}\sigma_n(x^\star)\Big) - \mu_n(x) + \sqrt{2u_n}\sigma_n(x),$$

Algorithm 10: GreedyCover $(\varepsilon, \mathcal{X})$

$T \leftarrow \varnothing$
while $\mathcal{X} \neq \varnothing$ **do**
  $x \leftarrow \operatorname{argmax}_{x \in \mathcal{X}} \left|\left\{x' \in \mathcal{B}(x, \varepsilon)\right\}\right|$
  $T \leftarrow T \cup \{x\}$
  $\mathcal{X} \leftarrow \mathcal{X} \backslash \mathcal{B}(x, \varepsilon)$
end
return T

holds with probability at least $1 - 2e^{-u}$. Thanks to our choice for x_{n+1}, we have:

$$\mathbb{P}\left[\forall n \geqslant 1,\ \sup_{x^\star \in \mathcal{T}} f(x^\star) - f(x_{n+1}) \leqslant \omega_{h(n)} + 2\sqrt{2u_n}\sigma_n(x_{n+1})\right] \geqslant 1 - 2e^{-u}.$$

Summing over n and taking the same steps as in Theorem 3.1, we obtain, with probability at least $1 - 2e^{-u}$ that,

$$R_n \leqslant 2\sqrt{2c_\eta u_n n \gamma_n} + \sum_{i=1}^{n} \omega_{h(i)}.$$

□

Choice of the Discretization Depth

Since the first part of the previous upper bound on the regret is of order at least $\sqrt{u_n n} = \mathcal{O}(\sqrt{n \log n})$, we choose to select $h(i)$ such that $\sum_{i=1}^{n} \omega_{h(i)} \leqslant \mathcal{O}(\sqrt{n \log n})$. The general rule to attain this order is to select, with c a constant:

$$h(i) = \min\left\{i \in \mathbb{N} : \omega_i \leqslant c\sqrt{\frac{\log i}{i}}\right\},$$

since $\sum_{i=1}^{n} \sqrt{\frac{\log i}{i}} \leqslant 2\sqrt{n \log n}$. That is, when the covering dimension is finite, it suffices to take:

$$h(i) = \left\lceil \tfrac{1}{2} \log_2 i \right\rceil,$$

to obtain, in the light of Theorem 3.3 with the tree computed by Algorithm 9,

$$\omega_{h(i)} \leqslant \mathcal{O}\left(\sqrt{\frac{u + d \log i}{i}}\right),$$

leading to $\sum_{i=1}^{n} \omega_{h(i)} \leqslant \mathcal{O}\big(\sqrt{(u+d)n \log n}\big)$. Finally, plugging this last inequality in Theorem 3.4, we have with probability at least $1 - 2e^{-u}$:

$$R_n \leqslant \mathcal{O}\left(\sqrt{c_\eta (u + d \log n) n \gamma_n \log^{a+1} n}\right).$$

As seen before in Section 2.2.4, the informational quantity γ_n may be further bounded for usual kernels.

3.2.3 Efficient Algorithms

The optimization algorithms described above require to build a discretization tree with Algorithm 9, in order to compute the metric entropy $H(\varepsilon_i)$ for geometrically decreasing ε_i. In this section, we present an efficient heuristic for the optimal covering problem and prove approximation ratio, that is the number of points in the net obtained by the heuristic divided by the optimal one. We see also that it is not necessary to compute the entire discretization tree: it suffices to stop at the required level $h(n)$. Finally, we provide tools to compute ε-nets on a compact $\mathcal{X}$ with theoretical guarantees.

Greedy Cover

The exact computation of an optimal ε-net is NP-hard. We show here how to build in practice a near-optimal ε-net using a greedy algorithm on a graph. First, remark that for any fixed ε we can define a graph $\mathcal{G}$ where the nodes are the elements of $\mathcal{X}$ and there is an edge between x_1 and x_2 if and only if $\ell(x_1, x_2) \leqslant \varepsilon$. The size of this construction is $\mathcal{O}(|\mathcal{X}|^2)$, but the sparse structure of the underlying graph can be exploited to get an efficient representation. The problem of finding an optimal ε-net reduces to the problem of finding a minimal dominating set on $\mathcal{G}$. We can therefore use the greedy Algorithm 10 which enjoys an approximation ratio of $\log d_{\max}(\mathcal{G})$, where $d_{\max}(\mathcal{G})$ is the maximum degree of $\mathcal{G}$, which is equal to $\max_{x\in\mathcal{X}}|\mathcal{B}(x,\varepsilon)|$. An interested reader may see for example Johnson (1973) for a proof of NP-hardness and approximation results. This construction leads to an additional (almost constant) term of $\max_{x\in\mathcal{X}}\sqrt{\log\log|\mathcal{B}(x,\varepsilon)|}$ in Theorem 3.3. Finally, note that this approximation is optimal unless P = NP as shown in Raz and Safra (1997).

UCB on Greedily Grown Tree

Combining the previous remarks we come up with Algorithm 11, a modification of the GP-UCB algorithm for arbitrary metric spaces using a greedily grown discretization tree. The tree is extended at logarithmic frequency, which reduces drastically the number of calls of GreedyCover, and form therefore a tractable algorithm. In the light of the above inequalities, we have the following result:

Theorem 3.5 (Regret for UCB on a Greedily Grown Tree)**.** *Fix any $u > 0$ and $a > 1$. Let $d > \dim(\mathcal{X}, \ell)$ or $d = \dim(\mathcal{X}, \ell)$ if the dimension is attained. When $k(\cdot,\cdot) \leqslant 1$, the cumulative regret of Algorithm 11 in upper bounded with probability at least $1 - 2e^{-u}$ by:*

$$R_n \lesssim \sqrt{c_\eta(u + d\log n)n\gamma_n \log^{a+1} n},$$

where $c_\eta \triangleq \frac{2}{\log(1+\eta^{-2})}$, η^2 is the variance of the noise, and the $\lesssim$ notation removes universal constants and $\log\log$ factors.

Computations on Non-Finite Compact Spaces

Even if all the theoretical analysis of this work assumes that $\mathcal{X}$ is finite for measurability reasons, it is not satisfactory from a numerical point of view. We show here that if the search space $\mathcal{X}$ is a compact, then there is a way to reduce computations to the finite case. First note that is $(\mathcal{X}, \ell)$ is compact, then there exists a uniform distribution $\mathcal{U}$ on $\mathcal{X}$ with respect to ℓ. We also point out that when the kernel is isotropic, then the uniform distribution for the

Algorithm 11: GP-UCB on Greedily Grown Tree (k, η, u, a)

$h_0 \leftarrow -1$
$\mathcal{T} \leftarrow \varnothing$
for $n = 0, 1, \ldots$ **do**
  $h_n \leftarrow \lceil 1/2 \log_2 n \rceil$
  if $h_n \neq h_{n-1}$ **then**
    $\varepsilon_h \leftarrow 2^{-h_n - 1} \Delta(\mathcal{X})$
    $T_h \leftarrow \mathsf{GreedyCover}\big(\varepsilon_h, \mathcal{X} \backslash \bigcup_{x \in \mathcal{T}} \mathcal{B}(x, \varepsilon_h)\big)$
    $\forall t \in T_h,\ p(t) \leftarrow \operatorname{argmin}_{s \in \mathcal{T}} \ell(t, s)$
    $\mathcal{T} \leftarrow \mathcal{T} \cup T_h$
  end
  Compute μ_n and σ_n^2 (Eq. 2.14, 2.16)
  $u_n \leftarrow u + \log|\mathcal{T}| + \log\left(n^a \zeta(a)\right)$
  for $x \in T$ **do**
    $U_n(x) \leftarrow \mu_n(x) + \sqrt{2 u_n \sigma_n^2(x)}$
  end
  $x_{n+1} \leftarrow \operatorname{argmax}_{x \in T_h} U_n(x)$
  $y_{n+1} \leftarrow$ **Query**$\big(x_{n+1}\big)$
end
return T

norm of $\mathcal{X}$ corresponds to the uniform distribution for $(\mathcal{X}, \ell)$. The following lemma describes the probability to get an ε-net via uniform sampling in $\mathcal{X}$.

Lemma 3.7 (Covering with Uniform Sampling)**.** *Let $\varepsilon > 0$, $\mathcal{U}$ be the uniform distribution on $(\mathcal{X}, \ell)$, $m \triangleq N(\varepsilon)$ the covering numbers of $\mathcal{X}$ (Eq. 2.7), and $X_n = (x_1, \ldots, x_n)$ be n points distributed independently according to $\mathcal{U}$ with $n \geqslant m(\log m + u)$. Then with probability at least $1 - e^{-u}$, X_n is a 2ε-net of $\mathcal{X}$.*

Proof. Let T be an ε-net on $\mathcal{X}$ of cardinality $|T| = m$. Then the probability P^c that it exists $t \in T$ such that $\min_{i \leqslant n} \ell(t, x_i) > \varepsilon$ is less than:

$$P^c \leqslant \sum_{t \in T} \mathbb{P}\Big[\forall i \leqslant n,\ x_i \notin \mathcal{B}(t, \varepsilon)\Big].$$

Since $\mathcal{U}$ attributes an equal probability mass for every ball of radius ε, $P^c \leqslant m\Big(\frac{m-1}{m}\Big)^n$. With $\log \frac{m}{m-1} \geqslant \frac{1}{m}$, we have for $n \geqslant m(\log m + u)$ that,

$$P^c \leqslant e^{-u}.$$

By the triangle inequality, with probability at least $1 - e^{-u}$, X_n is 2ε-net. □

Therefore when we want to compute an ε-net on a compact $\mathcal{X}$, an efficient way is to first sample $X_n = (x_1, \ldots, x_n)$ uniformly with $n \geqslant m(\log m + u)$ and $m = N(\frac{1}{4}\varepsilon)$, which gives a $\frac{1}{2}\varepsilon$-net with probability at least $1 - e^{-u}$. Then running $\mathsf{GreedyCover}\ \big(\varepsilon/2, X_n\big)$ outputs an ε-net of $\mathcal{X}$ with probability at least $1 - e^{-u}$.

3.2.4 Tightness Results on Discretization Trees

We present in this section a strong result on a tree $\mathcal{T}$ obtainable by a tractable algorithm such that both upper and lower bounds on the discretization error are available. We show that a converse of Theorem 3.2 is true with high probability, on arbitrary kernel k.

A High Probabilistic Lower Bound on the Supremum

We first recall that for all tree $\mathcal{T}$ such that $|\mathcal{T}_h| \leqslant e^{n_h}$ and $u > 0$, we have:

$$\forall h \geqslant 0, \forall s \in \mathcal{T}_h,\ \sup_{x \succ s} f(x) - f(s) \leqslant \widetilde{\mathcal{O}}\Big(\sup_{x \succ s} \sum_{i > h} \Delta_i(x)\sqrt{u + n_i}\Big),$$

with probability at least $1 - e^{-u}$, where the $\widetilde{\mathcal{O}}$ notation hides the logarithmic factors. For the sequel, we will fix for n_i a geometric sequence $n_i = 2^i$ for all $i \geqslant 1$. Therefore we have the following upper bound.

Corollary 3.1 (Generic Chaining with Doubly Exponential Growth)**.** *Fix any $u > 0$ and let $\mathcal{T}$ such that $|\mathcal{T}_h| \leqslant e^{2^h}$. Then,*

$$\sup_{x \succ s} f(x) - f(s) \leqslant \widetilde{\mathcal{O}}\Big(\sup_{x \succ s} \sum_{i > h} 2^{\frac{i}{2}} \Delta_i(x)\Big),$$

holds for all $h \geqslant 0$ and $s \in \mathcal{T}_h$ with probability at least $1 - e^{-u}$.

To show the tightness of this result, we prove in Section 3.2.5 that for a tree constructed with Algorithm 12, the following probabilistic bound also holds.

Theorem 3.6 (Generic Chaining Lower Bound)**.** *Fix any $u > 0$ and let $\mathcal{T}$ be constructed as in Algorithm 12. Then,*

$$\sup_{x \succ s} f(x) - f(s) \geqslant \widetilde{\mathcal{O}}\Big(\sup_{x \succ s} \sum_{i > h} 2^{\frac{i}{2}} \Delta_i(x)\Big),$$

holds for all $h \geqslant 0$ and $s \in \mathcal{T}_h$ with probability at least $1 - e^{-u}$.

The benefit of this lower bound is huge for theoretical and practical reasons. It first says that we cannot discretize $\mathcal{X}$ in a finer way that Algorithm 12 up to a constant multiplicative factor. This also means that even if the search space $\mathcal{X}$ is "smaller" than what is suggested using the metric entropy and Theorem 3.3, then Algorithm 12 finds the correct upper bound. Up to our knowledge, this result is the first construction of a tree leading to both an upper and a lower bounds at every depth with high probability. The proof of this theorem shares some similarity with the construction to obtain lower bound in expectation, see for example Talagrand (2014) or Ding et al. (2011) for a tractable algorithm.

Pruning the Discretization Tree to Obtain a Balanced Tree

Algorithm 12 proceeds as follows. It first computes $(\mathcal{T}_h)_{h \geqslant 0}$ a succession of ε_h-nets as in the DiscretizationTree algorithm with the GreedyCover procedure. with $\varepsilon_h = \Delta(\mathcal{X})2^{-h}$. Like before, the parent of a node is set to the closest node in the upper level,

$$\forall t \in \mathcal{T}_h,\ p(t) = \operatorname*{argmin}_{s \in \mathcal{T}_{h-1}} \ell(t, s),$$

that is the cells of the discretization are the Voronoï cells for ℓ. Therefore we have $\ell\big(t, p(t)\big) \leqslant \varepsilon_{h-1}$ for all $t \in \mathcal{T}_h$. Moreover, by looking at how the ε_h-net is computed in the GreedyCover procedure, we also have $\ell(t_i, t_j) \geqslant \varepsilon_h$ for all $t_i, t_j \in \mathcal{T}_h$. These two properties are crucial for the proof of the lower bound.

Then, the algorithm updates the tree to make it well balanced, that is such that no node $t \in \mathcal{T}_h$ has more that $e^{n_{h+1}-n_h} = e^{2^h}$ children. We note at this time that this condition will be already satisfied in every reasonable space, so that the complex procedure that follows is only required in extreme cases. To force this condition, Algorithm 12 starts from the leaves and "prunes" the branches if they outnumber e^{2^h}. We remark that this backward step is not present in the literature on generic chaining, and is needed for our objective of a lower bound with high probability. By doing so, this creates a node called a *pruned node* which will take as children the pruned branches. For this construction to be tight, the pruning step has to be handled with care. Algorithm 12 attaches to every pruned node a value, computed using the values of its children, making explicit the backward strategy. When pruning branches, the algorithm keeps the e^{2^h} nodes with maximum values and moves the others. The intuition behind this strategy is to avoid pruning branches that already contain pruned nodes.

Finally, note that this pruning step may create unbalanced pruned nodes when the number of nodes at depth h is way larger that e^{2^h}. When this is the case, Algorithm 12 restarts the pruning with the updated tree to recompute the values. Thanks to the doubly exponential growth in the balance condition, this cannot occur more that $\log\log|\mathcal{X}|$ times and the total complexity is $\mathcal{O}\big(|\mathcal{X}|^2\big)$ up to $\log\log$ factors.

Computing the Pruning Values and Anti-Concentration Inequalities

We end this section by describing the values used for the pruning step. We need a function φ satisfying the following anti-concentration inequality. For all $m \in \mathbb{N}$, let $s \in \mathcal{X}$ and $t_1, \dots, t_m \in \mathcal{X}$ such that $\forall i \leqslant m,\ p(t_i) = s$ and $\ell(s, t_i) \leqslant \Delta$, and finally $\ell(t_i, t_j) \geqslant \alpha$. Then φ is such that:

$$\mathbb{P}\Big[\max_{i \leqslant m} f(t_i) - f(s) \geqslant \varphi(\alpha, \Delta, m, u)\Big] > 1 - e^{-u}. \tag{3.13}$$

A function φ satisfying this hypothesis is described in Lemma 3.10. Then the value $V_h(s)$ of a node $s \in \mathcal{T}_h$ is computed as:

$$V_h(s) \triangleq \sup_{x > s} \sum_{i > h} \varphi\Big(\tfrac{1}{2}\Delta_h(x), \Delta_h(x), m, u\Big) \mathbb{1}\big\{p_i(x) \text{ is a pruned node}\big\}.$$

The two steps in Section 3.2.5 proving Theorem 3.6 are: first, show that $\sup_{x>s} f(x) - f(s) \geqslant c_u V_h(s)$ for $c_u > 0$ with probability at least $1 - e^{-u}$, second, show that $V_h(s) \geqslant c'_u \sup_{x>s} \sum_{i>h} \Delta_i(x) 2^{\frac{i}{2}}$ for $c'_u > 0$.

Gaussian Processes Indexed by Ellipsoids

As mentioned in Section 3.2.1, the classical chaining bound from Theorem 3.3 is not tight for every Gaussian process. An important example is when the search space is a (possibly infinite dimensional) ellipsoid:

$$\mathcal{X} = \left\{ x \in \ell^2 : \sum_{i \geqslant 1} \frac{x_i^2}{a_i^2} \leqslant 1 \right\}.$$

Algorithm 12: BalancedTree$(\mathcal{X}, \ell, \varphi)$

$\mathcal{T} \leftarrow$ DiscretizationTree$(\mathcal{X}, \ell)$ with GreedyCover
done $\leftarrow \bot$
while $\neg$ done **do** // Restart if needed
  done $\leftarrow \top$
  $h \leftarrow \mathrm{height}(\mathcal{T})$
  $\forall t \in \mathcal{T}_h,\ V_h(t) \leftarrow 0$ // Set value to 0 at the leafs
  while $h > 0$ **do** // Backward pruning
    for $s \in \mathcal{T}_{h-1}$ **do**
      $T_s \leftarrow \{t : p(t) = s\}$ // The children of s
      $\forall t \in T_s,\ V_h(t) \leftarrow \sup_{t':p(t')=t} V_{h+1}(t')$ // Default value
      $m \leftarrow e^{n_h - n_{h-1}}$
      if $|T_s| > m$ **then** // If the tree is not balanced
        Let $t_1, \dots, t_n \in T_s$ ordered by decreasing $V_h(t)$
        Create a pruned node t and set $p(t) \leftarrow s$
        $\forall i \geqslant m, \forall t' \, s.t. \, p(t') = t_j,\ p(t') \leftarrow t$ // Move the remaining to t
        if $\left|\{t' : p(t') = t\}\right| \leqslant e^{n_{h+1} - n_h}$ **then**
          $\Delta_h \leftarrow \sup_{x>t} \ell(x, t)$ // Update the value of the pruned node
          $u_h \leftarrow u + n_h + h \log 2$
          $V_h(t) \leftarrow \sup_{t':p(t')=t} V_{h+1}(t') + \varphi\left(\frac{1}{2}\Delta_h, \Delta_h, m, u_h\right)$
        else
          done $\leftarrow \bot$ // Cannot occur more that $\log\log|\mathcal{X}|$ times
        end
      end
    end
  end
end
return $\mathcal{T}$

where $a \in \ell^2$, and the Gaussian process is defined as:

$$\forall x \in \mathcal{X},\ f(x) \triangleq \sum_{i \geqslant 1} x_i g_i,$$

with g_i independent Gaussian random variables $\mathcal{N}(0,1)$. Here the pseudo-metric $\ell(\cdot,\cdot)$ coincides with the usual ℓ^2 metric. The study of the supremum of such processes is connected to learning error bounds for kernel machines like Support Vector Machines, as a quantity bounding the learning capacity of a class of functions in a RKHS, see for example Mendelson (2002). It can be shown by geometrical arguments that,

$$\forall \varepsilon > 0, s \in \mathcal{X},\ \mathbb{E}\left[\sup_{x \in \mathcal{B}(s,\varepsilon)} f(x) - f(s)\right] \leqslant \mathcal{O}\left(\sqrt{\sum_{i \geqslant 1} \min(a_i^2, \varepsilon^2)}\right),$$

and that this supremum exhibits χ^2-tails around its expectation, see for example Boucheron et al. (2013). This concentration is not grasped by Eq. 3.10. Indeed the previous equality is of

order $\sqrt{\sum_{i\geqslant 1} 2^i a_{2^i}^2}$ while the upper bound of Eq. 3.10 is of order $\sum_{i\geqslant 1} 2^i a_{2^i}$, see for instance Talagrand (2014). It is therefore required to leverage the construction of Algorithm 12 to get a tight estimate. The present work forms a step toward efficient and practical online model selection in such classes in the spirit of Rakhlin and Sridharan (2014) and Gaillard and Gerchinovitz (2015).

3.2.5 Proof of the Generic Chaining Lower Bound

In this section, we provide the proof Theorem 3.6 giving a high probabilistic lower bound obtained via Algorithm 12.

Probabilistic Tools for Gaussian Processes

We first prove a probabilistic bound on independent Gaussian variables and then show that a similar bound holds for f via a comparison inequality.

Lemma 3.8 (Anti-Concentration for Independent Gaussian Variables)**.** *Let $(N_i)_{i\leqslant m}$ be m independent standard normal variables. For $m \geqslant 2.6u$ we have with probability at least $1-e^{-u}$ that:*

$$\max_{i\leqslant m} N_i \geqslant \sqrt{\log \frac{m}{2.6u}}.$$

Proof. With $N_i \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$ for all $i \leqslant m$ we obtain for all $\lambda \in \mathbb{R}$:

$$\mathbb{P}\Big[\max_{i\leqslant m} N_i \geqslant \lambda\Big] = 1-\Phi(\lambda)^m,$$

where Φ is the standard normal cumulative distribution function, which satisfies $\Phi(\lambda) \leqslant 1-c_1 e^{-\lambda^2}$ with $c_1 > 0.38$, see for example Côté et al. (2012). For $\lambda \leqslant \sqrt{\log \frac{c_1}{1-e^{-\frac{u}{m}}}}$ and $u \leqslant m \log \frac{1}{1-c_1}$ we obtain $\Phi(\lambda)^m \leqslant e^{-u}$. Using that $1-e^{-x} \leqslant x$ for $x \geqslant 0$, we obtain with $u \leqslant c_1 m$ that:

$$\mathbb{P}\left[\max_{i\leqslant m} N_i \geqslant \sqrt{\log \frac{c_1 m}{u}}\right] \geqslant 1-e^{-u}.$$

□

The following lemma will be useful to derive anti-concentration inequalities for non independent Gaussian variables, provided that their L_2 distance are large enough. Similar results are well known if one replaces the probabilities by expectations, see for example Ledoux and Talagrand (1991).

Lemma 3.9 (Comparison Inequality for Gaussian Variables)**.** *Let $(X_i)_{i\leqslant m}$ and $(Y_i)_{i\leqslant m}$ be Gaussian random variables such that for all $i,j \leqslant m$, $\mathbb{E}\big[(X_i-X_j)^2\big] \geqslant \mathbb{E}\big[(Y_i-Y_j)^2\big]$ and $\mathbb{E}\big[X_i^2\big] \geqslant \mathbb{E}\big[Y_i^2\big]$. Then we have for all $\lambda \in \mathbb{R}$:*

$$\mathbb{P}\Big[\max_{i\leqslant m} X_i < \lambda - 2\sigma\Big] \leqslant \mathbb{P}\Big[\max_{i\leqslant m} Y_i < \lambda\Big],$$

where $\sigma \triangleq \max_{i\leqslant m} \mathbb{E}\big[X_i^2\big]^{1/2}$.

Proof. Let g be a Rademacher variable independent of X and Y. We introduce the following random variables:

$$\widetilde{X}_i \triangleq X_i + g\big(\sigma^2 + \mathbb{E}\big[Y_i^2\big] - \mathbb{E}\big[X_i^2\big]\big)^{1/2},$$
$$\widetilde{Y}_i \triangleq Y_i + g\sigma.$$

With this definition, we have by simple calculus that $\mathbb{E}\big[\widetilde{X}_i^2\big] = \mathbb{E}\big[Y_i^2\big] + \sigma^2 = \mathbb{E}\big[\widetilde{Y}_i^2\big]$. Furthermore, $\mathbb{E}\big[(\widetilde{Y}_i - \widetilde{Y}_j)^2\big] = \mathbb{E}\big(Y_i - Y_j)^2\big]$ and $\mathbb{E}\big[(\widetilde{X}_i - \widetilde{Y}_j)^2\big] \geqslant \mathbb{E}\big[(X_i - X_j)^2\big]$ for all i and j, that is:

$$\mathbb{E}\big[(\widetilde{X}_i - \widetilde{X}_j)^2\big] \geqslant \mathbb{E}\big[(\widetilde{Y}_i - \widetilde{Y}_j)^2\big].$$

Combining this with the previous remark we obtain $\mathbb{E}\big[\widetilde{X}_i\widetilde{X}_j\big] \leqslant \mathbb{E}\big[\widetilde{Y}_i\widetilde{Y}_j\big]$. Using Corollary 3.12 in Ledoux and Talagrand (1991) we know that for all $\lambda \in \mathbb{R}$:

$$\mathbb{P}\Big[\max_{i\leqslant m} \widetilde{X}_i \geqslant \lambda\Big] \geqslant \mathbb{P}\Big[\max_{i\leqslant m} \widetilde{Y}_i \geqslant \lambda\Big]. \tag{3.14}$$

Now it is easy to check that $\mathbb{P}\big[\max_{i\leqslant m} \widetilde{Y}_i < \lambda - \sigma\big] \leqslant \mathbb{P}\big[\max_{i\leqslant m} Y_i < \lambda\big]$ and similarly for $\widetilde{X}$ that $\mathbb{P}\big[\max_{i\leqslant m} X_i < \lambda - (\sigma^2 + \mathbb{E}\big[Y_i^2\big] - \mathbb{E}\big[X_i^2\big])^{\frac{1}{2}}\big] \leqslant \mathbb{P}\big[\max_{i\leqslant m} \widetilde{X}_i < \lambda\big]$. With Eq. 3.14 we have:

$$\mathbb{P}\Big[\max_{i\leqslant m} X_i < \lambda - \sigma - (\sigma^2 + \mathbb{E}\big[Y_i^2\big] - \mathbb{E}\big[X_i^2\big])^{\frac{1}{2}}\Big] \leqslant \mathbb{P}\Big[\max_{i\leqslant m} Y_i < \lambda\Big].$$

Using that $\mathbb{E}\big[X_i^2\big] \geqslant \mathbb{E}\big[Y_i^2\big]$ finishes the proof. □

Proof of the Lower Bound

We now use the previous lemmas to bound from below $\sup_{x>s} f(x) - f(s)$ for a node s satisfying properties of a pruned node. By doing so, we give the exact formula for the function φ in Eq. 3.13.

Lemma 3.10 (Anti-Concentration for a Pruned Node)**.** *Let $s \in \mathcal{T}_h$ and $(t_i)_{i\leqslant m}$ such that $t_1 = s$ and for all $2 \leqslant i \leqslant m$, $p(t_i) = s$ and $\ell(s, t_i) \leqslant \Delta$. If $\ell(t_i, t_j) \geqslant \alpha$ for all $i \neq j$ then the following holds with probability at least $1 - e^{-u}$ for $3u < m$:*

$$\max_{i\leqslant m} f(t_i) - f(s) \geqslant \frac{\alpha}{\sqrt{2}}\sqrt{\log\frac{m}{3u}} - 2\Delta.$$

Proof. For $i \leqslant m$, let $X_i \triangleq f(t_i) - f(s)$ and $Y_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \frac{\alpha^2}{2})$ be independent Gaussian variables. We have $\mathbb{E}\big[(X_i - X_j)^2\big] = \ell(t_i, t_j)^2 \geqslant \alpha^2 = \mathbb{E}\big[(Y_i - Y_j)^2\big]$ and $\Delta^2 \geqslant \mathbb{E}\big[X_i^2\big] \geqslant \alpha^2 > \mathbb{E}\big[Y_i^2\big]$ since $X_1 = 0$. Then using Lemma 3.9 we know that for all $\lambda \in \mathbb{R}$:

$$\mathbb{P}\Big[\max_{i\leqslant m} X_i < \lambda - 2\Delta\Big] \leqslant \mathbb{P}\Big[\max_{i\leqslant m} Y_i < \lambda\Big].$$

Now using Lemma 3.8 we obtain for $m \geqslant 3u$:

$$\mathbb{P}\left[\max_{i\leqslant m} X_i < \frac{\alpha}{\sqrt{2}}\sqrt{\log\frac{m}{3u}} - 2\Delta\right] \leqslant e^{-u}.$$

□

The following lemma describes the key properties of the tree $\mathcal{T}$ as computed by Algorithm 12. We show that the supremum $\sup_{x>s} f(x) - f(s)$ at every depth is bounded from below by the sum of the values found in Lemma 3.10, up to constant factors.

Lemma 3.11 (Anti-Concentration for the Tree)**.** *Fix any $u > 0$ and set accordingly $u_i = u + 2^i + i \log 2$ for any $i > 0$. For $\mathcal{T}$ the tree obtained by Algorithm 12, we have for all $s \in \mathcal{T}_h$ with probability at least $1 - e^{-u_h}$ that:*

$$\sup_{x>s} f(x) - f(s) \geqslant c_u^{-1} \sup_{x>s} V_h(s,x),$$

where $V_h(s,x) = \sum_{i=h}^{\infty} \Delta_i(x)\Big(\sqrt{2^{i-3} - \frac{1}{8}\log(3u_i + 3\log 2)} - 2\Big)$, and $\Delta_i(x)$ is the radius of the cell of x at depth i, and $c_u \in \mathbb{R}$ depends on u only.

Proof. We first show that we can restrict the study of $V_h(s,x)$ to only the summands obtained by pruning $\mathcal{T}$, up to constant factors. To lighten the notations, we write:

$$b_i \triangleq \sqrt{2^{i-3} - \tfrac{1}{8}\log(3u_i + 3\log 2)} - 2.$$

Then for a sequence $t_h \triangleq p_h(x), \dots, t_{h+j} \triangleq p_{h+j}(x)$ of parents of x, if t_h is the single pruned node, then,

$$\begin{aligned}\sum_{i=h}^{h+j-1} \Delta_i(x) b_i = \Delta_h(x) \sum_{i=h}^{h+j-1} 2^{h-i} b_i \\ \leqslant c_u \Delta_h(x) b_h,\end{aligned}$$

where $c_u \in \mathbb{R}$ depends on u only, and we used that $\Delta_{h+i}(x)$, the radius of the cell at depth $h+i$ containing x, decreases geometrically for non-pruned nodes. By denoting $\mathcal{P}_h(x)$ the set of parents of x from depth h which are pruned nodes, we thus proved for all $x \in \mathcal{X}$:

$$V'_h(s,x) \triangleq \sum_{t_i \in \mathcal{P}_h(x)} \Delta_i(t_i) b_i \geqslant c_u^{-1} V_h(s,x). \tag{3.15}$$

We now prove Lemma 3.11 by showing that $\sup_{x>s} f(x) - f(s) \geqslant V'_h(s,x^\star)$ for all $x^\star > s$ with probability at least $1 - e^{-u_h}$, by backward induction on $\mathcal{P}_h(x)$, from the deepest nodes to the shallowest ones. Since for the leaves $\sup_{x>s} f(x) - f(s) = 0 = V'_h(s,x^\star)$, the property is initially true. Let us assume that it is true at depth $h' > h$ and prove it at depth h. Let s in $\mathcal{T}_h$ and $x^\star$ in $\mathcal{X}$. If $p_{h+1}(x^\star)$ is not pruned, we have nothing to do and just call the induction hypothesis with $\sup_{x>s} f(x) - f(s) \geqslant \sup_{x>t} f(x) - f(t)$ where $p(t) = s$. Otherwise note that,

$$\begin{aligned}\sup_{x>s} f(x) - f(s) &= \max_{t:p(t)=s} \Big(f(t) - f(s) + \sup_{x \geqslant t} f(x) - f(t)\Big) \\ &\geqslant \max_{t:p(t)=s} \Big(f(t) - f(s)\Big) + \min_{t:p(t)=s} \Big(\sup_{x \geqslant t} f(x) - f(t)\Big).\end{aligned} \tag{3.16}$$

Since the children have been pruned, we know that their number is e^{2^h}. Now thanks to Lemma 3.10, with probability at least $1 - \frac{1}{2}e^{-u_h}$,

$$\max_{t:p(t)=s} f(t) - f(s) \geqslant \frac{\Delta_h(x^\star)}{2\sqrt{2}} \sqrt{2^h - \log(3u_h + 3\log 2)} - 2\Delta_h(x^\star) = \Delta_h(x^\star) b_h, \tag{3.17}$$

where we used that $\ell(t_i, t_j) \geqslant \frac{1}{2}\Delta_h(x^\star)$ for $p(t_i) = p(t_j) = s$ by construction of $\mathcal{T}$. Now by the induction hypothesis and a union bound, we have with probability at least $1 - e^{-u_{h+1}+2^h}$ that:

$$\min_{t:p(t)=s} \sup_{x \geqslant t} \big(f(x) - f(t)\big) \geqslant \min_{t:p(t)=s} \sup_{x>t} V'_{h+1}(t, x). \tag{3.18}$$

By construction of the pruning procedure, we know that the children minimizing the function $t \to \sup_{x>t} V'_{h+1}(t, x)$ is the pruned node $p_{h+1}(x^\star)$. With $u_{h+1} - 2^h = u_h + \log 2$, the results of Eq. 3.18 holds with probability at least $1 - \frac{1}{2}e^{-u_h}$, we thus obtain with probability at least $1 - e^{-u_h}$:

$$\sup_{x>s} f(x) - f(s) \geqslant V'_h(s, x^\star),$$

which uses Eq. 3.16 together with Eq. 3.17, closes the induction and the proof of Lemma 3.11 with Eq. 3.15. □

The proof of Theorem 3.6 follows from Lemma 3.11 by a union bound on $h \in \mathbb{N}$ and remarking that $\omega_h \geqslant \sup_{x>s} V_h(s, x)$ up to constant factors.

3.2.6 Conclusion and Discussions

In this section on Gaussian process optimization with metric spaces, we showed that sound algorithms can be implemented when the kernel generates arbitrary totally bounded metric spaces. By using generic chaining and discretization trees, we exhibited the link between the metric entropy of this metric space and regret bounds for optimization algorithms. These contributions form a step toward Bayesian optimization for non-parametric search spaces. The lower bound we derived is a crucial element to further derive lower bounds on the cumulative regret.

3.3 Beyond Gaussian Processes

One benefit of the nonparametric model presented in the previous section is that it is easily adaptable to other stochastic processes more complex that centered Gaussian processes. We first describe what properties are necessary for the theorems to hold. We then give a concrete example of non-Gaussian processes relevant for many settings.

3.3.1 Generic Stochastic Processes

For arbitrary stochastic process f, we first define the following function, which extends the previous canonical pseudo-metric of Gaussian processes. Let $\ell_u(x_1, x_2)$ for $x_1, x_2 \in \mathcal{X}$ and $u \geqslant 0$ be the following confidence bound on the increments of f:

$$\ell_u(x_1, x_2) \triangleq \inf \Big\{ s \in \mathbb{R} : \mathbb{P}\big[f(x_1) - f(x_2) > s\big] < e^{-u} \Big\}.$$

In short, $\ell_u(x_1, x_2)$ is the best bound satisfying $\mathbb{P}\big[f(x_1) - f(x_2) \geqslant \ell_u(x_1, x_2)\big] < e^{-u}$. For particular distributions of f, it is possible to obtain closed formulae for ℓ_u. However in the

present work, upper bounds on ℓ_u will suffice. If it exists a pseudo-metric $\ell(\cdot,\cdot)$ and a function $\psi(\cdot,\cdot)$ bounding the logarithm of the Laplace transform of the increments, that is,

$$\log \mathbb{E}\Big[e^{\lambda(f(x_1)-f(x_2))}\Big] \leqslant \psi(\lambda, \ell(x_1,x_2)),$$

for $x_1, x_2 \in \mathcal{X}$ and $\lambda \in I \subseteq \mathbb{R}$, then using the Cramer-Chernoff method as before,

$$\ell_u(x_1,x_2) \leqslant \psi^{*-1}(u, \ell(x_1,x_2)), \tag{3.19}$$

where $\psi^*(s,\delta) \triangleq \sup_{\lambda\in I}\big(\lambda s - \psi(\lambda,\delta)\big)$ is the Legendre-Fenchel dual of ψ and $\psi^{*-1}(u,\delta) \triangleq \inf\big\{s\in\mathbb{R} : \psi^*(s,\delta) > u\big\}$ denotes its generalized inverse. In that case, we say that f is a (ℓ,ψ)-process. For example if f is (c,ν)-sub-Gamma, that is:

$$\psi(\lambda,\delta) \leqslant \frac{\nu\lambda^2\delta^2}{2(1-c\lambda\delta)}, \tag{3.20}$$

for $c,\nu\in\mathbb{R}$, we obtain,

$$\ell_u(x_1,x_2) \leqslant \big(cu + \sqrt{2\nu u}\big)\ell(x_1,x_2). \tag{3.21}$$

The generality of Eq. 3.20 makes it convenient to derive bounds for a wide variety of processes beyond Gaussian processes, as we see for example in Section 3.3.2.

Generic Chaining for Stochastic Processes

Without any additional effort, we can take the proof of Theorem 3.2 and obtain directly an equivalent result for stochastic processes. It suffices to replace all occurrences of $\sqrt{2u}\ell(\cdot,\cdot)$ by the corresponding term $\ell_u(\cdot,\cdot)$.

Theorem 3.7 (Generic Chaining for Stochastic Processes)**.** *Fix any $u>0$ and $a>1$, and $(n_h)_{h\in\mathbb{N}}$ an increasing sequence of integers. Set $u_i \triangleq u + n_i + \log\big(i^a\zeta(a)\big)$ where ζ is the Riemann zeta function. Then for any tree such that $|\mathcal{T}_h| \leqslant e^{n_h}$ we have that,*

$$\forall h\geqslant 0, \forall s\in\mathcal{T}_h,\ \sup_{x\succ s} f(x) - f(s) \leqslant \omega(s),$$

holds with probability at least $1-e^{-u}$, where for $s\in\mathcal{T}_h$,

$$\omega(s) \equiv \omega(s,\mathcal{X},\mathcal{T},\ell) \triangleq \sup_{x\succ s}\sum_{i>h}\ell_{u_i}\big(p_i(x),p_{i-1}(x)\big).$$

Classical Chaining for (ℓ,ψ)-Processes and Sub-Gamma Processes

The previous theorem is extremely general but one cannot tell much without restricting to smooth classes of stochastic processes. We specify what it implies for (ℓ,ψ)-processes. In that case, we have with Eq. 3.19:

$$\forall h\geqslant 0, \forall s\in\mathcal{T}_h,\ \omega(s) \leqslant \sup_{x\succ s}\sum_{i>h}\psi^{*-1}\Big(u_i, \ell\big(p_i(x),p_{i-1}(x)\big)\Big).$$

Introducing the ℓ-radius of the cells, this rewrites as:

$$\forall h \geqslant 0, \forall s \in \mathcal{T}_h,\ \omega(s) \leqslant \sup_{x \succ s} \sum_{i>h} \psi^{*-1}\big(u_i, \Delta_{i-1}(x)\big).$$

Therefore, the previous greedy construction of a discretization tree by Algorithm 9 leads to the following upper bound involving the metric entropy $H(\cdot, \mathcal{X}, \ell)$:

$$\begin{aligned} \forall h \geqslant 0,\ \omega_h &\leqslant \sum_{i>h} \psi^{*-1}(u_i, \varepsilon_i), \\ \text{where } \varepsilon_h &\triangleq \Delta(\mathcal{X}) 2^{-h}, \\ \text{and } u_h &= u + H(\varepsilon_h) + \log\big(h^a \zeta(a)\big). \end{aligned} \tag{3.22}$$

When f is sub-Gamma and the covering dimension (Definition 2.1) is finite, we obtain a generalization of the classical chaining Theorem 3.3.

Theorem 3.8 (Sub-Gamma Process with Covering Dimension)**.** *If f is (c, ν)-sub-Gamma, for all $d > \dim(\mathcal{X}, \ell)$ and $d = \dim(\mathcal{X}, \ell)$ if the dimension is attained, with probability at least $1 - e^{-u}$:*

$$\forall h \geqslant 0, \forall s \in \mathcal{T}_h,\ \sup_{x \succ s} f(x) - f(s) = \mathcal{O}\Big(\big(c(u + dh) + \sqrt{\nu(u + dh)}\big) 2^{-h}\Big).$$

Proof. Following the lines of Theorem 3.3 we know that is exists a constant $c' \in \mathbb{R}$ such that for all $i \geqslant 0$,

$$\begin{aligned} H(\varepsilon_i) &\leqslant c' - d \log \varepsilon_i, \\ u_i &= \mathcal{O}\big(u + di\big). \end{aligned}$$

With Eq. 3.22 for a sub-Gamma process we get, knowing that $\sum_{i \geqslant h} i 2^{-i} = \mathcal{O}\big(h 2^{-h}\big)$,

$$\omega_h = \mathcal{O}\Big(\big(c(u + dh) + \sqrt{\nu(u + dh)}\big) 2^{-h}\Big).$$

□

High Confidence Empirical Intervals

Assume that given i observations $Y_i \triangleq (y_1, \dots, y_i)$ at queried locations X_i, we can compute empirical bounds $L_{i,u}(x)$ and $U_{i,u}(x)$ for all $u > 0$ and $x \in \mathcal{X}$, such that:

$$\mathbb{P}\Big[f(x) \in \big(L_{i,u}(x), U_{i,u}(x)\big)\Big] \geqslant 1 - e^{-u}. \tag{3.23}$$

Then for any $h(i) \in \mathbb{N}$ chosen like in Section 3.2.2, we obtain by union bounds on $x \in \mathcal{T}_{h(i)}$ and $i \in \mathbb{N}$ that:

$$\mathbb{P}\Big[\forall i \in \mathbb{N}, \forall x \in \mathcal{T}_{h(i)},\ f(x) \in \big(L_{i,u_i}(x), U_{i,u_i}(x)\big)\Big] \geqslant 1 - e^{-u},$$

where $u_i = u + n_{h(i)} + \log\big(i^a \zeta(a)\big)$ for any $a > 1$. Our UCB decision rule for the next query becomes:

$$x_i \in \operatorname*{argmax}_{x \in \mathcal{T}_{h(i)}} U_{i,u_i}(x). \tag{3.24}$$

Combining this with Theorem 3.7, we are able to prove the following bound linking the regret with $\omega_{h(i)}$ and the width of the confidence interval.

Theorem 3.9 (Generic Regret Bound with Stochastic Processes)**.** *For all $u > 0$, the algorithm selecting $x_{n+1} \in \operatorname{argmax}_{x \in \mathcal{T}_{h(n)}} U_{n,u_n}(x)$ has a cumulative regret lower than, with probability at least $1 - 2e^{-u}$:*

$$R_n \leqslant \sum_{i=1}^{n} \Big(\omega_{h(i)} + U_{i,u_i}(x_i) - L_{i,u_i}(x_i) \Big).$$

The proof is similar to the one of Theorem 3.4, using U_{i,u_i} and L_{i,u_i} instead of the Gaussian posterior distribution. In order to select the level of discretization $h(i)$ to reduce the bound on the regret, it is required to have explicit bounds on ω_i and the confidence intervals. Like before, choosing with c a constant,

$$\forall i \in \mathbb{N},\ h(i) = \min \left\{ i : \mathbb{N} : \omega_i \leqslant c\sqrt{\frac{\log i}{i}} \right\},$$

we obtain $\sum_{i=1}^{n} \omega_{h(i)} \leqslant \mathcal{O}\big(\sqrt{n \log n}\big)$. When f is sub-Gamma and the covering dimension is finite, Theorem 3.8 tells us that our previous choice of $h(i) = \lceil 1/2 \log_2 i \rceil$ leads to:

$$\omega_{h(i)} \leqslant \mathcal{O}\left((u + d \log i) i^{-1/2} \right),$$

and since $\sum_{i=1}^{n} i^{-1/2} \leqslant 2\sqrt{n}$ and $\sum_{i=1}^{n} i^{-1/2} \log i \leqslant 2\sqrt{n} \log n$, we know that:

$$\sum_{i=1}^{n} \omega_{h(i)} \leqslant \mathcal{O}\Big((u + d \log n)\sqrt{n} \Big),$$

a slightly bigger term that what we obtained for centered Gaussian processes.

3.3.2 Quadratic Forms of Gaussian Processes

The dominating model in Bayesian optimization is by far the Gaussian process. Yet, it is a very common task to attempt minimizing a regret on functions which do not look like Gaussian processes. Consider the typical cases where f has the form of a mean square error or a Gaussian likelihood. In both cases, minimizing f is equivalent to minimize a sum of squares, which we cannot assume to be sampled from a Gaussian process. To alleviate this problem, we show that this objective fits in our generic analysis. Indeed, if we consider that f is a sum of squares of Gaussian processes, then f is sub-Gamma with respect to a natural pseudo-metric. Three realizations of a sum of squared Gaussian processes are illustrated in Figure 3.6. Note that in this section we are dealing with minimization instead of maximization. In this particular setting we allow the algorithm to observe directly the noisy values of the *separated* Gaussian processes, instead of the sum of their square. To simplify the forthcoming arguments, we will choose independent and identically distributed processes, but one can remove the covariances between the processes by Cholesky decomposition of the covariance matrix, and then our analysis adapts easily to processes with non identical distributions.

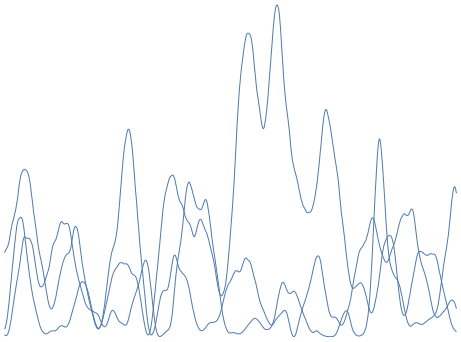


Figure 3.6. – Realizations of sums of squared Gaussian processes

The Stochastic Smoothness of a Sum of Squared Gaussian Processes

Let $f(x) \triangleq \sum_{j=1}^{N} g_j^2(x)$, where $(g_j)_{1 \leqslant j \leqslant N}$ are independent centered Gaussian processes $g_j \overset{\text{iid}}{\sim} \mathcal{GP}(0, k)$ with stationary covariance k such that $k(x, x) = \kappa$ for every $x \in \mathcal{X}$. We have by exact computation of the squared Gaussian integral, for $x_1, x_2 \in \mathcal{X}$ and $\lambda < (2\kappa)^{-1}$:

$$\log \mathbb{E}\Big[e^{\lambda(f(x_1)-f(x_2))}\Big] = -\tfrac{N}{2} \log\Big(1 - 4\lambda^2(\kappa^2 - k^2(x_1, x_2))\Big).$$

Therefore with ℓ and ψ defined as follows:

$$\ell(x_1, x_2) = 2\sqrt{\kappa^2 - k^2(x_1, x_2)},$$
$$\text{and } \psi(\lambda, \delta) = -\tfrac{N}{2} \log\big(1 - \lambda^2\delta^2\big),$$

we conclude that f is a (ℓ, ψ)-process. Since $-\log(1 - x^2) \leqslant \frac{x^2}{1-x}$ for $0 \leqslant x < 1$, which can be proved by series comparison, we obtain that f is sub-Gamma with parameters $\nu = N$ and $c = 1$. Now with Eq. 3.21,

$$\ell_u(x_1, x_2) \leqslant (u + \sqrt{2uN})\ell(x_1, x_2).$$

Furthermore when $\mathcal{X} \subseteq [0, R]^d$, we have that $\ell(x_1, x_2) \leqslant c' \, \|x_1 - x_2\|_2^{1/\alpha}$ for all $x_1, x_2 \in \mathcal{X}$ is satisfied with $\alpha = 1$ when k is linear, squared exponential or Matérn with parameter $\nu > 1$, and satisfied with $\alpha = 2$ for Matérn kernel with parameter $\nu = 1/2$. In these cases, the covering dimension attains $\dim(\mathcal{X}, \ell) = \alpha d$, and Theorem 3.8 leads to:

$$\forall h \geqslant 0, \ \omega_h \leqslant \mathcal{O}\Big(\big(u + \alpha d h + \sqrt{N(u + \alpha d h)}\big)2^{-h}\Big). \tag{3.25}$$

Confidence Intervals for Squared Gaussian Processes

As mentioned above, we consider here that we are given separated noisy observations $\mathbf{Y}_n^j$ for each of the N processes. Deriving confidence intervals for f given $\big(\mathbf{Y}_n^j\big)_{j \leqslant N}$ is a tedious task since the posterior processes g_j given $\mathbf{Y}_n^j$ are not standard nor centered, which excludes

known results for χ^2 random variables. We propose here a solution based directly on a careful analysis of Gaussian integrals. We write for positive a:

$$\operatorname{erf}(a) \triangleq \frac{2}{\sqrt{\pi}} \int_0^a e^{-t^2} \, \mathrm{d}t \text{ and } \operatorname{erfc}(a) \triangleq 1 - \operatorname{erf}(a).$$

Lemma 3.12 (Tails of Squared Gaussian)**.** *Let $X \sim \mathcal{N}(\mu, \sigma^2)$ and $s > 0$. We have:*

$$\mathbb{P}\Big[X^2 \notin \big(l^2, u^2\big)\Big] < e^{-s^2},$$

for $u \geqslant |\mu| + \sqrt{2}\sigma s$ and $l \leqslant \big(|\mu| - \sqrt{2}\sigma s\big) \vee \sqrt{2}\sigma \operatorname{erf}^{-1}\Big(\frac{1}{2}\operatorname{erf}(\sqrt{2}\mu\sigma^{-1} + s) - \frac{1}{2}\operatorname{erf}(s)\Big)$.

Proof. Let $X \sim \mathcal{N}(\mu, \sigma^2)$ with $\mu \geqslant 0$ without loss of generality. For all $0 < l < u \in \mathbb{R}$ we have:

$$\begin{aligned}\mathbb{P}\Big[X^2 \notin (l^2, u^2)\Big] &= \mathbb{P}\Big[X \notin (l, u) \cup (-u, -l)\Big] \\ &= \tfrac{1}{2}\Big(\operatorname{erfc}\Big(\frac{u-\mu}{\sqrt{2}\sigma}\Big) + \operatorname{erfc}\Big(\frac{u+\mu}{\sqrt{2}\sigma}\Big) + \operatorname{erf}\Big(\frac{\mu+l}{\sqrt{2}\sigma}\Big) - \operatorname{erf}\Big(\frac{\mu-l}{\sqrt{2}\sigma}\Big)\Big).\end{aligned}$$

Fix $s > 0$ and $u = \mu + \sqrt{2}\sigma s$. If $l \leqslant \mu - \sqrt{2}\sigma s$, which means $s < \mu(\sqrt{2}\sigma)^{-1}$, we get:

$$\mathbb{P}\Big[X^2 \notin (l^2, u^2)\Big] \leqslant \tfrac{1}{2}\Big(\operatorname{erfc}(s) + \operatorname{erfc}\big(\sqrt{2}\mu\sigma^{-1} + s\big) + \operatorname{erf}\big(\sqrt{2}\mu\sigma^{-1} - s\big) - \operatorname{erf}(s)\Big).$$

Remarking that $\operatorname{erfc}\big(\sqrt{2}\mu\sigma^{-1} + s\big) + \operatorname{erf}\big(\sqrt{2}\mu\sigma^{-1} - s\big) \leqslant 1$, we obtain:

$$\mathbb{P}\big[X^2 \notin (l^2, u^2)\big] \leqslant \operatorname{erfc}(s).$$

Now for $s > \mu(\sqrt{2}\sigma)^{-1}$, if $l \leqslant \sqrt{2}\sigma \operatorname{erf}^{-1}\Big(\frac{1}{2}\operatorname{erf}(\sqrt{2}\mu\sigma^{-1} + s) - \frac{1}{2}\operatorname{erf}(s)\Big)$ we have that $\operatorname{erf}\Big(\frac{\mu+l}{\sqrt{2}\sigma}\Big) - \operatorname{erf}\Big(\frac{\mu-l}{\sqrt{2}\sigma}\Big) \leqslant 2\operatorname{erf}\big(\frac{l}{\sqrt{2}\sigma}\big) \leqslant \operatorname{erf}(\sqrt{2}\mu\sigma^{-1} + s) - \operatorname{erf}(s)$. Therefore we also get:

$$\mathbb{P}\Big[X^2 \notin (l^2, u^2)\Big] \leqslant \operatorname{erfc}(s).$$

We finish the proof of Lemma 3.12 by the standard inequality $\operatorname{erfc}(s) \leqslant e^{-s^2}$. □

Using this lemma, we compute the confidence interval for $f(x)$ by a union bound over N. Denoting μ_i^j and σ_i^j the posterior expectation and deviation of g_j given $\mathbf{Y}_i^j$ (computed as in Eq. 2.14 and 2.16), the confidence interval follows for all $x \in \mathcal{X}$:

$$\mathbb{P}\Big[\forall j \leqslant N,\ g_j^2(x) \in \big(L_{i,u}^j(x), U_{i,u}^j(x)\big)\Big] \geqslant 1 - e^{-u}, \tag{3.26}$$

where we choose:

$$U_{i,u}^j(x) = \Big(|\mu_i^j(x)| + \sqrt{2(u + \log N)}\sigma_{i-1}^j(x)\Big)^2 \tag{3.27}$$

$$\text{and } L_{i,u}^j(x) = \Big(|\mu_i^j(x)| - \sqrt{2(u + \log N)}\sigma_{i-1}^j(x)\Big)_+^2. \tag{3.28}$$

Note that in Lemma 3.12 we provide a better confidence interval for $L_{i,u}$, but we do not use it here to simplify the notations. We are now ready to use Theorem 3.9 to control R_n by a union bound for all $i \in \mathbb{N}$ and $x \in \mathcal{T}_{h(i)}$. Note that under the event of Theorem 3.9, we have the following:

$$\forall j \leqslant N, \forall i \in \mathbb{N}, \forall x \in \mathcal{T}_{h(i)},\ g_j^2(x) \in \left(L_{i,u_i}^j(x), U_{i,u_i}^j(x)\right),$$

Then we also have:

$$\forall j \leqslant N, \forall i \in \mathbb{N}, \forall x \in \mathcal{T}_{h(i)},\ \left|\mu_i^j(x)\right| \leqslant \left|g_j(x)\right| + \sqrt{2(u_i + \log N)}\sigma_{i-1}^j(x),$$

Since $\mu_0^j(x) = 0$, $\sigma_0^j(x) = \kappa$ and $u_0 \leqslant u_i$ we obtain $\left|\mu_i^j(x)\right| \leqslant \sqrt{2(u_i + \log N)}\left(\sigma_{i-1}^j(x) + \kappa\right)$. We now consider the regret of Algorithm 13 that selects the minimizer of the lower confidence bound. The regret here is taken with respect to the infimum of f to match the problem of minimization. Theorem 3.9 says with probability at least $1 - 2e^{-u}$:

$$R_n \leqslant \sum_{i \leqslant n}\left(\omega_{h(i)} + 8\sum_{j \leqslant N}(u_i + \log N)\left(\sigma_{i-1}^j(x) + \kappa\right)\sigma_{i-1}^j(x_i)\right).$$

It is now possible to proceed as in Section 2.2.4, and bound the sum of posterior variances with the informational quantity γ_n:

$$R_n \leqslant \mathcal{O}\left(N u_n\left(\sqrt{n\gamma_n} + \gamma_n\right) + \sum_{i \leqslant n}\omega_{h(n)}\right).$$

As before, under the conditions of Eq. 3.25 and choosing the discretization level $h(i) = \left\lceil \frac{1}{2}\log_2 i \right\rceil$ we obtain $\omega_{h(i)} = \mathcal{O}\left(i^{-\frac{1}{2}}\left(u + \frac{1}{2}D\log i\right)\sqrt{N}\right)$, that is, the following guarantees holds.

Corollary 3.2 (Regret Bounds for Quadratic Form of Gaussian Processes). *Let f be a sum of independent squared centered Gaussian processes with known kernel. For all $u > 0$, the algorithm selecting $x_{n+1} \in \operatorname{argmin}_{x \in \mathcal{T}_{h(n)}} L_{n,u_n}(x)$ with $h(n) = \left\lceil \frac{1}{2}\log_2 i \right\rceil$ and $L_{n,u}(x) = \sum_{j \leqslant N} L_{n,u}^j(x)$ from Eq. 3.28 incurs a (minimization) cumulative regret on f lower than:*

$$R_n \leqslant \mathcal{O}\left(N\left(\sqrt{n\gamma_n \log n} + \gamma_n\right) + \sqrt{Nn}\log n\right),$$

with probability at least $1 - 2e^{-u}$.

3.3.3 Conclusion and Discussions

In this section, we described a methodology to build sound Bayesian optimization algorithm working on non-Gaussian processes. We illustrated the importance of non-Gaussian setting with the optimization of quadratic forms, and gave a precise algorithm for this setting. The regret bound we obtained are almost equivalent to the ones from Gaussian processes. Since optimization of quadratic forms is an extremely common task, we believe that this contribution may have substantial impact for practitioners.

Algorithm 13: GP^2-UCB (k, η, u, a) for Minimizing Sum of N Squared GPs

$h_0 \leftarrow -1$
$\mathcal{T} \leftarrow \varnothing$
for $n = 0, 1, \ldots$ **do**
  $h_n \leftarrow \lceil 1/2 \log_2 n \rceil$
  if $h_n \neq h_0$ **then**
    $\varepsilon_h \leftarrow 2^{-h_n - 1} \Delta(\mathcal{X})$
    $T_h \leftarrow \mathsf{GreedyCover}\big(\varepsilon_h, \mathcal{X} \backslash \bigcup_{x \in \mathcal{T}} \mathcal{B}(x, \varepsilon_h)\big)$
    $\forall t \in T_h,\ p(t) \leftarrow \operatorname{argmin}_{s \in \mathcal{T}} \ell(t, s)$
    $\mathcal{T} \leftarrow \mathcal{T} \cup T_h$
  end
  for $1 \leqslant j \leqslant N$ **do**
    Compute μ_n^j and σ_n^j (Eq. 2.14, 2.16)
    $u_n \leftarrow u + \log|\mathcal{T}| + \log\left(n^a \zeta(a)\right)$
    for $x \in T$ **do**
      $L_1 \leftarrow |\mu_n^j(x)| - \sqrt{2(u_n + \log N)}\sigma_n^j(x)$
      $L_2 \leftarrow \sqrt{2}\sigma_n^j(x) \operatorname{erf}^{-1}\left(\frac{1}{2}\operatorname{erf}\left(\sqrt{2}\mu_n^j(x){\sigma_n^j}^{-1} + u_n\right) - \frac{1}{2}\operatorname{erf}(u_n)\right)$
      $L_n^j(x) \leftarrow \left(L_1 \vee L_2\right)^2$
    end
  end
  $x_{n+1} \leftarrow \operatorname{argmin}_{x \in T_h} \sum_{j \leqslant N} L_n^j(x)$
  $\left\{y_{n+1}^j\right\}_{j \leqslant N} \leftarrow \mathbf{Query}\left(x_{n+1}\right)$
end
return T

Non-Smooth Optimization and Ranking

4

This chapter introduces the innovative framework from Malherbe et al. (2016), a joint work with Cédric Malherbe for deterministic optimization of non-smooth functions, possibly discontinuous. In Section 4.2, we describe the framework and provide the main definitions. We express the complexity of the optimization problem with respect to ranking structures, a novel assumption on the level sets. In Section 4.3, we propose and analyze the RankOpt algorithm which requires a prior information on the ranking structure underlying the unknown function. In Section 4.4, an adaptive version of the algorithm is presented. Companion results which establish the equivalence between learning algorithms and optimization procedures are discussed in Section 4.5, as they support implementation choices. The adaptive version of the algorithm is compared to other global optimization algorithms in Section 4.6.

Contents

4.1 Introduction **101**
4.2 Global Optimization and Ranking Structure **101**
4.2.1 Setup and Notations 101
Setup 101
Notations 102
4.2.2 The Ranking Structure of a Real-Valued Function 102
Induced Ranking Rules 102
Ranking Structures 103
Identifiability and Regularity 104
4.3 Optimization with Fixed Ranking Structure **105**
4.3.1 The RankOpt Algorithm 105
Active Subset of Ranking Rules 105
Properties and Connection to Active Learning 106
4.3.2 Convergence analysis 106
Consistency 107
Upper and Lower Bounds on the Loss 108
4.4 Adaptive Algorithm and Stopping Time Analysis **110**
4.4.1 The AdaRankOpt Algorithm 110
4.4.2 Theoretical Properties of AdaRankOpt 111
Consistency 111
Stopping Time and Rademacher Complexity 111
Upper Bound on the Loss 113
4.5 Computational Aspects **113**
4.5.1 General ranking structures 113
Uniform Sampling in the Relevant Region 114
Updating the Index 114
4.5.2 Practical Solutions for Particular Ranking Structures 114
Polynomial ranking rules 115
Convex ranking rules 116
4.6 Experiments **117**
4.6.1 Protocol of the Empirical Assessment 117

Parameters of AdaRankOpt 117
Optimization Objectives 117
4.6.2 Empirical Comparisons 118
Controlled Random Search 118
Lipschitz Optimization 118
Evolutionary Algorithms 118
4.7 Conclusion and Discussion **119**

4.1 Introduction

Previous theoretical works in the literature and in this thesis systematically require the unknown function to be smooth, at least locally around the optimum. In this chapter, we propose to explore concepts from ranking theory based on overlaying estimated level sets Clémençon et al. (2008) in order to develop global optimization algorithms that do not rely on the smoothness of the function. The idea behind this approach is simple: even if the unknown function presents arbitrary large variations, most of the information required to identify its optimum may be contained in its induced ranking rule, i.e. how the level sets of the function are included one in another. To exploit this idea, we introduce a novel optimization scheme where the complexity of the function is characterized by the underlying pairwise ranking which it defines. Our contribution is twofold: first, we introduce two novel global optimization algorithms that learn the ranking rule induced by the unknown function with a sequential scheme, and second, we provide mathematical results in terms of statistical consistency and convergence to the optimum. Moreover the algorithms proposed lead to efficient implementations and they display competitive performance on the classical benchmarks for global optimization as shown at the end of this chapter.

4.2 Global Optimization and Ranking Structure

This section introduces the definitions and concepts on which our approach is built. We define the ranking rules of functions, and deduce our notion of complexity for the optimization problem via ranking structures. We finally give the precise assumptions required to obtain convergence rates.

4.2.1 Setup and Notations

In this chapter, we consider the problem of sequentially maximizing a fixed unknown function $f : \mathcal{X} \to \mathbb{R}$, potentially not continuous, using deterministic observations. The regularity of f will be controlled by ranking structures.

Setup

We restrict our analysis to $\mathcal{X} \subset \mathbb{R}^d$, and we further require that $\mathcal{X}$ is compact and convex for technical reasons. We focus here on the objective of identifying some point

$$x^\star \in \operatorname*{argmax}_{x \in \mathcal{X}} f(x),$$

with a minimal amount of function evaluations. Since we do not have assumptions neither on the smoothness on f nor on its continuity, the extend of the values of f is arbitrary, so are the cumulative or simple regrets R_n or S_n. The observations of an optimization algorithm do not suffer any source of perturbation due to noise. After n iterations, the algorithm returns the location of the highest value observed so far:

$$x_{\hat{\imath}_n} \text{ where } \hat{\imath}_n \in \operatorname*{argmax}_{i=1,\dots,n} f(x_i).$$

The analysis provided here considers that the horizon is unknown, that is the number n of evaluation points is not fixed.

Notations

For any $x \triangleq [x_1 \dots x_d] \in \mathbb{R}^d$, we define the standard ℓ_2-norm $\|x\|_2^2 \triangleq \sum_{i=1}^d x_i^2$, we denote by $\langle \cdot, \cdot \rangle$ the corresponding inner product and we denote by:

$$\mathcal{B}(x, r) \triangleq \left\{ x' \in \mathbb{R}^d : \|x - x'\|_2 \leqslant r \right\},$$

the corresponding ℓ_2-ball of radius $r \geqslant 0$ centered in x. For any set $\mathcal{X} \subset \mathbb{R}^d$, we define its inner-radius as:

$$\mathrm{rad}(\mathcal{X}) \triangleq \sup \left\{ r > 0 : \exists x \in \mathcal{X} \text{ s.t. } \mathcal{B}(x, r) \subseteq \mathcal{X} \right\},$$

and its diameter as:

$$\Delta(\mathcal{X}) \triangleq \sup_{x, x' \in \mathcal{X}} \|x - x'\|_2 .$$

We denote by $\lambda(\mathcal{X})$ the volume of $\mathcal{X}$ where λ stands for the Lebesgue measure. Finally, we denote by $C^0(\mathcal{X})$ the set of continuous functions defined on $\mathcal{X}$ taking values in $\mathbb{R}$, and we denote by $\mathcal{P}_N(\mathcal{X})$ the set of (multivariate) polynomial functions of degree N defined on $\mathcal{X}$. We denote by $\mathcal{U}(\mathcal{A})$ the uniform distribution over a bounded measurable domain $\mathcal{A}$.

4.2.2 The Ranking Structure of a Real-Valued Function

In this section, we introduce the ranking structure as a complexity characterization for a general real-valued function to be optimized.

Induced Ranking Rules

First, we observe that every real-valued function induces an order relation over the input space $\mathcal{X}$, and the underlying ordering induces a ranking rule which records pairwise comparisons between evaluation points.

Definition 4.1 (Induced ranking rule)**.** *The ranking rule $r_f : \mathcal{X} \times \mathcal{X} \to \{-1, 0, 1\}$ induced by a function $f : \mathcal{X} \to \mathbb{R}$ is defined by:*

$$r_f(x, x') \triangleq \begin{cases} 1 & \text{if } \ f(x) > f(x') \\ 0 & \text{if } \ f(x) = f(x') \\ -1 & \text{if } \ f(x) < f(x') \end{cases}$$

for all $x, x' \in \mathcal{X}$.

The key argument of the paper is that the difficulty of the optimization of any weakly regular real-valued function only depends on the nested structure of its level sets. Hence there is an equivalence class of real-valued functions that induce the same induced ranking rule as shown by the following proposition.

Lemma 4.1 (Ranking Rule Equivalence)**.** *Let $g \in C^0(\mathcal{X})$ be any continuous function. Then, a function $f : \mathcal{X} \to \mathbb{R}$ induces the same induced ranking rule with g (i.e. $r_f = r_g$) if and only if there exists a strictly increasing (not necessary continuous) function $h : \mathbb{R} \to \mathbb{R}$ such that $g = h \circ f$.*

Figure 4.1. – Two functions f and g that induce the same ranking

Proof. The equivalence of the ranking rules when such a h exists is a direct consequence of the definition of the ranking rules:

$$r_g(x, x') = \operatorname{sgn}\big(h \circ f(x) - h \circ f(x')\big) = \operatorname{sgn}\big(f(x) - f(x')\big) = r_f(x, x').$$

To prove the existence of h when the ranking rules are equal, we introduce the function:

$$M(x) \triangleq \lambda\Big(\big\{x' \in \mathcal{X} : r_f(x, x') = -1\big\}\Big).$$

We then show that there exists a strictly increasing function $h_1 : \mathbb{R} \to \mathbb{R}$ such that $f = h_1 \circ M$. Since for $x, x' \in \mathcal{X}$ we have that $M(x) = M(x')$ implies $f(x) = f(x')$ (proved by contradiction), f is constant on the iso-level sets $M^{-1}(y) \triangleq \big\{x \in \mathcal{X} : M(x) = y\big\}$, and let $h_1(y)$ be its value. We have $f(x) = h_1(M(x))$ for all $x \in \mathcal{X}$. Doing the same for g, there exists h_2 such that $g = h_2 \circ M$. Finally, $g = h \circ f$ with $h = h_2 \circ h_1^{-1}$. □

Lemma 4.1 states that even if the unknown function f admits non-continuous or large variations, up to a transformation h, there might exist a simpler function $g = h \circ f$ that shares the same induced ranking rule. Figure 4.1 gives an example of two functions that induce the same ranking while they display highly different regularity properties. As a second example, we may consider the problem of maximizing the following function over $\mathcal{X} = [0, 1/2]$:

$$f(x) = \begin{cases} 1 - \big|\ln(x)\big|^{-1} & \text{if } x \neq 0 \\ 1 & \text{otherwise} \end{cases}.$$

The function f in this case is not smooth around its unique global maximizer $x^\star = 0$ but shares the same induced ranking rule with $g(x) = -x$ over $\mathcal{X}$.

Ranking Structures

We can now introduce a complexity characterization of real-valued functions of a set $\mathcal{X}$ through the complexity class of its induced ranking rule. We call this class a ranking structure.

Definition 4.2 (Continuous Ranking Rules)**.** *We denote by:*

$$\mathcal{R}_\infty \triangleq \Big\{r_f : f \in C^0(\mathcal{X})\Big\},$$

the set of continuous ranking rules, that is ranking rules induced by continuous functions. Let f be a real-valued function. We say that f has a continuous ranking rule if $r_f \in \mathcal{R}_\infty$.

$f(x_1, x_2) = \exp(-x_1^2 - 2x_2^2),\ \alpha = 0$ $\qquad$ $f(x_1, x_2) = -x_1^4 - x_2^2,\ \alpha = 1$

Figure 4.2. – Illustration of the regularity of the level sets on two simple functions

Note that f having a continuous ranking rule does not imply that f is continuous. In the continuation of this definition, we further introduce two examples of more stringent ranking structures.

Definition 4.3 (Polynomial Ranking Rules)**.** *The set of polynomial ranking rules of degree N is defined as:*

$$\mathcal{R}_{\mathcal{P},N} \triangleq \Big\{ r_f :\ f \in \mathcal{P}_N(\mathcal{X}) \Big\}.$$

Similarly, we point out that even a polynomial function of degree N may admit a lower degree polynomial ranking rule. For example, consider the polynomial function $f(x) = (x^2 - 3x + 1)^9$. Since $f(x) = g(x^2 - 3x)$ where $g : x \mapsto (x+1)^9$ is a strictly increasing function, the ranking rule induced by f is a polynomial ranking rule of degree 2. The second class of ranking structures we introduce is a class of non-parametric rankings.

Definition 4.4 (Convex Ranking Rules)**.** *The set of convex ranking rules of degree N is defined as:*

$$\mathcal{R}_{\mathcal{C},N} \triangleq \Big\{ r \in \mathcal{R}_\infty : \forall x \in \mathcal{X}, \exists C_1, \dots, C_N \subset \mathcal{X},\ \{x' \in \mathcal{X} : r(x', x) \geqslant 0\} = \bigcup_{i=1}^{N} C_i,\ C_i \text{ is convex} \Big\}.$$

It is easy to see that the ranking rule of a function f is a convex ranking rule of degree N if and only if all the level sets of the function f are unions of at most N convex sets.

Identifiability and Regularity

We now state two conditions that will be used in the theoretical analysis: the first condition is about the identifiability of the maximum of the function and the second is about the regularity of f around its maximum.

Definition 4.5 (Identifiable Maximum)**.** *The maximum of a function $f : \mathcal{X} \to \mathbb{R}$ is said to be identifiable if for any $\varepsilon > 0$ arbitrary small,*

$$\lambda\Big(\big\{ x \in \mathcal{X} : f(x) \geqslant \sup_{x \in \mathcal{X}} f(x) - \varepsilon \big\} \Big) > 0.$$

Condition 4.5 prevents the function from having a jump on its maximum and will be useful to state asymptotic results of the type $f(x_{\hat{i}_n}) \to \sup_{x \in \mathcal{X}} f(x)$ when $n \to \infty$.

Definition 4.6 (Regular Level Sets)**.** *A function $f : \mathcal{X} \to \mathbb{R}$ has (c_α, α)-regular level sets for some $c_\alpha > 0$ and $\alpha \geqslant 0$ when:*

- *The global optimizer $x^\star \in \mathcal{X}$ is unique.*

Algorithm 14: RankOpt($\mathcal{R}$)

$\mathcal{R}_0 \leftarrow \mathcal{R}$
for $n = 0, 1, \ldots$ **do**
  done $\leftarrow \bot$
  while $\neg$ done **do**
    $x_{n+1} \leftarrow \mathcal{U}(\mathcal{X})$
    if $\exists r \in \mathcal{R}_n,\ r(x_{n+1}, x_{\hat{\imath}_n}) \geqslant 0$ **then**
      $y_{n+1} \leftarrow$ **Query**(x_{n+1})
      $\mathcal{R}_{n+1} \leftarrow \{r \in \mathcal{R} : L_{n+1}(r) = 0\}$
      done $\leftarrow \top$
    end
  end
end

- *The iso-level sets* $f^{-1}(y) \triangleq \{x \in \mathcal{X} : f(x) = y\}$ *satisfy:*

$$\sup_{x \in f^{-1}(y)} \|x^\star - x\|_2 \leqslant c_\alpha \inf_{x \in f^{-1}(y)} \|x^\star - x\|_2^{1/(1+\alpha)}.$$

Condition 4.6 guarantees that the points associated with high evaluations are close to the unique optimizer with respect to the Euclidean distance. This condition will be used to derive some finite-time bounds on the distance $\|x^\star - x_{\hat{\imath}_n}\|_2$ between the optimizer and its estimation. Note that for any iso-level set $f^{-1}(y)$ with bounded distance to the optimum, the condition is satisfied with $\alpha = 0$ and $c_\alpha \triangleq \Delta(\mathcal{X}) / \inf_{x \in f^{-1}(y)} \|x^\star - x'\|_2$. Therefore, this condition concerns the behavior of the level sets when $\inf_{x \in f^{-1}(y)} \|x^\star - x\|_2 \to 0$. As an example, the iso-level sets of two simple functions satisfying the condition with different values of α are shown in Figure 4.2.

4.3 Optimization with Fixed Ranking Structure

In this section, we consider the problem of optimizing an unknown function f given the prior knowledge that its ranking r_f belongs to a given ranking structure $\mathcal{R} \subseteq \mathcal{R}_\infty$.

4.3.1 The RankOpt Algorithm

The input of Algorithm 14 is a ranking structure $\mathcal{R} \subseteq \mathcal{R}_\infty$. At each iteration n, a point x_{n+1} is uniformly sampled over $\mathcal{X}$ until the algorithm decides to evaluate the function at this point.

Active Subset of Ranking Rules

The decision rule involves the active subset of $\mathcal{R}$ which contains the ranking rules that are consistent with the ranking rule induced by f over the points sampled so far. We thus set:

$$\mathcal{R}_n \triangleq \{r \in \mathcal{R} : L_n(r) = 0\},$$

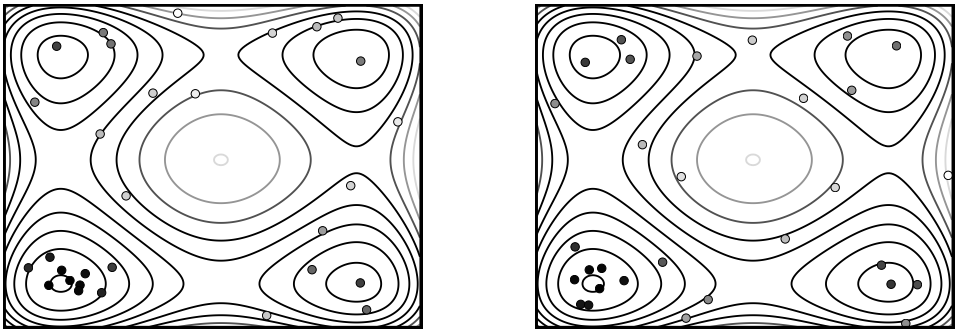


Figure 4.3. – Two samples generated by the RankOpt algorithm after $n = 30$ iterations with the polynomial ranking rules $\mathcal{R}_{\mathcal{P},4}$ on the Styblinski-Tang function defined in Section 4.6.

where L_n is the empirical ranking loss:

$$L_n(r) \triangleq \frac{2}{n(n+1)} \sum_{1 \leqslant i < j \leqslant n} \mathbb{1}\{r_f(x_i, x_j) \neq r(x_i, x_j)\}.$$

This set contains the ranking rules that are still on the run to be the true ranking rule. Indeed, it contains the rankings that perfectly rank the sample and we know, by definition of the empirical ranking loss, that the true ranking of the unknown function satisfies $r_f \in \mathcal{R}_n$ since $L_n(r_f) = 0$. As a direct consequence of the definition of the active subset, if there does not exist any $r \in \mathcal{R}_n$ such that $r(x_{n+1}, x_{\hat{i}_n}) \geqslant 0$, it implies that $r_f(x_{n+1}, x_{\hat{i}_n}) = -1$ which means that $f(x_{n+1}) < f(x_{\hat{i}_n})$. Thus, the algorithm never evaluates the function at a point that will not return certainly an evaluation at least equal to the highest evaluation $f(x_{\hat{i}_n})$ observed so far.

Properties and Connection to Active Learning

The sampling strategy arising from Algorithm 14 can be interpreted as follows. After n iterations the RankOpt algorithm evaluates the function on a sequence $\{x_i\}_{i=1}^n$ distributed as:

$$\begin{aligned} x_1 &\sim \mathcal{U}(\mathcal{X}), \\ x_{n+1} \,|\, \{(x_i, y_i)\}_{i=1}^n &\sim \mathcal{U}(\mathfrak{R}_n), \\ \text{where } \mathfrak{R}_n &\triangleq \Big\{ x \in \mathcal{X} : \exists r \in \mathcal{R}_n \text{ s.t. } r(x, x_{\hat{i}_t}) \geqslant 0 \Big\}. \end{aligned}$$

The subspace $\mathfrak{R}_n$ is similar to the relevant region used in the previous chapter. It is the smallest set that contains certainly the level set $\{x \in \mathcal{X} : f(x) \geqslant f(x_{\hat{i}_t})\}$ of the best value observed so far. The set of the maximizers $\operatorname{argmax}_{x \in \mathcal{X}} f(x)$ always belong to $\mathfrak{R}_n$.

The algorithm can be seen as an extension to ranking of the active learning algorithm CAL (Cohn et al., 1994; Hanneke, 2011). However, this algorithm aim at estimating a classifier $c : \mathcal{X} \to \{0, 1\}$ where the goal in global optimization is to estimate the winner of a tournament deriving from the ranking rule $r_f : \mathcal{X} \times \mathcal{X} \to \{-1, 0, 1\}$ and not the ranking rule itself.

4.3.2 Convergence analysis

We state here some convergence properties of the RankOpt algorithm. The results are stated in a probabilistic framework. The source of randomness comes from the algorithm itself

(which generates uniform random variables) and not from the evaluations which are assumed noiseless.

Consistency

The next result will be important in order to formulate the consistency property of the algorithm.

Lemma 4.2 (Random Search Comparison)**.** *Fix any $n \in \mathbb{N}$ and let $\mathcal{R} \subseteq \mathcal{R}_\infty$ be any set of ranking rules. Then, for any function $f : \mathcal{X} \to \mathbb{R}$ such that $r_f \in \mathcal{R}$ and any $y \in \mathbb{R}$, if $x_{\hat{\imath}_n}$ denotes the random output of RankOpt($\mathcal{R}$) on f, we have that,*

$$\mathbb{P}\big[f(x_{\hat{\imath}_n}) \geqslant y\big] \geqslant \mathbb{P}\Big[\max_{i=1\ldots n} f(x'_i) \geqslant y\Big],$$

where $x'_i \overset{iid}{\sim} \mathcal{U}(\mathcal{X})$.

Proof. The result is obtained by induction. Since $x_1 \sim \mathcal{U}(\mathcal{X})$, the result trivially holds for $n = 1$. Assume that the statement holds for a given $n > 0$, fix any $y \in \mathbb{R}$ and let

$$\mathcal{X}_{\geqslant y} \triangleq \big\{x \in \mathcal{X} : f(x) \geqslant y\big\}.$$

Using the fact that,

$$\big\{x \in \mathcal{X} : f(x) \geqslant f(X_{\hat{\imath}_n})\big\} \subseteq \mathfrak{R}_n \subseteq \mathcal{X},$$

we show that,

$$\mathbb{P}\big[f(x_{\hat{\imath}_{n+1}}) \geqslant y\big] \geqslant \mathbb{P}\big[f(x_{\hat{\imath}_n}) \geqslant y\big] + \mathbb{P}\big[f(x_{\hat{\imath}_n}) < y\big]\frac{\lambda(\mathcal{X}_{\geqslant y})}{\lambda(\mathcal{X})}.$$

Now plugging the induction assumption in the last equation and using the fact that:

$$\mathbb{P}\big[f(x'_{n+1}) \geqslant y\big] = \frac{\lambda(\mathcal{X}_{\geqslant y})}{\lambda(\mathcal{X})},$$

gives the result. □

Combining the previous proposition with the identifiability condition gives the following asymptotic result.

Corollary 4.1 (Consistency of RankOpt)**.** *Using the same notations and assumptions as in Lemma 4.2 and if the maximum of the function f is identifiable (Definition 4.5), then,*

$$f(x_{\hat{\imath}_n}) \to \sup_{x \in \mathcal{X}} f(x) \qquad \textit{in probability.}$$

Proof. Fix any $\varepsilon > 0$ and let

$$\mathcal{X}_\varepsilon \triangleq \Big\{x \in \mathcal{X} : f(x) \geqslant \sup_{x \in \mathcal{X}} f(x) - \varepsilon\Big\},$$

be the corresponding near-optimal level set. Applying Lemma 4.2 leads to

$$\mathbb{P}\Big[f(x_{\hat{\imath}_n}) < \sup_{x\in\mathcal{X}} f(x) - \varepsilon\Big] \leqslant \Big(1 - \frac{\lambda(\mathcal{X}_\varepsilon)}{\lambda(\mathcal{X})}\Big)^n \underset{n\to\infty}{\to} 0,$$

which proves the corollary. □

Corollary 4.1 reveals that the generic optimization scheme ends up finding the true maximum of any identifiable function that has a ranking in the given ranking structure $\mathcal{R}$.

Upper and Lower Bounds on the Loss

We now provide our finite-sample loss bounds.

Theorem 4.1 (Loss Upper Bound for RankOpt)**.** *Under the same assumptions as in Lemma 4.2 and if the function f has (c_α, α)-regular level sets (Definition 4.6), then, for any $u > 0$, with probability at least $1 - e^{-u}$,*

$$\|x^\star - x_{\hat{\imath}_n}\|_2 \leqslant C_\alpha \left(\frac{u}{n}\right)^{\frac{1}{d(1+\alpha)^2}},$$

where $C_\alpha \triangleq c_\alpha^{(2+\alpha)/(1+\alpha)} \Delta(\mathcal{X})^{1/(1+\alpha)^2}$.

Proof. Fix $u > 0$ and let $r_{n,u}$ be the upper bound of the theorem, and $r'_{n,u} \triangleq (r_{n,u}/c_\alpha)^{1+\alpha}$. Let us denote $x'_i \overset{\text{iid}}{\sim} \mathcal{U}(\mathcal{X})$ and $\mathcal{S}_\varepsilon \triangleq \big\{x \in \mathcal{X} : \|x^\star - x\|_2 = \varepsilon\big\}$ the sphere or radius ε centered in $x^\star$. We have

$$\begin{aligned}
\mathbb{P}\Big[\, \|x_{\hat{\imath}_n} - x^\star\|_2 \leqslant r_{n,u}\Big] &= \mathbb{P}\Big[x_{\hat{\imath}_n} \in \mathcal{B}(x^\star, r_{n,u})\Big] \\
&\geqslant \mathbb{P}\Big[f(x_{\hat{\imath}_n}) \geqslant \inf_{x\in\mathcal{S}_{r'_{n,u}}} f(x)\Big] \\
&\geqslant \mathbb{P}\Big[\max_{i\leqslant n} f(x'_i) \geqslant \inf_{x\in\mathcal{S}_{r'_{n,u}}} f(x)\Big],
\end{aligned}$$

where the first inequality comes from the regularity of the level sets and the second inequality comes from Lemma 4.2. Now, let $r''_{n,u} \triangleq (r'_{n,u}/c_\alpha)^{1+\alpha} = \Delta(\mathcal{X})\big(u/n\big)^{1/d}$. Applying once more the regularity assumption,

$$\begin{aligned}
\mathbb{P}\big[\, \|x_{\hat{\imath}_n} - x^\star\|_2 \leqslant r_{n,u}\big] &\geqslant \mathbb{P}\Big[\bigcup_{i\leqslant n} x'_i \in \mathcal{X} \cap \mathcal{B}(x^\star, r''_{n,u})\Big], \\
&= 1 - \left(1 - \frac{\lambda\big(\mathcal{X} \cap \mathcal{B}(x^\star, r''_{n,u})\big)}{\lambda(\mathcal{X})}\right)^n,
\end{aligned}$$

by independence of the x'_i. From Zabinsky and Smith (1992), we have that for all $r > 0$,

$$\begin{aligned}
\frac{\lambda\big(\mathcal{X} \cap \mathcal{B}(x^\star, r''_{n,u})\big)}{\lambda(\mathcal{X})} &\geqslant \Big(\frac{r''_{n,u}}{\Delta(\mathcal{X})}\Big)^d \\
&= \frac{u}{n}.
\end{aligned}$$

Finally, by classical comparison, $\mathbb{P}\big[\, \|x_{\hat{\imath}_n} - x^\star\|_2 \leqslant r_{n,u}\big] \geqslant 1 - e^{-u}$. □

More surprisingly, a lower bound can be derived by making the link with the theoretical PureAdaptiveSearch (Zabinsky and Smith, 1992) that uses the knowledge of the level sets of the unknown function.

Lemma 4.3 (PureAdaptiveSearch Comparison)**.** *Fix any $n > 0$ and let $\left(\widetilde{x}_n\right)_{i \leqslant n}$ be a sequence distributed as the Markov process defined by:*

$$\begin{aligned} \widetilde{x}_1 &\sim \mathcal{U}(\mathcal{X}) \\ \widetilde{x}_{n+1} \,|\, \widetilde{x}_n &\sim \mathcal{U}(\mathcal{X}_{\geqslant f(\widetilde{x}_n)}) \end{aligned}$$

Then, using the same notations and assumptions as in Lemma 4.2, for any $y \in \mathbb{R}$, we have that,

$$\mathbb{P}\big[f(x_{\hat{\imath}_n}) \geqslant y\big] \leqslant \mathbb{P}\big[f(\widetilde{x}_n) \geqslant y\big].$$

Proof. We use again an induction argument. Assume that the result holds for a given $n > 0$ and fix any $y \in \mathbb{R}$. Using the fact that $\mathcal{X}_{\geqslant f(X_{i_n})} \subseteq \mathfrak{R}_n$ we show that:

$$P\big[f(x_{\hat{\imath}_{n+1}}) \geqslant y\big] \leqslant \mathbb{E}\left[1 \wedge \frac{\lambda(\mathcal{X}_{\geqslant y})}{\lambda(\mathcal{X}_{\geqslant f(x_{\hat{\imath}_n})})}\right].$$

Using the induction hypothesis we show that:

$$\begin{aligned} \mathbb{E}\left[1 \wedge \frac{\lambda(\mathcal{X}_{\leqslant y})}{\lambda(\mathcal{X}_{\leqslant f(x_{\hat{\imath}_n})})}\right] &\leqslant \mathbb{E}\left[1 \wedge \frac{\lambda(\mathcal{X}_{\leqslant y})}{\lambda(\mathcal{X}_{\leqslant f(\widetilde{x}_n)})}\right] \\ &\leqslant \mathbb{P}\big[f(\widetilde{x}_{n+1}) \geqslant y\big]. \end{aligned}$$

□

We are now ready to establish our second loss bound by combining Lemma 4.3 with the level set assumption.

Theorem 4.2 (Loss Lower Bound for RankOpt)**.** *Under the same assumptions as in Lemma 4.2 and if the function f has (c_α, α)-regular level sets (Definition 4.6), then, for any $u > 0$, with probability at least $1 - e^{-u}$,*

$$C_\alpha e^{-\frac{(1+\alpha)^2}{d}\left(n+\sqrt{2nu}+u\right)} \leqslant \|x^\star - x_{\hat{\imath}_n}\|_2 \,,$$

where $C_\alpha \triangleq c_\alpha^{-(1+\alpha)(2+\alpha)} \operatorname{rad}(\mathcal{X})^{(1+\alpha)^2}$.

Proof. Let $r_{n,u}$ be the lower bound of the theorem. Using Lemma 4.3 and the level set assumption, with the same steps as before with $r'_{n,u} \triangleq c_\alpha r_{n,u}^{1/(1+\alpha)}$ and $r''_{n,u} \triangleq c_\alpha {r'_{n,u}}^{1/(1+\alpha)}$, we have:

$$\begin{aligned} \mathbb{P}\Big[\,\|x_{\hat{\imath}_n} - \widetilde{x}\|_2 \leqslant r_{n,u}\Big] &\leqslant \mathbb{P}\left[f(\widetilde{x}_n) \geqslant \inf_{x \in \mathcal{S}_{r'_{n,u}}} f(x)\right] \\ &\leqslant \mathbb{P}\left[\frac{\lambda(\mathcal{X}_{\geqslant f(\widetilde{x}_n)})}{\lambda(\mathcal{X})} \leqslant \frac{\lambda(\mathcal{B}(x^\star, r''_{n,u}))}{\lambda(\mathcal{X})}\right], \end{aligned}$$

$$\mathbb{P}\Big[\, \|x_{\hat{\imath}_n} - \widetilde{x}\|_2 \leqslant r_{n,u}\Big] \leqslant \mathbb{P}\left[\frac{\lambda(\mathcal{X}_{\geqslant f(\widetilde{x}_n)})}{\lambda(\mathcal{X})} \leqslant \left(\frac{r''_{n,u}}{\mathrm{rad}(\mathcal{X})}\right)^d\right]$$
$$\leqslant \mathbb{P}\left[\frac{\lambda(\mathcal{X}_{\geqslant f(\widetilde{x}_n)})}{\lambda(\mathcal{X})} \leqslant \exp\left(u - n - \sqrt{2nu}\right)\right].$$

Now, since

$$\forall u > 0,\ \mathbb{P}\left[\frac{\lambda(\mathcal{X}_{\geqslant f(\widetilde{x}_n)})}{\lambda(\mathcal{X})} \leqslant u\right] \leqslant \mathbb{P}\left[\prod_{i\leqslant n} U_i \leqslant u\right],$$

where $U_i \overset{\text{iid}}{\sim} \mathcal{U}([0,1])$, we have using concentration inequalities for sub-Gamma random variables that:

$$\mathbb{P}\left[\prod_{i\leqslant n} U_i \leqslant \exp\left(u - n - \sqrt{2nu}\right)\right] < e^{-u}.$$

□

The level set assumption, which is used in Theorem 4.1 and Theorem 4.2, is invariant to any strictly increasing composition h (i.e. if f has (c_α, α)-regular level sets so has $g = h \circ f$). It implies that the bounds on the distance $\|x^\star - x_{\hat{\imath}_n}\|_2$ between the exact solution and its approximation hold independently of the smoothness of the function. To the best of our knowledge, this is the first analysis of an optimization algorithm which uses the ranking rule induced by the unknown function.

4.4 Adaptive Algorithm and Stopping Time Analysis

We now consider the problem of optimizing a function f when no information is available on its ranking rule.

4.4.1 The AdaRankOpt Algorithm

The AdaRankOpt algorithm (Algorithm 15) is an extension of the RankOpt algorithm which involves model selection following the principle of Structural Risk Minimization. We consider a parameter $0 < p < 1$ and a nested sequence of ranking structures $\{\mathcal{R}_N\}_{N>0}$ satisfying:

$$\mathcal{R}_1 \subset \mathcal{R}_2 \subset \ldots \subset \mathcal{R}_\infty. \tag{4.1}$$

The algorithm is initialized by considering the smallest ranking structure $\mathcal{R}_1$ of the sequence. At each iteration n, with probability p the algorithm explores the space by evaluating the function at a point uniformly sampled over $\mathcal{X}$. Otherwise, the algorithm exploits the previous evaluations by making an iteration of the RankOpt algorithm with the smallest ranking structure $\mathcal{R}_{N_n}$ of the sequence that probably contains the true ranking r_f. Once a new evaluation has been made, the index N_{n+1} is updated. The parameter p drives the trade-off between the exploitation phase and the exploration phase which prevents the algorithm from getting stuck in a local maximum. Condition 4.1 is crucial for practical reasons discussed in Section 4.5. We point out that both the sequence of polynomial ranking rules $(\mathcal{R}_{\mathcal{P},N})_{N>0}$ and the sequence of convex ranking rules $(\mathcal{R}_{\mathcal{C},N})_{N>0}$ defined in Section 4.2 satisfy this condition.

Algorithm 15: AdaRankOpt$\big(p, (\mathcal{R}_N)_{N>0}\big)$

$N \leftarrow 1$
$\mathcal{R} \leftarrow \mathcal{R}_1$
for $n = 0, 1, \ldots$ **do**
  $B \leftarrow \mathcal{U}([0,1])$
  if $B < p$ **then**
    $x_{n+1} \leftarrow \mathcal{U}(\mathcal{X})$
  else
    $x_{n+1} \leftarrow \mathcal{U}\big(\{x \in \mathcal{X} : \exists r \in \mathcal{R},\ r(x, x_{\hat{\imath}_n}) \geqslant 0\}\big)$
  end
  $y_{n+1} \leftarrow$ **Query**(x_{n+1})
  $i_{\hat{\imath}_{n+1}} \leftarrow \operatorname{argmax}_{i \leqslant n+1} y_i$
  $N \leftarrow \min\big\{N > 0 : \min_{r \in \mathcal{R}_N} L_n(r) = 0\big\}$
  $\mathcal{R} \leftarrow \big\{r \in \mathcal{R}_N : L_n(r) = 0\big\}$
end

4.4.2 Theoretical Properties of AdaRankOpt

Consistency

We start by casting the consistency result for the AdaRankOpt algorithm.

Lemma 4.4 (Consistency of AdaRankOpt)**.** *Fix any $0 < p < 1$ and any sequence of ranking structures $(\mathcal{R}_N)_{N>0}$ satisfying Eq. 4.1. Then, if the function f has an identifiable maximum (Definition 4.5) and $x_{\hat{\imath}_n}$ denotes the random output of AdaRankOpt$(p, (\mathcal{R}_N)_{N>0})$ on f, we have that,*

$$f(x_{\hat{\imath}_n}) \to \sup_{x \in \mathcal{X}} f(x) \qquad \textit{in probability.}$$

Proof. Fix any $\varepsilon > 0$. Using the fact that

$$\mathbb{P}\big[X_i \in \mathcal{X}_\varepsilon\big] \geqslant p\frac{\lambda(\mathcal{X}_\varepsilon)}{\lambda(\mathcal{X})},$$

for any $i > 0$, we have by induction that

$$\mathbb{P}\Big[f(x_{\hat{\imath}_n}) < \sup_{x \in \mathcal{X}} f(x) - \varepsilon\Big] \leqslant \left(1 - p\frac{\lambda(\mathcal{X}_\varepsilon)}{\lambda(\mathcal{X})}\right)^n \to 0.$$

□

Lemma 4.4 reveals that even if the algorithm is poorly tuned, it will end up finding the true maximum of any function with an identifiable maximum.

Stopping Time and Rademacher Complexity

We now investigate the number of iterations required to identify a ranking structure that contains the true ranking rule.

Definition 4.7 (IDENTIFICATION STOPPING TIME). *Let* $N^\star \triangleq \min\{N > 0 : r_f \in \mathcal{R}_N\}$ *be the index of the smallest ranking structure that contains the true ranking rule. Let* $(N_n)_{n>0}$ *be the sequence of random variables defined in the AdaRankOpt algorithm. We define the stopping time:*

$$\tau_{N^\star} \triangleq \min\{n > 0 : N_n = N^\star\},$$

which corresponds to the number of iterations required to identify $N^\star$.

In order to bound $\tau_{N^\star}$ we need to control the complexity of the sequence of ranking structures. Let us denote by:

$$L(r) \triangleq \mathbb{P}\big[r_f(X, X') \neq r(X, X')\big]$$
$$X, X' \overset{\text{iid}}{\sim} \mathcal{U}(\mathcal{X}),$$

the true ranking loss, and define the Rademacher average of a ranking structure $\mathcal{R}$ as:

$$\widehat{R}_n \equiv \widehat{R}_n(\mathcal{R}) \triangleq \sup_{r\in\mathcal{R}} \frac{1}{\lfloor n/2\rfloor}\left|\sum_{i=1}^{\lfloor n/2\rfloor} \epsilon_i \mathbb{1}\Big\{r_f(X_i, X_{\lfloor n/2\rfloor+i}) \neq r(X_i, X_{\lfloor n/2\rfloor+i})\Big\}\right|$$

where $X_i \overset{\text{iid}}{\sim} \mathcal{U}(\mathcal{X})$ and $\epsilon_1 \dots \epsilon_{\lfloor n/2\rfloor}$ are $\lfloor n/2\rfloor$ independent Rademacher random variables.

Lemma 4.5 (STOPPING TIME UPPER BOUND). *Assume that the index* $N^\star > 1$ *is finite, and that* $\inf_{r\in\mathcal{R}_{N^\star-1}} L(r) > 0$*, and that there exists* $V > 0$ *such that the Rademacher complexity of* $\mathcal{R}_{N^\star-1}$ *satisfies:*

$$\forall n > 0,\ \mathbb{E}\big[\widehat{R}_n\big] \leqslant \sqrt{V/n}.$$

Then, for any $u > 0$*, with probability at least* $1 - e^{-u}$*,*

$$\tau_{N^\star} \leqslant \frac{10}{p}\left(\frac{V + u + \log 2}{\inf_{r\in\mathcal{R}_{N^\star-1}} L(r)^2}\right).$$

Proof. Fix any $u > 0$ and let n_u be the integer part of the upper bound of the proposition. Since we have a nested sequence of sets of ranking rules,

$$\mathbb{P}\big[\tau \leqslant n_u\big] = \mathbb{P}\Big[\min_{r\in\mathcal{R}_{N^\star-1}} L_{n_u}(r) > 0\Big].$$

Let us write $n'_u \triangleq \Big\lfloor pn_\delta - \sqrt{\frac{1}{2}n_u(u + \log 2)}\Big\rfloor$. Using Hoeffding's inequality gives a lower bound on the number of exploration samples collected:

$$\mathbb{P}\left[\sum_{i\leqslant n_\delta} B_i \geqslant n'_u\right] \geqslant 1 - \tfrac{1}{2}e^{-u},$$

where $B_i \overset{\text{iid}}{\sim} \mathcal{B}(p)$. Applying concentration results of ranking rules over the n'_u exploration samples gives that:

$$\mathbb{P}\left[\min_{r\in\mathcal{R}_{N^\star-1}} L_{n'_u}(r) \geqslant \min_{r\in\mathcal{R}_{N^\star-1}} L(r) - 2\sqrt{\frac{V}{n'_u}} - 2\sqrt{\frac{u + \log 2}{n'_u - 1}} > 0\right] \geqslant 1 - \tfrac{1}{2}e^{-u}.$$

□

Upper Bound on the Loss

In the situation described above, one can recover an upper bound similar to the one of Theorem 4.1 using the fact that the ranking structure $\mathcal{R}_{N^\star}$ can be identified in a finite time and by combining the previous result with the analysis of the RankOpt algorithm where the structure $\mathcal{R}_{N^\star}$ is assumed to be known.

Theorem 4.3 (Loss Upper Bound for AdaRankOpt)**.** *Consider the same assumptions as in Lemma 4.5 and assume that the function f has (c_α, α)-regular level sets (Definition 4.6). Then, if $x_{\hat{\imath}_n}$ denotes the random output of AdaRankOpt$(p, (\mathcal{R}_N)_{N>0})$, for any $u > 0$ and any $n > n_u$, with probability at least $1 - e^{-u}$,*

$$\|x^\star - x_{\hat{\imath}_n}\|_2 \leqslant C_\alpha \left(\frac{u + \log 2}{n - n_u} \right)^{\frac{1}{d(1+\alpha)^2}},$$

where C_α is the same constant as in Theorem 4.1 and:

$$n_u \triangleq \left\lfloor 10 \frac{V + u + \log 4}{p \inf_{r \in \mathcal{R}_{N^\star - 1}} L(r)^2} \right\rfloor.$$

Proof. Fix any $u > 0$. We know that after n_u iterations the true ranking structure $\mathcal{R}_{N^\star}$ is identified with probability at least $1 - \frac{1}{2}e^{-u}$ by Lemma 4.5. Once the structure $\mathcal{R}_{N^\star}$ is identified, one can use a similar technique as the one used in Theorem 4.1 to get an upper bound with probability at least $1 - \frac{1}{2}e^{-u}$ thanks to the $n - n_u$ samples. □

We point out that standard metric entropy arguments can be used in order to bound $\mathbb{E}\big[\widehat{R}_n\big]$ (Agarwal et al., 2005; Clémençon et al., 2008; Clémençon, 2011). If the class of functions $\big\{x, x' \in \mathcal{X} \mapsto \mathbb{1}\{r_f(x, x') \neq r(x, x')\} : r \in \mathcal{R}\big\}$ is a VC-major class with finite VC-dimension V, then for a universal constant $c > 0$,

$$\mathbb{E}\big[\widehat{R}_n\big] \leqslant c\sqrt{V/n}.$$

This covers the case of polynomial ranking rules, and we refer to Boucheron et al. (2005) for similar inequalities for nonparametric classes such as kernel machines.

4.5 Computational Aspects

We discuss here some technical aspects involved in the practical implementation of the AdaRankOpt algorithm.

4.5.1 General ranking structures

Fix any nested sequence of ranking structures $(\mathcal{R}_N)_{N>0}$ and any sample $\{(x_i, f(x_i))\}_{i \leqslant n}$. We first address the questions of sampling x_{n+1} uniformly over the non-trivial subset

$$\mathfrak{R}_n = \Big\{ x \in \mathcal{X} : \exists r \in \mathcal{R}_{N_n} \text{ s.t. } L_n(r) = 0 \text{ and } r(x, x_{\hat{\imath}_n}) \geqslant 0 \Big\},$$

and second updating the index N_{n+1} once $f(x_{n+1})$ has been evaluated. We start to show that both these steps can be done by testing if:

$$\min_{r \in \mathcal{R}_N} L_{n+1}(r) = 0 \tag{4.2}$$

holds true for a given $N > 0$ where the empirical ranking loss is taken over a set of $n+1$ samples.

Uniform Sampling in the Relevant Region

Sampling $x \sim \mathcal{U}(\mathcal{X})$ until $x \in \mathfrak{R}_n$ allows to sample uniformly over $\mathfrak{R}_n$. Using the definition of the subset, we know that $x \in \mathfrak{R}_n$ if there exists a ranking $r \in \mathcal{R}_{N_n} \cap \{r : L_n(r) = 0\}$ such that $r(x, x_{\hat{i}_n}) \in \{0, 1\}$. Rewriting the previous statement in terms of minimal error gives that $x \in \mathfrak{R}_n$ if:

- either $\min_{r \in \mathcal{R}_{N_n}} L_{n+1}(r) = 0$, where L_{n+1} is taken over the sample:

$$\left\{(x_i, f(x_i))\right\}_{i \leqslant n} \cup \left\{(x, f(x_{\hat{i}_n}))\right\};$$

- or $\min_{r \in \mathcal{R}_{N_n}} L_{n+1}(r) = 0$ where L_{n+1} is taken over the sample:

$$\left\{(x_i, f(x_i))\right\}_{i \leqslant n} \cup \left\{(x, f(x_{\hat{i}_n}) + \varepsilon)\right\},$$

  and ε is any strictly positive constant.

Updating the Index

Now, assume that $f(x_{n+1})$ has been evaluated. Since $\{\mathcal{R}_N\}_{N>0}$ forms a nested sequence, we have that:

$$N_{n+1} = N_n + \min\left\{i > 0 : \min_{r \in \mathcal{R}_{N_n+i}} L_{n+1}(r) = 0\right\},$$

where the empirical loss is taken over:

$$\left\{(x_i, f(x_i))\right\}_{i \leqslant n+1}.$$

Therefore, N_{n+1} can be updated by sequentially testing if:

$$\min_{r \in \mathcal{R}_{N_n+i}} L_{n+1}(r) = 0,$$

for $i = 0, 1, 2 \ldots$.

As mentioned earlier, both the previous steps can be done using a generic procedure that tests if Eq. 4.2 holds true.

4.5.2 Practical Solutions for Particular Ranking Structures

We now provide some equivalences that can be used to design this procedure for the ranking structures introduced in Section 4.2. For simplicity, we assume that all the evaluations of the sample are distinct:

$$f(x_{(1)}) < f(x_{(2)}) < \ldots < f(x_{(n+1)}), \tag{4.3}$$

where $(1)\dots(n+1)$ denote the indexes of the corresponding reordering.

Polynomial ranking rules

Consider the sequence of polynomial ranking rules $(\mathcal{R}_{\mathcal{P},N})_{N>0}$ and let $\phi_N : \mathbb{R}^d \to \mathbb{R}^{D(d,N)}$ be the function that maps any point of $\mathbb{R}^d$ into the polynomial feature space of degree N where $D(d,N) \triangleq \binom{N+d}{d} - 1$. For example, $\phi_2(x_1,x_2) = \left(x_1, x_2, x_1x_2, x_1^2, x_2^2\right)$. We start by making the link with linear separability in the polynomial feature space.

Lemma 4.6 (Linear Separability)**.** *Fix any $N > 0$ and assume that all the evaluations are distinct. Then, Eq. 4.2 holds true if and only if there exists $h \in \mathbb{R}^{D(d,N)}$ such that,*

$$\forall i \leqslant n,\ \left\langle h, \phi_N(x_{(i+1)}) - \phi_N(x_{(i)})\right\rangle > 0.$$

Proof. The proof is a consequence of the definition of polynomial ranking rules: if $r \in \mathcal{R}_{\mathcal{P},N}$ then there exists $h \in \mathbb{R}^{D(d,N)}$ and $c \in \mathbb{R}$ such that:

$$\begin{aligned} r(x,x') &= \operatorname{sgn}\left(h^\top \phi_N(x) + c - h^\top \phi_N(x') - c\right) \\ &= \operatorname{sgn}\left(h^\top\left(\phi_N(x) - \phi_N(x')\right)\right). \end{aligned}$$

Noticing that for all $i \leqslant n$ we have $r_f(x_{(i+1)}, x_{(i)}) = 1$, gives the result. □

Interestingly, testing the linear separability of a sample is equivalent to testing the emptiness of a sample-dependent polyhedron.

Corollary 4.2 (Emptiness of a Polyhedron)**.** *Let $\mathbf{M}_N$ be the $(D(d,N), n)$-matrix where its i-th column is equal to:*

$$\left[\mathbf{M}_N\right]_{\cdot,i} = \left(\phi_N(x_{(i+1)}) - \phi_N(x_{(i)})\right)^\top,$$

and let the $\succeq$ operator stands for the component-wise inequality. Then, under the same assumptions as in Lemma 4.6, Eq. 4.2 holds true if and only if the following polyhedron is empty:

$$\left\{\mathbf{v} \in \mathbb{R}^n : \mathbf{M}_N\mathbf{v} = \mathbf{0},\ \mathbf{1}^\top\mathbf{v} = 1,\ \mathbf{v} \succeq \mathbf{0}\right\} = \varnothing.$$

Proof. For any $i \leqslant n$ let $X_i \triangleq \phi_N(x_{(i+1)}) - \phi_N(x_{(i)})$ and let $\operatorname{conv}\left(\{X_i\}_{i\leqslant n}\right)$ be the convex hull of $\{X_i\}_{i\leqslant n}$. First, we have by convexity the following equivalence:

$$\forall i \leqslant n, \exists h \in \mathbb{R}^{D(d,N)},\ h^\top X_i > 0 \iff \mathbf{0} \notin \operatorname{conv}\left(\{X_i\}_{i\leqslant n}\right).$$

Then using the definition of convex hull we get that:

$$\mathbf{0} \in \operatorname{conv}\left(\{X_i\}_{i\leqslant n}\right) \iff \exists \mathbf{v} \in \mathbb{R}^n,\ X^\top\mathbf{v} = \mathbf{0}, \mathbf{1}^\top\mathbf{v} = 1, \mathbf{v} \succeq 0.$$

□

We remark that the problem of testing the emptiness of a polyhedron can be seen as the problem of finding a feasible point of a linear program. We refer to Chapter 11.4 in Boyd and Vandenberghe (2004) where algorithmic solutions are discussed.

Convex ranking rules

Consider the sequence of convex ranking rules $\{\mathcal{R}_{\mathcal{C},N}\}_{N>0}$. Following the steps of Clémençon and Vayatis (2010) leads to the next equivalence.

Lemma 4.7 (Overlaying Classifiers)**.** *Fix any $N > 0$ and let $\mathcal{X} = [a, b]$. Then, Eq. 4.2 holds true if and only if there a exists a nested sequence $h_1 \geqslant h_2 \geqslant \ldots \geqslant h_{n+1}$ of $n+1$ classifiers of the form:*

$$h_i(x) = \sum_{k \leqslant N} \mathbb{1}\{l_{i,k} \leqslant x \leqslant u_{i,k}\},$$

satisfying for all $i, j \leqslant n+1$:

$$h_i(x_{(j)}) = \mathbb{1}\{(j) \geqslant i\}.$$

Proof. The implication from left to right a direct consequence of the definition of convex ranking rules. Now, assume that there exists such a nested sequence of classifiers $h_1 \geqslant \ldots \geqslant h_{n+1}$ satisfying the conditions. To state the reverse implication we build a continuous approximation of the step function:

$$h(x) = \sum_{i \leqslant n} h_i(x),$$

that perfectly ranks the sample and induces a convex ranking, for ε small enough:

$$\widehat{h}(x) = \sum_{i \leqslant n+1} \sum_{k \leqslant N} \phi(x, l_{i,k}, u_{i,k})$$

$$\text{where } \phi(x, l, u) = \begin{cases} 1 - \frac{l-x}{\varepsilon} & \text{if } x \in (l-\varepsilon, l) \\ 1 & \text{if } x \in (l, u) \\ 1 - \frac{x-u}{\varepsilon} & \text{if } x \in (u, u+\varepsilon) \\ 0 & \text{otherwise.} \end{cases}$$

By construction we have that $L_{n+1}(r_{\widehat{h}}) = L_{n+1}(r_h) = 0$ and $r_{\widehat{h}} \in \mathcal{R}_{\mathcal{C},N}$. □

The problem of overlaying classifiers admits a tractable solution when $d = 1$. In the specific case where $N = 1$ and $d \geqslant 1$, the problem of testing the existence of nested convex classifiers is equivalent to the problem of testing the emptiness of a cascade of polyhedrons.

Lemma 4.8 (Emptiness of a Cascade of Polyhedrons)**.** *Fix any $d \geqslant 1$, let $N = 1$ and assume that all the evaluations are distinct. Then, Eq. 4.2 holds true if and only if for each $k \leqslant n$ the following polyhedron is empty:*

$$\left\{\mathbf{v} \in \mathbb{R}^k : \mathbf{M}_k \mathbf{v} = x_{(n+1-k)}, \mathbf{1}^\top \mathbf{v} = 1, \mathbf{v} \geq \mathbf{0}\right\},$$

where $\mathbf{M}_k$ is the (d, k)-matrix where its i-th column is equal to $x_{(n+2-i)}^\top$.

Proof. Using the definition of convex hulls, we show that each polyhedron of the cascade is empty if and only if:

$$\text{conv}\left(\{x_{(i)}\}_{i=n}^{n+1}\right) \subset \text{conv}\left(\{x_{(i)}\}_{i=n-1}^{n+1}\right) \subset \ldots \subset \text{conv}\left(\{x_{(i)}\}_{i=1}^{n+1}\right).$$

Like above, we can build a continuous approximation of the function

$$h(x) = \sum_{k \leqslant n} \mathbb{1}\big\{x \in \mathrm{conv}(\{x_{(i)}\}_{k \leqslant i \leqslant n+1})\big\},$$

which has a convex ranking rule and perfectly ranks the sample. □

4.6 Experiments

We now assess the empirical performances of the AdaRankOpt algorithm on several benchmarks functions and compare the results to other deterministic optimization algorithms.

4.6.1 Protocol of the Empirical Assessment

Parameters of AdaRankOpt

The tuning parameters of the competitors were set to default and the parameter p of AdaRankOpt was set to $1/4$ for the convex ranking rules and to $1/10$ for the polynomial ranking rules. We consider three synthetic problems:

Optimization Objectives

1. This task consists in maximizing the function:

$$f(x) = \tfrac{1}{10}\mathbb{1}\{x \leqslant x^\star\}\Big(\big|\cos(50(x-x^\star))\big|^{3/2} - 15\big|x-x^\star\big|^{1/2}\Big) - \mathbb{1}\{x > x^\star\}\Big(|x|^{1/2} + \tfrac{1}{20}\big|\sin(50x)\big|^{3/2}\Big),$$

   over $\mathcal{X} = [0,1]$ where $x^\star = 0.499$. The function f has 17 local maxima and presents a discontinuity around its unique optimizer $x^\star$. The convex ranking rules were used.

2. This task consists in minimizing a perturbed version of the Styblinski-Tang function:

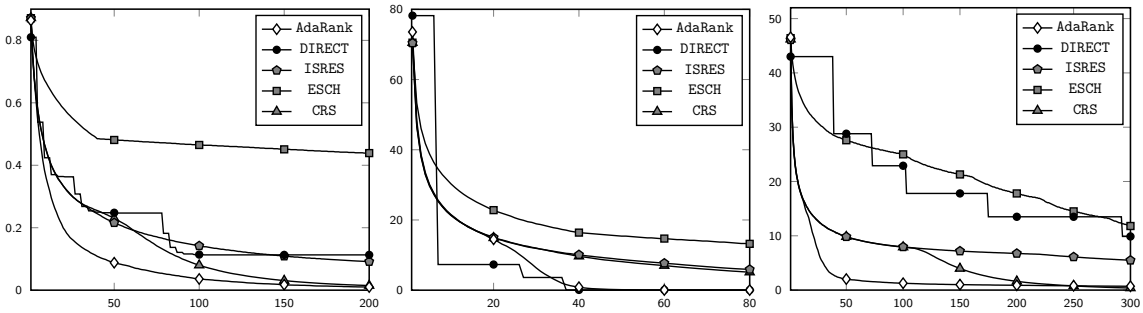
$$f(x) = \sum_{i=1}^{2}(x_i^4 - 16x_i^2 + 5x_i)/2 + \cos(x_1 + x_2),$$

   over $\mathcal{X} = [-5,5]^2$. The level sets of the Styblinski-Tang function are displayed on Figure 4.3 and the function has 4 local optima. The polynomial ranking rules were used.

3. This task consists in maximizing the function:

$$f(x) = 1 - \left|\frac{1}{10}\sum_{i=1}^{10}(x_i - 4.5)\right|^{5/2},$$

   over $\mathcal{X} = [-5,5]^{10}$. The hyperplane $\Big\{x \in \mathbb{R}^{10} : \sum_{i=1}^{10} x_i = 45\Big\}$ maximizes the function. The polynomial ranking rules were used.



1. Discontinuous ($d = 1$) 2. Styblinski ($d = 2$) 3. Convex ($d = 10$)

Figure 4.4. – Empirical estimation of $\max_{x \in \mathcal{X}} f(x) - \mathbb{E}[f(x_{\hat{i}_t})]$ in terms of iteration n where the expectation was obtained by running a 1000 times each algorithm.

4.6.2 Empirical Comparisons

We compared the performances of the AdaRankOpt against four global optimization algorithms taken from the NLOpt library (Johnson, 2014): The results are shown in Figure 4.4. The plots present the values of the approximation of the expected simple regret:

$$\mathbb{E}\big[S_n\big] = \sup_{x^\star \in \mathcal{X}} f(x^\star) - \mathbb{E}\big[f(x_{\hat{i}_n})\big],$$

for each iteration n, where the expectation was obtained by running 1000 times each algorithm. We remark that the AdaRankOpt converges fast and avoids falling in local maxima, as opposed to most of its competitors.

Controlled Random Search

The CRS algorithm from Kaelo and Ali (2006) is a controlled random search with local mutations. It starts with a random population and randomly evolve these points by an heuristic rule. This strategy consistently find the maximum of the objective, but we observed that it requires more evaluations than AdaRankOpt to attain a given simple regret.

Lipschitz Optimization

The DIRECT algorithm from Jones et al. (1993) is a Lipschitz optimization algorithm where the Lipschitz constant is unknown. It uses partitioning techniques of the search space. Since the splitting threshold are deterministic, the curve from Figure 4.4 are not smoothed by the empirical expectation. We see on Figure 4.4 that the performances of this algorithm are not consistent, in that it sometimes requires an order of magnitude more iteration than AdaRankOpt to obtain a given simple regret.

Evolutionary Algorithms

The ESCH algorithm from da Sliva Santos et al. (2010) and the ISRES algorithm from Runarsson and Yao (2000) are two evolutionary algorithms. The evolution strategies are based on a combination of mutation rules and differential variations. The drawback of evolutionary algorithms is that they typically use lots of queries to maintain a "population" of candidates, which results in slow convergence.

4.7 Conclusion and Discussion

We have provided a global optimization strategy based on a sequential estimation of the ranking of the unknown function. We introduced two algorithms: RankOpt which requires a prior knowledge of the ranking rule of the unknown function and its adaptive version AdaRankOpt which performs model selection. A theoretical analysis is provided and the adaptive algorithm is shown to be empirically competitive with the state-of-the-art methods on representative synthetic examples. To the best of our knowledge, this is the first approach of this nature. Possible future research directions include extension of the algorithms to noisy evaluations and characterization of the class of functions that attain the exponential rate presented in the lower bound.

Applications and Implementation Details

5

As stated in the introduction, the research presented in this dissertation has been motivated by specific applications. In Section 5.1, we review practical and numerical tools required for efficient implementations of the proposed algorithms. We then present, in Section 5.2, two physical applications where we leverage these algorithms to solve research questions from various domains. The first one relates to tsunamis analysis, and is a joint work with Themistoklis Stefanakis published in Stefanakis et al. (2014). The second one relates to wave energy converters, and is a joint work with Dripta Sarkar published in Sarkar et al. (2015) and Sarkar et al. (2016). In Section 5.3, we describe applications for model calibration and we present a joint work with Fabien Cailliez and Pascal Pernot on force field calibration.

Contents

5.1 Efficient Computations and Software Library for Bayesian Optimization **122**
5.1.1 Bayesian Inference 122
Computation of Posterior Distributions 122
Iterative Updates of Posteriors 122
5.1.2 Gaussian Process Prior Selection and Validation 123
Leave-One-Out Posterior Likelihood 123
Efficient Computations of the Likelihoods 123
5.1.3 Non-Gaussian Processes 124
Computing Confidence Bounds with the Cramer-Chernoff Method 124
Numerical Inversion with Newton's Method 125
5.1.4 Software Library Release 125
5.2 Applications for Physical Simulations **125**
5.2.1 Tsunamis Amplification Phenomenon 125
Maximization of the Output of Tsunamis Simulations 126
Experimental Configuration 126
A Novel Stopping Criterion with Ranking 127
The Effect of the Island 128
5.2.2 Wave Energy Converters 129
Optimization of Spatial Layout of an Array of WECs 129
Methodology and Bayesian Inference 130
Optimization of the Predictions with Genetic Algorithms 131
Optimal Predicted WEC Arrays 132
Conclusion 132
5.3 Applications to Model Calibration **133**
5.3.1 Calibration of Force Fields Parameters for Molecular Simulation 133
Computational Chemistry Methodology 133
Global Optimization of Quadratic Forms 133
Experimental Results 134
5.3.2 Hyper-Parameter Optimization and Further Perspectives 134
Squared Gaussian Processes 135
Simple Ranking Structures in High Dimension 135

5.1 Efficient Computations and Software Library for Bayesian Optimization

In this section, we describe computational techniques and implementation details for the Bayesian optimization algorithms presented in Chapter 3.

5.1.1 Bayesian Inference

The computational complexity of Bayesian inference, from Eq. 2.14 and Eq. 2.16, is $\mathcal{O}(n^3)$, where n is the number of observations. When not implemented carefully this may create a computational barrier even for problems of medium size.

Computation of Posterior Distributions

Let us recall the following notations: $\mathbf{Y}_n$ is the vector of observations at points X_n, k the kernel of the Gaussian process, and η^2 the variance of the noise. We write $\mathbf{K}_n$ for the square matrix of kernel evaluations between the points in X_n. The computational cost of the Bayesian inference is driven by the inversion of the kernel matrix plus a diagonal of noise variance,

$$\mathbf{C}_n \triangleq \mathbf{K}_n + \eta^2 \mathbf{I}.$$

Since this matrix is positive semi-definite by definition, we heavily rely on Cholesky decomposition (Stewart, 1998), that is we find an upper triangular matrix $\mathbf{U}_n$ such that:

$$\mathbf{U}_n^\top \mathbf{U}_n = \mathbf{C}_n.$$

The inversion is then replaced by the direct resolutions of two triangular systems, in $\mathcal{O}(n^2)$:

$$\mathbf{C}_n^{-1} \mathbf{Y}_n = \mathbf{U}_n^{-1} \mathbf{U}_n^{-\top} \mathbf{Y}_n.$$

Cholesky decomposition is faster and more stable than normal inversion (Press, 2007). If $\eta = 0$ and the kernel is degenerate, like a linear or polynomial kernel, $\mathbf{C}_n$ might not be strictly positive-definite and available algorithms may fail. To alleviate this constraint, we numerically approximate the limit of the Cholesky decompositions:

$$\mathbf{U}_{n,i}^\top \mathbf{U}_{n,i} = \mathbf{C}_n + i^{-1} \mathbf{I},$$

when $i \to \infty$. The sequence of $\mathbf{U}_{n,i}$ converges to $\mathbf{U}_{n,\infty}$, and the limit satisfies the required property:

$$\mathbf{U}_{n,\infty}^\top \mathbf{U}_{n,\infty} = \mathbf{C}_n.$$

Iterative Updates of Posteriors

The computational complexity of Cholesky decomposition is still $\mathcal{O}(n^3)$. Fortunately, the cost to compute $\mathbf{U}_{n+1}$ knowing $\mathbf{U}_n$ is only $\mathcal{O}(n^2)$. Therefore, Bayesian optimization algorithm may do only updates of the posterior at each iterations. Using the block notation,

$$\mathbf{C}_{n+1} = \begin{bmatrix} \mathbf{C}_n & \mathbf{C}_{n,n+1} \\ \mathbf{C}_{n,n+1}^\top & \mathbf{C}_{n+1,n+1} \end{bmatrix},$$

we give here the update formulae for the Cholesky decomposition (Stewart, 1998):

$$\begin{aligned}\mathbf{U}_{n+1} &= \begin{bmatrix}\mathbf{U}_n & \mathbf{V}\\ \mathbf{0} & \mathbf{Z}\end{bmatrix},\\ \text{where } \mathbf{V} &\triangleq \mathbf{U}_n^{-\top}\mathbf{C}_{n,n+1},\\ \text{and } \mathbf{Z} &\triangleq \text{cholesky}(\mathbf{C}_{n+1,n+1} - \mathbf{V}^\top\mathbf{V}).\end{aligned}$$

These formulae adapt to the case where we update the posterior after a batch of K observations. In that case the computational complexity is $\mathcal{O}(n^2K^3)$.

5.1.2 Gaussian Process Prior Selection and Validation

As mentioned in Section 2.1.5, the parameters θ of the prior Gaussian process distribution and the noise variance are typically not known. A common method to select these parameters is to minimize the negative posterior log-likelihood:

$$\mathcal{L}_n(\theta) \triangleq \tfrac{1}{2}\mathbf{Y}_n^\top\mathbf{C}_{n,\theta}^{-1}\mathbf{Y}_n + \tfrac{1}{2}\log|\mathbf{C}_{n,\theta}| + \tfrac{n}{2}\log 2\pi.$$

Full-Bayesian perspectives where one puts prior on the parameters θ is often not a practical approach for Bayesian optimization since the computational burden is prohibitive.

Leave-One-Out Posterior Likelihood

One drawback of optimizing the likelihood is that it is prone to overfitting for general priors. The popular solution in machine learning is to perform cross-validations (Hastie et al., 2008). We consider here the simplest case of cross-validation, the leave-out-out validation, where we perform the above step with all single observations successively hidden. That is, we try to minimize the loss:

$$\mathcal{L}_n^{\text{loo}}(\theta) \triangleq \sum_{i=1}^{n}\left(\tfrac{1}{2}\log\sigma^2_{n\backslash i,\theta}(x_i) + \left(2\sigma^2_{n\backslash i,\theta}(x_i)\right)^{-1}\left(y_i - \mu_{n\backslash i,\theta}(x_i)\right)^2\right), \tag{5.1}$$

where $\mu_{n\backslash i,\theta}$ and $\sigma^2_{n\backslash i,\theta}$ are respectively the posterior expectation and variance using the n observations but the i-th. In general, the $\mathcal{L}_n^{\text{loo}}$ function may not be convex. Nevertheless, its values can be computed quickly together with its gradients, and the minimization is commonly performed with gradient-based algorithm like the BFGS algorithm or the ConjugateGradient descent (Press, 2007).

Efficient Computations of the Likelihoods

As before, the value of $\mathcal{L}_n$ can be efficiently obtained by Cholesky decomposition of $\mathbf{C}_n$, since both the quadratic form and the determinant are easily extracted thereafter. An efficient way to get the values of $\mathcal{L}_n^{\text{loo}}$ is to first compute the Cholesky decomposition of $\mathbf{C}_n$ using all the observations, and then successively extract the decompositions with one hidden observation. Removing one observation requires a rank-one update of the decomposition. Using the block notation:

$$\mathbf{U}_n = \begin{bmatrix}\mathbf{U}_{n,i-1} & \mathbf{U}_{n,i-1,i} & \mathbf{U}_{n,i-1,i+1}\\ \mathbf{U}_{n,i-1,i}^\top & \mathbf{U}_{n,i} & \mathbf{U}_{n,i,i+1}\\ \mathbf{U}_{n,i-1,i+1}^\top & \mathbf{U}_{i,i+1}^\top & \mathbf{U}_{n,i+1}\end{bmatrix},$$

the formulae to downgrade the Cholesky decomposition is:

$$\mathbf{U}_{n\setminus i} = \begin{bmatrix} \mathbf{U}_{n,i-1} & \mathbf{U}_{n,i-1,i+1} \\ \mathbf{0} & \mathbf{V} \end{bmatrix},$$
$$\text{where } \mathbf{V} = \text{cholupdate}(\mathbf{U}_{n,i+1}, \mathbf{U}_{n,i,i+1}),$$

where $\text{cholupdate}(\mathbf{U}, \mathbf{Z})$ returns in $\mathcal{O}(n^2)$ the decomposition of $\mathbf{U} + \mathbf{Z}^\top \mathbf{Z}$.

5.1.3 Non-Gaussian Processes

For Gaussian processes, deriving confidence intervals is easy thanks to closed formulae for the Bayesian inference. In the case of other stochastic processes, like the one presented in Section 3.3.2, such closed formulae may be hard to derive. In that case, we fall back to the numerical inversion of one-dimensional functions involved in the Cramer-Chernoff method. This techniques may be used both in the computation of the smoothness $\ell(\cdot, \cdot)$ from Eq. 3.19 and the upper and lower confidence bounds $U_n(\cdot)$ and $L_n(\cdot)$ from Eq. 3.23.

Computing Confidence Bounds with the Cramer-Chernoff Method

Let X by a random variable, and ψ its log-Laplace transform, which is a convex function on a set $I \subseteq \mathbb{R}$:

$$\forall \lambda \in I,\ \psi(\lambda) \triangleq \log \mathbb{E}\big[e^{\lambda X}\big].$$

As in the previous chapters, lets ψ^* be the Legendre-Fenchel dual of ψ:

$$\psi^*(s) \triangleq \sup_{\lambda \in I} \big(\lambda s - \psi(\lambda)\big).$$

Since ψ is increasing, computing ψ^* is equivalent to invert the derivative:

$$\psi^*(s) \triangleq \psi\big(\psi'^{-1}(s)\big).$$

For a squared Gaussian $X = Y^2$ with $Y \sim \mathcal{N}(\mu, \sigma^2)$, we obtain by classical calculus:

$$\psi'^{-1}(s) = \frac{1}{4}\left(-2s\sigma^2 + 2\mu^2\sigma^2 + \sigma^4 - \sqrt{-4s\mu^2\sigma^4 + 4\mu^4\sigma^4 + 4\mu^2\sigma^6 + \sigma^8}\right)\left(s\sigma^4 - \mu^2\sigma^2 - \sigma^6\right)^{-1}.$$

When closed forms are not available ψ^* can be computed by numerical inversion, as described in the next paragraph. Next, we recall ψ^{*-1} the generalized inverse of ψ^*, that is:

$$\psi^{*-1}(u) = \inf\big\{s \in \mathbb{R} : \psi^*(s) > u\big\}.$$

Up to our knowledge, no closed formulae exists in case of a squared Gaussian. Since the convex conjugate is also convex the inversion is easily done by classical numerical algorithm as in the previous case. With this techniques we obtain the high confidence bound:

$$\mathbb{P}\big[X \leqslant \psi^{*-1}(u)\big] \geqslant 1 - e^{-u}.$$

Numerical Inversion with Newton's Method

The two numerical inversions introduced above are easily solved by Newton's method or even a dichotomic search. Both the derivative ψ' and the convex dual ψ^* are monotonic one-dimensional functions, so the numerical convergence is extremely fast. In our experiments, we performed Newton's algorithm combined with the bisection method (Press, 2007), and always observed quadratic convergence. We modified the Newton's algorithm to include lower and upper bounds on the targeted root, and performed a bisection step each time the Newton's step fall outside the bounds. Other techniques such as the secant method, the Regula Falsi method or the Brent's method (Press, 2007) could be other approaches if initial upper and lower bounds are known.

5.1.4 Software Library Release

We released the source code of the algorithms presented in Chapter 3 as a Matlab library and a Python library. The code is available at this address:

`https://reine.cmla.ens-cachan.fr/e.contal/gpoptimization`

and the documentation on the following web page:

`http://econtal.perso.math.cnrs.fr/software`

The previous numerical techniques from Section 5.1.1 to Section 5.1.3 are all implemented in vector forms, which makes the software roughly as fast as the low level linear algebra libraries. For example the previous Newton-bisection method is implemented so as to compute a large number of steps together and, in Matlab, using all the available cores.

5.2 Applications for Physical Simulations

This section is dedicated to a presentation of the results obtained in two optimization problems in fluid dynamics applications.

5.2.1 Tsunamis Amplification Phenomenon

Small islands in the vicinity of the mainland are widely believed to offer protection from wind waves, and, under some conditions they do. Thus many coastal communities have grown in mainland areas behind small islands. However, whether they offer protection from tsunamis is unclear. Recent post-tsunami onland survey measurements supported by numerical simulations suggest that the run-up—the elevation of the maximum wave uprush on a beach or structure above still water level—on coastal areas behind small islands was significantly higher than on neighboring locations, not affected by the presence of the islands. To study the conditions of this run-up amplification, we solved numerically the nonlinear shallow water equations (NSWE). We use the simplified geometry of a conical island on a flat seafloor in front of a uniform sloping beach. Our objective is to find the maximum run-up amplification with the least number of simulations. To achieve this goal, we used Bayesian optimization with Gaussian process. The search space is defined by five physical parameters, namely the island slope, the beach slope, the water depth, the distance between the island and the plane beach and the incoming wavelength. Our active learning approach reduces

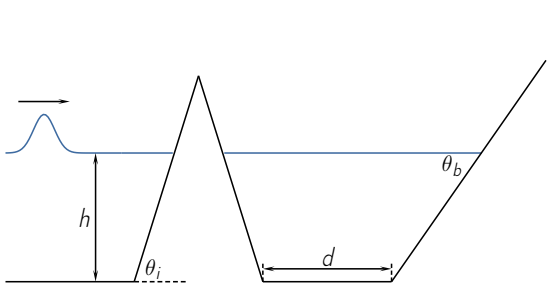
substantially the computations required to determine the maximum run-up amplification. We found that in none of the geometries considered do islands offer any protection, and, that in most cases they amplify the run-up in the shadow zone behind them compared to adjacent unshadowed locales.

Maximization of the Output of Tsunamis Simulations

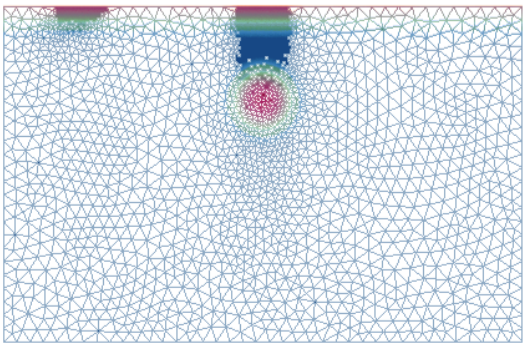
In the last decade, the 26 December 2004 tsunami in Indonesia and the 11 March 2011 event in Japan spread death and destruction and huge economic loss at affected sites. Both events increased public understanding of tsunamis, and raised awareness and preparedness in at-risk communities. Preparedness remains the only effective countermeasure to save lives. The most used quantitative indicator for tsunami impact is the run-up. Since the 1950s tsunami run-up on a plane beach has been extensively studied by Stefanakis et al. (2011) and many authors. Laboratory experiments, numerical computations and analytical models showed that long waves can amplify wave run-up on the lee side of conical islands, compared with the run-up on the side of the island fronting the wave. Interesting recent observations have shown enhanced tsunami run-up in coastal areas in the Mentawai islands off Sumatra, in locales behind small islands, supposedly protected by the islands. Can small islands widely believed to act as natural barriers for tsunamis, transform into amplifiers of wave energy in coastal areas they shadow, which is often where coastal communities thrive? We investigated whether the observation for the 25 October 2010 Mentawais tsunami was caused by unusual combinations of bathymetry and tsunami characteristics, or whether it is indicative of a more general phenomenon. Without computational enhancements, we would have to vary island geometries, coastal beach slopes, offshore depths, distance between the islands and the coastline, and tsunami wavelengths, independently, performing literally thousands of computations to identify patterns, and whether combinations of these parameters produce unusual amplification. We used the GP-UCB-PE algorithm from Section 3.1.2 to limit the numbers of combinations and identify run-up extremes to help us better understand the interaction of the physical parameters and thus identify locales which may be at higher risk of inundation. Moreover, building a Bayesian posterior has further advantages, the most important one being the ability to use it instead of the actual simulator, as it is much less computationally demanding to evaluate, and thus can be applied very rapidly. This can be a substantial advantage in tsunami prediction, when a quick forecast is needed. It is also possible to perform a sensitivity analysis of the model output to the several input parameters. In none of our experiments, did our small islands produce amplification less than one. It appears that, contrary to popular belief and intuition, small islands can act as tsunami lenses focusing energy behind them.

Experimental Configuration

Our simplified bathymetric profile consists of a conical island sitting on a flat seafloor fronting a plane beach (Figure 5.1 Left). The height of the crest of the island above still water level is always 100m. The numerical simulations were performed using VOLNA (Dutykh et al., 2011) which solves the NSWE. VOLNA can simulate the whole life cycle of a tsunami from generation to run-up. The run-up was measured on the plane beach exactly behind the island and on a lateral location along the beach, which was far enough from the island and thus was not directly affected by its presence (Figure 5.1 Right). Since the unknown function



Schematic of the geometry



The unstructured triangular grid

Figure 5.1. – Experimental set-up for tsunami amplification phenomenon

was assumed to be smooth, we choose a squared exponential kernel (Eq. 2.12) as a prior covariance structure. We selected the parameters of the kernel and the noise variance by empirical minimization of the pseudo likelihood (Eq. 5.1) on a data set of 200 observations at locations placed by LHS design (McKay et al., 2000). The fact that the physical simulations were obtainable in parallel by computations on multiple cores motivated the development of novel algorithms for batch-sequential optimization. We then introduced the GP-UCB-PE algorithm (displayed as Algorithm 8 in Chapter 3). The batch size was 20, which led to large improvement compared to purely sequential strategies. We performed 35 iterations, until an innovative stopping criterion was fulfilled.

A Novel Stopping Criterion with Ranking

Our approach to decide when to stop the iterative strategy is to monitor when the procedure ceases to learn relevant information. We attempt to measure the global changes in the estimator μ_n between two successive iterations, with more focus on the highest values, where μ_n from Eq. 2.14 is the posterior expectation of the process. The algorithm then stops when these changes become insignificant for a short period. The change between μ_n and μ_{n+1} is measured by the correlation between their respective values on a finite validation data set $\mathcal{X}_v \subset \mathcal{X}$. Let us denote by n_v the number of elements in the validation set $\mathcal{X}_v$, and $\mathfrak{G}_{n_v}$ the set of all permutations of $1, \ldots, n_n$. Let $\pi_n \in \mathfrak{G}_{n_v}$ be the ranking function associated to μ_n, such that:

$$\pi_n\big(\operatorname*{argmax}_{x\in\mathcal{X}_v} \mu_n(x)\big) = 1,$$
$$\text{and } \pi_n\big(\operatorname*{argmin}_{x\in\mathcal{X}_v} \mu_n(x)\big) = n_v.$$

We then define the discounted rank dissimilarity $d_{\mathcal{X}_v}$ as:

$$d_{\mathcal{X}_v}(\pi_{n-1}, \pi_n) \triangleq \sum_{x\in\mathcal{X}_v} \frac{\big(\pi_n(x) - \pi_{n-1}(x)\big)^2}{\pi_n(x)^2},$$

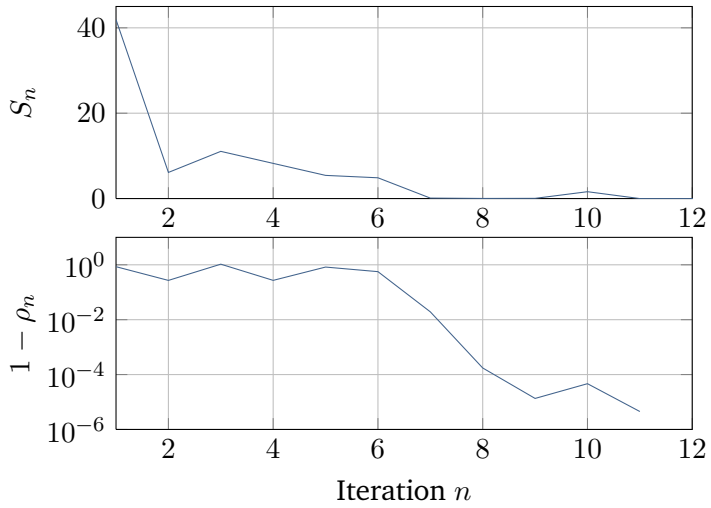


Figure 5.2. – Empirical relationship between the simple regret S_n and the rank dissimilarity $1-\rho_n$ (in log-scale) on a synthetic function.

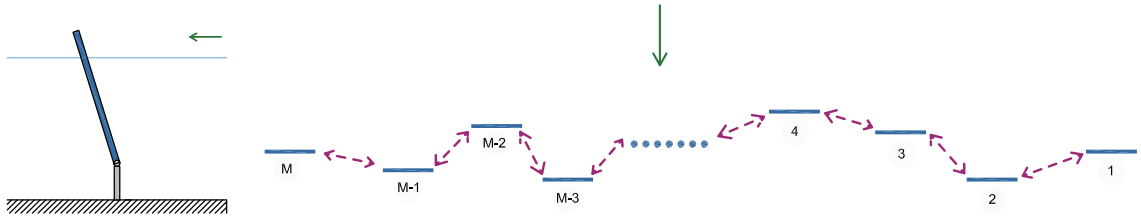
and the normalized rank correlation ρ_n as:

$$\rho_n \triangleq \rho_{\mathcal{X}_v}(\pi_{n-1}, \pi_n) = 1 - Z^{-1} d_{\mathcal{X}_v}(\pi_{n-1}, \pi_n) \text{ where } Z \triangleq \max_{\pi^+, \pi^- \in \mathfrak{S}_{n_v}} d_{\mathcal{X}_v}(\pi^+, \pi^-).$$

The normalization factor Z represents the discounted rank dissimilarity between two reversed ranks π^+ and π^-. This normalized rank correlation can be seen as a modified Spearman's rank correlation adapted to measure changes around the maximum. We stop the algorithm when ρ_n stays below a given threshold t_0 for i_0 iterations in a row. The value of this threshold is fixed empirically. In Figure 5.2, we performed a comparison between ρ_n and the simple regret S_n on synthetic functions, and fixed $t_0 = 10^{-4}$ and $i_0 = 4$.

The Effect of the Island

After running the algorithm, we have found that in none of the situations considered did the island offer protection to the coastal area behind it. On the contrary, we have measured amplified run-up on the beach behind it, compared with a lateral location on the beach, not directly affected by the presence of the island. This finding shows that small islands in the vicinity of the mainland act as amplifiers of long waves at the region directly behind them and not as natural barriers as it was commonly believed. The maximum amplification found by GP-UCB-PE was approximately 70% more than if the island was absent. The island focuses the wave on its lee side. The amplified wave propagates towards the beach and causes higher run-up in the region directly behind the island. One of the key questions is which parameters control the run-up amplification and how. To answer these questions, we can use the posterior mean function of the Gaussian process. We perform a local sensitivity analysis around the maximum by fixing all parameters, except one each time at the value which corresponds to the maximum, and we vary the excluded parameter across the whole range of its input space. We observed that the water depth, the beach slope and the frequency the wave are more important. Our analysis provided one example of what may be possible in the future for tsunami forecasts. Instead of relying on vast databases of pre-computed



Schematic of a WEC Simplified interactions in a WECs array

Figure 5.3. – Simple model of wave energy converters

scenarios, it may well be possible to vary uncertainties in parameters of the initial estimates of fault characteristics available shortly after an event and obtain faster than real time estimates of maximal inundation.

5.2.2 Wave Energy Converters

The wave energy converters (WECs) are devices that uses the motion of the waves to generate electricity. Optimization of the layouts of arrays of wave energy converters (WECs) is a challenging problem. The hydrodynamic analysis and performance estimation of such systems are obtained using semi-analytical and numerical models such as the boundary element method. However, the analysis of an array of such converters becomes computationally expensive, and the computational time increases rapidly with the number of devices in the system. Therefore finding of optimal layouts of WECs in arrays becomes extremely difficult. We present a methodology involving multiple optimization strategies to arrive at the solution to the complex problem. The approach includes predictions of the performance of the WECs in arrays from a Gaussian process learned with the GP-UCB-PE algorithm, followed by a genetic algorithm to obtain the optimal layouts of WECs. The method is extremely fast and easily scalable to arrays of any size. Case studies are performed on a wavefarm comprising of 40 WECs subject to arbitrary bathymetry and space constraints.

Optimization of Spatial Layout of an Array of WECs

Arrays of WECs have been extensively studied in the literature (Budal, 1977; Simon, 1982; Kagemoto and Yue, 1986; Child and Venugopal, 2010). Advances in numerical and analytical techniques in the analysis of wave-structure interactions have enabled the investigation of the behavior of arrays of WECs of arbitrary shapes, taking into account the effects of both the diffracted and the radiated wave fields. The general objective is to understand the effect of the interactions on the performance of the WECs and to determine layouts which would maximize the power captured from the whole system. In order to quantify the effects of the interactions on the performance of the array, the q factor is defined as the ratio of the net power captured (ideally the maximum possible) to the power absorbed by the same WECs in isolation. It is possible to have q factors larger than 1 for particular wave frequencies. But the peaks in the q factor are accompanied with wide troughs in its variation, and since a real ocean is polychromatic, it was suggested that a properly designed array configuration should minimize the effect of the destructive influences. However, the identification of optimal layouts for a particular wave-climate is still a big challenge. The complexity of the optimization problem is manifold. The number of WECs in arrays can vary from one site

Figure 5.4. – Example of bathymetry constraints for the locations of the 40 WECs

to another, and separate optimization needs to be performed in each case. Every single evaluation of the numerical/semi-analytical models has a computational cost which increases with the size of the array. In addition, there can be various constraints to such a problem (e.g. bathymetry variations for nearshore WECs). In Child and Venugopal (2010) the authors present a parabolic intersection method and a genetic algorithm to arrive at the optimal layouts of arrays. While the parabolic intersection approach uses simple calculations for a quick estimate of the array layouts, the genetic algorithm requires many evaluations of a semi-analytical (or numerical) method which is computationally expensive. Such a direct application of sequential optimization techniques also implies that if the number of WECs is changed, a new set of evaluations needs to be computed and analyzed. In this work we propose a fast and scalable approach to address the challenge of determining the best layout for any number of WECs and arbitrary bathymetry constraints. We first use GP-UCB-PE to train a statistical emulator of the individual WECs inside the array. We then predict the performance of the whole wavefarm by evaluating only the quasi-instantaneous posterior. The optimization of the layout under the various constraints is then performed on the predicted performances with a genetic algorithm designed specifically for this task.

Methodology and Bayesian Inference

A wavefarm comprising of $M = 40$ WECs is considered, and so the $2M$ variables that needs to be optimized are $x_1, y_1, \ldots, x_M, y_M$ the horizontal and vertical coordinates of the centers of the WECs. The overall performance q of the array is decomposed as the sum of the powers q_i captured by each individual WEC:

$$q \triangleq 1 + M^{-1} \sum_{i \leqslant M} q_i .$$

In a realistic scenario, the seas are highly irregular, and the interaction effects on a particular WEC due to WECs located away from it are largely diminished. It is reasonable to assume that individual WECs in an array are predominantly influenced by those located very close to them, as illustrated on Figure 5.3. Our approach targets the approximated performance $\widetilde{q}$ where we simplify the model of the individual WECs in order to take into account only a limited number of interactions. A WEC located inside the array is strongly influenced by the two WECs which are nearest to it, *i.e.* one on each side. To model the behavior of such a

WEC, we consider a 3-WECs cluster and focus on the behavior of the central WEC. On the other hand, the edge WECs are modeled using a 2-WECs cluster:

$$\widetilde{q}(x_1, y_1, \ldots, x_M, y_M) \triangleq 1 + M^{-1}\Big(\widetilde{q}_2(x_1, y_1, x_2, y_2) + \sum_{i=2}^{M-1} \widetilde{q}_3(x_{i-1}, y_{i-1}, x_i, y_i, x_{i+1}, y_{i+1}) + \widetilde{q}_2(x_{M-1}, y_{M-1}, x_M, y_M)\Big).$$

The prediction function is computed with the posterior of a Gaussian process with a modified squared exponential kernel, trained with the GP-UCB-PE algorithm on both $\widetilde{q}_2$ and $\widetilde{q}_3$. The introduction of the GP-UCB-PE algorithm is crucial for this approach to succeed. It permits to explore the values of $\widetilde{q}_2$ and $\widetilde{q}_3$ while converging to their optimal configurations using only a manageable number of expensive physical simulations. A traditional methodology involving any space filling exploration would not be affordable in this context. The kernel was designed to incorporate invariance by translation and symmetry of the clusters along the vertical axis. Note that in this setting, computing the exact value of q was not possible in a reasonable amount of time, so we cannot perform optimization of q directly but only $\widetilde{q}$. In our optimization methodology, we considered some constraints which are relevant to the problem. The devices are nearshore WECs with depth specific designs, and as such bathymetry will play a significant role in deciding their locations. We consider an upper and a lower bound on the bathymetry contours, within which the placement of the center of the devices is restricted. Although the mathematical model (for simulations) is based on a constant water depth assumption, the bathymetry constraints in the optimization problem take account of the spatial limitations imposed by the depth variations at real locations, in the placement of the WECs, as illustrated in Figure 5.4. The simulations for the various layouts are performed using the mathematical model of Sarkar et al. (2014). As previously, the parameters of the prior were selected by minimization of the pseudo likelihood, on a data set of 200 observations for $\widetilde{q}_2$ and 800 observations for $\widetilde{q}_3$, and the same stopping criterion was used. The algorithm stopped after 7 iterations for $\widetilde{q}_3$ and only 2 iterations for $\widehat{q}_2$, which is explained by its highly predictable value with a single neighboring WEC.

Optimization of the Predictions with Genetic Algorithms

The determination of the optimal layouts is a challenging task, due to the number of variables and the various constraints. Thanks to the quasi-instantaneous computations of the posterior, we are able to approximate the power captured by any layout in a fast and scalable manner. Once we trained the two predictors of the performance of a WEC, we validated the quality of the predictions by performing leave-one-out cross-validation, which ensured that the predictors are able to generalize the physical properties of any WEC array. We then summed the outputs for the M WECs to get $\widehat{q}$ a prediction of $\widetilde{q}$, and finally aimed at optimizing $\widehat{q}$. The optimization of the predicted performance of an array given some bathymetry constraints is fundamentally different from the global optimization problems considered in this thesis. Indeed the dimension of the search space is large and the evaluations are immediate. We explored the space of valid layouts using a genetic algorithm. The population size of each generation of the genetic algorithm is chosen to be 2^{11}, since the computational time per

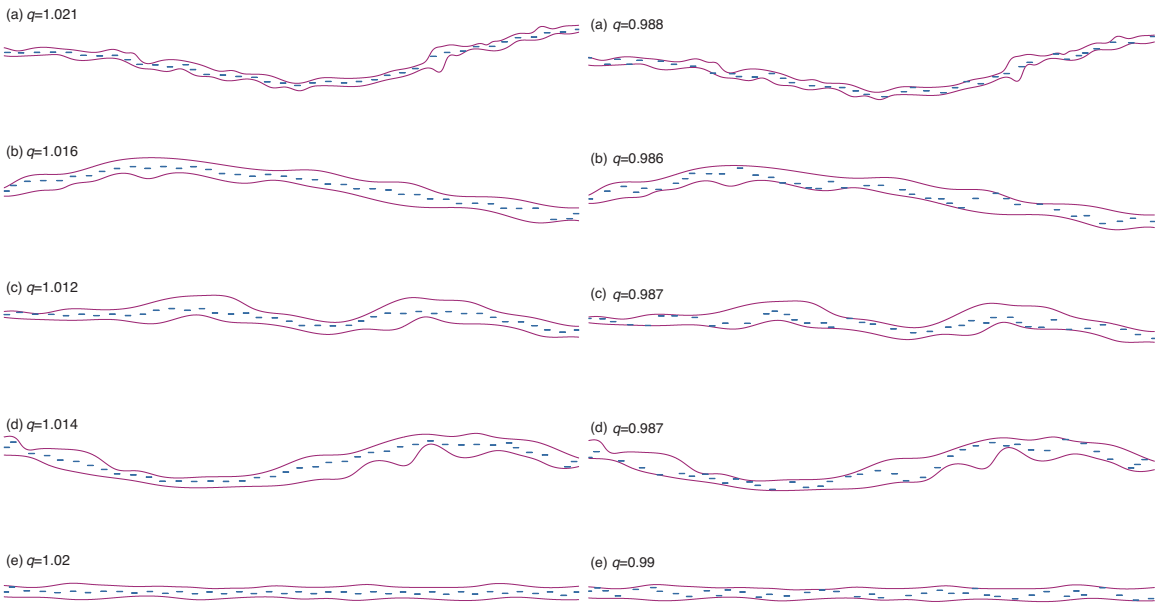


Figure 5.5. – Best WEC array after 16.10^6 array predictions. The displayed values q account for the predicted $\widehat{q}$ factor.

unit prediction ceased to improve beyond it. The initial population of the WEC layouts was selected uniformly such that the bathymetry constraints are all satisfied. In our approach, we used custom crossover and mutation operators such that all the constraints are kept satisfied. Our methodology for initialization, crossover and mutation is described in the reference (Sarkar et al., 2016).

Optimal Predicted WEC Arrays

We considered five different arbitrary bathymetry constraints which take account of the limitations in the placement of the WECs due to depth variations, and fixed the spatial extend of the wavefarm in the horizontal direction. In Figure 5.5 we show the output of the genetic algorithm after 8000 generations (which corresponds to 16 millions array predictions) compared to the best prediction obtained with 16 millions random arrays stratifying the constraints. The predicted $\widehat{q}$ factor obtained by the genetic algorithm is consistently greater than 1, even for stringent bathymetries, which indicates WEC array of performance larger than if the WEC were placed far from each other. The comparison to the best of purely random arrays confirms that the genetic algorithm is useful in that case. The predicted $\widehat{q}$ factor obtained by randomly generated arrays is consistently lower than 1. In our paper (Sarkar et al., 2016), we further analyze the quality and scientific relevance of the predictions by assessing the predictions $\widehat{q}$ and the approximations $\widetilde{q}$ of q, on specific layouts with known q factors. These results are not of optimization concern and go beyond the scope of this thesis.

Conclusion

The developments in this work can have several implications. Besides providing a procedure for optimization, the analysis can offer some practical guidelines to wavefarm developers

in designing arrays. In general, many problems related to ocean engineering/wave energy require evaluation of costly functions (simulation models). Surrogate models based on Gaussian processes can be used to mimic the outcome of the simulations, and the introduction of Bayesian optimization is crucial to obtain relevant emulators in few iterations. The general approach of developing a Bayesian predictor and then performing an optimization using genetic algorithm could be pursued for determining optimal layouts of other type of WECs or renewable energy devices.

5.3 Applications to Model Calibration

In this section, we present further experiments on real applications requiring the non-Gaussian model from Section 3.3.2.

5.3.1 Calibration of Force Fields Parameters for Molecular Simulation

Thanks to the large computational power available, simulations of molecular mixtures is now a popular way to study thermodynamic properties of pure chemical compounds. These force field models define the interactions between molecules, and the involved analytical expressions require calibration parameters that do not have physical interpretation.

Computational Chemistry Methodology

In order to calibrate the force field parameters, we consider the simplified approach below. For a vector of parameters $\theta \in \Theta$, molecular dynamics simulations are computed and multiple thermodynamic properties $\{g_i(\theta)\}_{i \leqslant N}$ are obtained with respective uncertainty u_i, where $g_i : \Theta \to \mathbb{R}$ and $u_i \in \mathbb{R}$ for $1 \leqslant i \leqslant N$. These properties correspond to measurable physical quantities such as liquid density at N different temperatures. Let $\{g_i^{\text{exp}}\}_{i \leqslant N}$ be the experimental values of these quantities. In a Bayesian framework with Gaussian assumptions and uniform prior, the likelihood of the force field parameters is given by:

$$\mathbb{P}\Big[\theta \,|\, \{g_i(\theta), g_i^{\text{exp}}\}_{i \leqslant N}\Big] = Z \exp\big(-\tfrac{1}{2} f(\theta)\big),$$
$$\text{where } f(\theta) \triangleq \sum_{i=1}^{N} \frac{\big(g_i(x) - g_i^{\text{exp}}\big)^2}{u_i^2},$$

and $Z \in \mathbb{R}$ is a normalization constant. The methodology is then to minimize f with respect to θ by successive computations of molecular dynamics. Once the parameters θ are optimized, the force field model can be used to predict unknown thermodynamic properties $g(\theta)$.

Global Optimization of Quadratic Forms

The minimization of f is often done with gradient-based algorithms. However, the function f may not be convex and this approach may fail to find the global minimum. In Cailliez et al. (2014), the authors assumed that the properties $g_i(\cdot)$ are realizations of independent

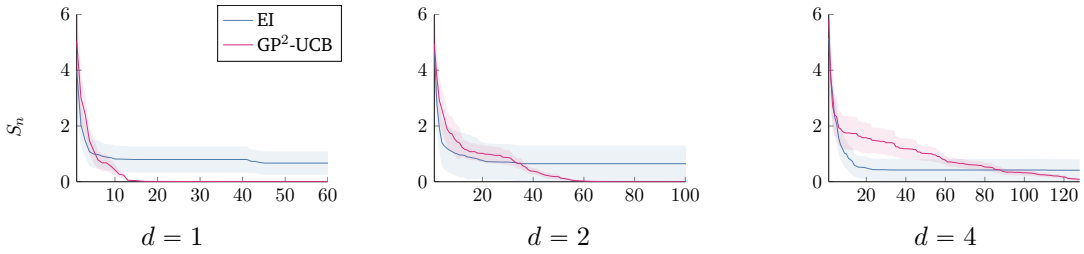


Figure 5.6. – Empirical mean and confidence interval of the simple regret S_n in term of iteration n on sums of squared Gaussian process.

Gaussian processes $\Theta \to \mathbb{R}$. They used the EI rule (Jones et al., 1998) to sequentially optimize f:

$$x_{n+1} \in \operatorname*{argmax}_{x \in \mathcal{X}} \mathbb{E}\Big[\big(f(x_{\hat{\imath}_n}) - f(x) \big)_+ \,|\, \mathcal{F}_n \Big],$$
$$\text{where } \hat{\imath}_n \in \operatorname*{argmin}_{i \leqslant n} y_i.$$

However this selection strategy is designed for f being a Gaussian process, not a quadratic form. The authors bypassed this issue by computing numerical expectation of f, along samples of the Gaussian processes g_i. A collaboration with the authors motivated the work previously presented in Section 3.3.2, where we introduce a sound and computationally efficient algorithm to optimize sums of squared Gaussian processes.

Experimental Results

We present here an experimental comparison of the performances of our novel Algorithm 13 from Section 3.3.2, that we called GP^2-UCB, against the EI strategy with Monte Carlo expectations. The objective function is a sum three squared non-centered Gaussian processes:

$$f(x) = \sum_{i=1}^{3} g_i^2(x),$$
$$\text{where } g_i \sim \mathcal{GP}(m_i, k_i).$$

In this setting k_i is known but m_i is not. Bayesian inference adapts easily to the problem of estimating m_i, as stated in Section 2.7 of Rasmussen and Williams (2006). For the EI strategy, the expectation is computed with 10^4 Monte Carlo samples. The evolution of the simple regret S_n along the iterations is reported in Figure 5.6. It is clear that the EI rule fails to converge to the global optimum of f. As guaranteed by the theoretical analysis, the GP^2-UCB algorithm consistently finds the optimum.

5.3.2 Hyper-Parameter Optimization and Further Perspectives

We conclude this section by presenting further possible applications of the algorithms developed in this thesis.

Squared Gaussian Processes

A famous application of Bayesian optimization is the selection of hyper-parameters of machine learning algorithms. The hyper-parameters are typically selected in order to minimize a generalization loss computed empirically by cross-validation (Snoek et al., 2012). However standard loss functions do not look like Gaussian processes. A common approach considers to model the logarithm of the loss function as a Gaussian process, to alleviate the positivity and scale issues. The previous methodology using the GP^2-UCB Algorithm 13 is a perfect candidate for optimizing any (monotone composition of) sum of squares, when one has access to the individual terms. The classical mean-squared-error loss is the canonical example of such objective, and other losses such as Gaussian likelihood also fit in this setting.

Simple Ranking Structures in High Dimension

In addition to the GP^2-UCB algorithm, we have stated that the AdaRankOpt Algorithm 15 has shown fast empirical convergence for functions in higher dimension with simple ranking structures. From empirical observation, we believe that the optimization objectives from cross-validation have simple ranking structures even when the number of hyper-parameters is large. The AdaRankOpt algorithm will perform better than its competitor for many dimensional functions with few modes and a presence of "barriers", that is regions in the search space where the loss explodes. Indeed, Lipschitzian or Bayesian optimization are not well suited for such cases.

Conclusion and Perspectives

This thesis presented new algorithms and theoretical results for global optimization. We first described a Bayesian optimization technique selecting the sequential queries by mini-batches instead of one by one. We proved theoretical guarantees for this method for various notions of regrets. In particular for the usual cumulative regret, the fact that we select multiple queries simultaneously without waiting for observations has an impact on the regret only by constant factors. We then leveraged novel techniques using discretization trees for the canonical metric of Gaussian processes, in order to adapt Bayesian optimization algorithms such as UCB to arbitrary compact spaces. The regret bounds we derived are similar to the state-of-the-art on finite search space up to poly-logarithmic factors. These regret bounds involve the metric entropy of $\mathcal{X}$ instead of its cardinality, and especially the covering dimension. We proved that the discretization error of our method is optimal up to constant terms, which forms a step toward proving regret lower bounds for Bayesian optimization in metric spaces. We then gave further results on Bayesian optimization, and shown that Bayesian optimization is still sound for objectives that are not Gaussian processes but other stochastic processes with less stringent smoothness assumptions. We analyzed in particular quadratic forms of Gaussian processes. Later we introduced a novel optimization framework for non-smooth functions. Our method relies on ranking structures, a condition on the level sets, and the obtained algorithms perform only function comparisons. We gave theoretical analysis of the error under mild assumptions, and shown that this method performs well in practice against state-of-the-art competitors. Finally, we presented practical considerations and multiple applications leveraging the above works, for various real research problems.

This thesis exhibits connections between hierarchical optimistic optimization and Bayesian optimization. Yet, we plan to analyze further the links between those approaches, and notably with respect to fast rates for the simple regret in Bayesian optimization algorithms. The work on Bayesian optimization in metric space opens new perspectives to deriving theoretical lower bounds on the regret and optimal algorithms. Another benefit of having lower bounds on the supremum is that it permits to derive strategies with fast convergence rate for deterministic observations, in the spirit of the DOO Algorithm 5 from Section 2.2.3. Unlike Lipschitzian optimization, in Bayesian optimization the metric ℓ does not control directly the maximum increment of f, and we may use instead the values $\omega(x)$ obtained by generic chaining from Theorem 3.2. We believe that the following deterministic optimization algorithm may exhibit fast convergence rates in typical scenarios, and we planed to analyze its simple regret. This strategy, detailed in Algorithm 16 performs as follows: it greedily grows a sub-tree of the discretization tree, for which every internal nodes has been evaluated, and queries leaf nodes.

Algorithm 16: GP-DOO

$x_1 \leftarrow \mathcal{T}_0$
$f(x_1) \leftarrow$ **Query**(x_1)
$L \leftarrow \text{Children}(x_1)$
for $n = 1, 2, \ldots$ **do**
  $\forall x \in L,\ U(x) \leftarrow f(p(x)) + \omega(p(x))$
  $x_{n+1} \leftarrow \text{argmax}_{x \in L}\, U(x)$
  $f(x_{n+1}) \leftarrow$ **Query**(x_{n+1})
  $L \leftarrow L \cup \text{Children}(x_{n+1})$
  $L \leftarrow L \backslash \{x_{n+1}\}$
end

The selected leaf is among the ones whose parent maximizes the upper confidence bound, that is, with L the current set of leaves:

$$x_{n+1} \leftarrow \operatorname*{argmax}_{x \in L} f(p(x)) + \omega(p(x)).$$

The tree is computed incrementally with the GreedyCover procedure as for Algorithm 11. Like in Section 2.2.3, this algorithm selects queries in a particular set of near-optimal nodes containing the optimum of f. Now, when it exists $\varepsilon > 0$ such that $\omega(p(x_{n+1})) < \varepsilon$, the simple regret is at most ε. An analysis of the near-optimal dimension would lead to fast regret bounds when $\dim_\rho(\mathcal{X}, \ell) = 0$ holds. Yet the analysis of the near-optimal dimension, a complicated random variable, cannot be done with usual tools from global optimization.
Finally, the work on non-smooth optimization using ranking opens new prospects, specifically to adapt the algorithm for noisy observations, and to investigate sufficient conditions to obtain fast convergence rates.

Appendix

A

Attempt for Improved Regret Bounds

In this section, we describe a fruitless attempt to prove logarithmic cumulative regrets with high probability for Gaussian process optimization. This work has been presented in Contal et al. (2014), before we discovered a mistake in the proof.

1 Proof Techniques to Get Rid of the Square Root

The lower bound from Eq. 2.31 tells us that logarithmic cumulative regrets cannot be achieved in expectation, in the general setting. Yet, this does not include results with high probability $1 - e^{-u}$ if the parameter u is fixed and known. Since the $\mathcal{O}(\sqrt{n})$ regret obtained by the GP-UCB algorithm and other related algorithms comes from the use of the Cauchy-Schwarz inequality:

$$\sum_{i=1}^{n} s_n \leqslant \sqrt{n \sum_{i=1}^{n} s_n^2} \leqslant \sqrt{c_\eta n \gamma_n},$$

this motivates the investigation of better regret bounds in particular settings where the Cauchy-Schwarz inequality would be over conservative.

Preliminary Observations

Our idea works as follows. Since for Gaussian processes the posterior given n observations at points x_1, x_2 is Gaussian, as stated in Eq. 3.5, we may be able to get better concentration from analysis of Gaussian martingale. Let $x^\star$ be a fixed point of $\mathcal{X}$ and x_n a $\mathcal{F}_{n-1}$-measurable query. Given $\mathcal{F}_{n-1}$, we have that $f(x^\star) - f(x_n)$ is a Gaussian $\mathcal{N}\big(\mu_{n-1}(x^\star) - \mu_{n-1}(x_n), \ell_{n-1}^2(x^\star, x_n)\big)$. Let M_n be the centered cumulative differences,

$$M_n \triangleq \sum_{i=1}^{n} \Delta M_i, \tag{A.1}$$
$$\text{with } \Delta M_i \triangleq f(x^\star) - f(x_i) - \big(\mu_{i-1}(x^\star) - \mu_{i-1}(x_i)\big).$$

The increment ΔM_i is thus distributed as a centered Gaussian with variance $\ell_{i-1}^2(x^\star, x_i) \leqslant \sigma_{i-1}^2(x^\star) + \sigma_{i-1}^2(x_i)$. Let the deterministic sequence y_n equals to $c_\eta \gamma_n$ and the stochastic sequence N_n equals to $\sum_{i=1}^{n} \sigma_{i-1}^2(x^\star)$, with $c_\eta = \frac{2}{\log(1+\eta^{-2})}$ like previously and γ_n defined in Eq. 2.33, so that $\sum_{i=1}^{n} \ell_{i-1}^2(x^\star, x_i) \leqslant y_n + N_n$.

Self-Normalized Martingale Inequalities

In the next lemma we get self-normalized Cramér-type exponential inequalities for a Gaussian martingale, also known as the *method of mixtures*. Unfortunately, we will see later that this lemma cannot be applied to obtain upper bounds on the cumulative regret.

Lemma A.1 (Gaussian Martingale with Predictable Quadratic Variation). *Let $(M_n)_{n\geqslant 1}$ be a Gaussian martingale adapted to $\mathcal{F}_n$, that is it exists $(\ell_n)_{n\geqslant 1}$ a sequence such that ℓ_n is $\mathcal{F}_n$-measurable and for all $\lambda \in \mathbb{R}$:*

$$\log \mathbb{E}\Big[e^{\lambda \Delta M_n} \,|\, \mathcal{F}_{n-1}\Big] = \frac{1}{2}\lambda^2 \ell_{n-1}^2,$$
$$\text{where } \Delta M_n \triangleq M_{n+1} - M_n.$$

If $\langle M\rangle_n \triangleq \sum_{i\leqslant n} \ell_{i-1}^2 \leqslant y_n + N_n$ where N_n is $\mathcal{F}_{n-1}$-measurable and y_n is a scalar, then for all $u > 0$:

$$\mathbb{P}\left[M_n \leqslant \sqrt{2uy_n} + \sqrt{\frac{u}{2y_n}} N_n\right] \geqslant 1 - e^{-u}.$$

Proof. The proof of this lemma is inspired by the work of de la Peña (1999) and Bercu and Touati (2008). For all $x > 0$, let A_n the event $\left\{M_n \geqslant x\left(1 + \frac{N_n}{2y_n}\right)\right\}$. By Markov's inequality, for all $a > 0$ we know that:

$$\mathbb{P}\big[A_n\big] \leqslant \mathbb{E}\left[\exp\left(aM_T - ax\left(1 + \frac{N_T}{2y_T}\right)\right)\right].$$

Let us define $W_n(a) \triangleq \exp\left(aM_n - \frac{a^2}{2}\langle M\rangle_n\right)$. Plugging W_n in the previous inequality we obtain $\mathbb{P}\big[A_n\big] \leqslant \mathbb{E}\left[W_n(a)\exp\left(\frac{a^2}{2}\langle M\rangle_n - ax\left(1 + \frac{N_n}{2y_n}\right)\right)\right]$. Using $\langle M\rangle_n \leqslant y_n + N_n$ and choosing $a = \frac{x}{y_n}$, this simplifies to:

$$\mathbb{P}\big[A_n\big] \leqslant \exp\left(-\frac{x^2}{2y_n}\right)\mathbb{E}\big[W_n(a)\big].$$

The martingale (M_n) being Gaussian, we know by the law of iterated expectations and by induction that $\mathbb{E}\big[W_n(a)\big] = 1$ for all $a > 0$. The proof of Lemma A.1 follows by choosing $x = \alpha\sqrt{y_n}$. □

In order to obtain sharp bounds on the martingale from Lemma A.1, we choose x_n such that the sum of the expectations $\mu_{n-1}(x^\star) - \mu_{n-1}(x_n)$ cancels out the term involving N_n, at a price smaller than $\sqrt{2uy_n}$.

Lemma A.2 (Inequality with the Exploration Function). *Let $y_n \in \mathbb{R}$, $\mu_n : \mathcal{X} \to \mathbb{R}$ and $\sigma_n : \mathcal{X} \to \mathbb{R}$ such that $0 \leqslant \sigma_n(x) \leqslant 1$. Fix $u > 0$ and let x_n be the maximizer of $\mu_{n-1}(x) + \phi_{n-1}(x)$ for $x \in \mathcal{X}$, where the exploration function ϕ_{n-1} is defined as:*

$$\forall x \in \mathcal{X},\ \phi_{n-1}(x) \triangleq \sqrt{2u}\left(\sigma_{n-1}^2(x) + \sum_{i=1}^{n-1}\sigma_{i-1}^2(x_i)\right)^{\frac{1}{2}}.$$

Then the following inequality holds for all $x^\star \in \mathcal{X}$:

$$\sum_{i=1}^{n}\big(\mu_{i-1}(x^\star) - \mu_{i-1}(x_n)\big) \leqslant \sqrt{2uy_n} - \sqrt{\frac{u}{2(y_n+1)}}\sum_{i=1}^{n}\sigma_{n-1}^2(x^\star),$$

when $y_n \geqslant \sum_{i=1}^{n}\sigma_{i-1}^2(x_i)$.

Proof. By construction of x_n, for all $n \geqslant 1$, $\mu_{n-1}(x^\star) - \mu_{n-1}(x_n) \leqslant \phi_{n-1}(x_n) - \phi_{n-1}(x^\star)$. We can then rewrite the sum of the exploration terms as follows,

$$\sum_{i=1}^{n} \big(\phi_{i-1}(x_i) - \phi_{i-1}(x^\star)\big) = \sum_{i=1}^{n} \big(\widetilde{\phi}_{i-1}(x_i) - \widetilde{\phi}_{i-1}(x^\star)\big),$$

where $\widetilde{\phi}_n$ differs from ϕ_n only by a constant term,

$$\widetilde{\phi}_n(x) \triangleq \phi_n(x) - \sum_{i=1}^{n-1} \widetilde{\phi}_i(x_i).$$

Let $\xi_n \triangleq \sum_{i=1}^{n} \sigma_{i-1}^2(x_i)$. With the introduction of $\widetilde{\phi}_n$ it is easy to see that $\sum_{i=1}^{n} \widetilde{\phi}_{i-1}(x_i) = \sqrt{2u\xi_n}$ and,

$$\sum_{i=1}^{n} \big(\phi_{i-1}(x_i) - \phi_{i-1}(x^\star)\big) = \sqrt{2u\xi_n} + \sqrt{2u} \sum_{i=1}^{n} \Big(\sqrt{\xi_{i-1}} - \sqrt{\xi_{i-1} + \sigma_{i-1}^2(x^\star)}\Big).$$

By concavity of the square root, we have for all $a \geqslant -b$ that $\sqrt{a+b} - \sqrt{a} \leqslant \frac{b}{2\sqrt{a}}$. Introducing the notations $a_i \triangleq \xi_{i-1} + \sigma_{i-1}^2(x^\star)$ and $b_i \triangleq -\sigma_{i-1}^2(x^\star)$, we obtain,

$$\sum_{i=1}^{n} \big(\phi_{i-1}(x_i) - \phi_{i-1}(x^\star)\big) \leqslant \sqrt{2u\xi_n} + \sqrt{2u} \sum_{i=1}^{n} \frac{b_i}{2\sqrt{a_i}}.$$

Moreover, with $0 \leqslant \sigma_i^2(x) \leqslant 1$ for all $x \in \mathcal{X}$, we have $a_i \leqslant \xi_i + 1$ and $b_i \leqslant 0$ for all $i \geqslant 1$ which gives,

$$\sum_{i=1}^{n} \frac{b_i}{\sqrt{a_i}} \leqslant -\frac{\sum_{i=1}^{n} \sigma_{i-1}^2(x^\star)}{\sqrt{\xi_n + 1}},$$

leading to the inequality of Lemma A.2. □

We now combine both Lemma to obtain an upper bound which does not involves an additional $\sqrt{n}$ term.

Lemma A.3 (Concentration with Exploration Function)**.** *Let M_n be a sequence adapted to $\mathcal{F}_n$, such that for all $\lambda \in \mathbb{R}$:*

$$\log \mathbb{E}\Big[\exp(\lambda \Delta M_n) \,|\, \mathcal{F}_{n-1}\Big] = \lambda\big(\mu_{n-1}(x^\star) - \mu_{n-1}(x_n)\big) + \frac{\lambda^2}{2} \ell_{n-1}^2(x^\star, x_n),$$

with $\ell_{n-1}^2(x^\star, x_n) \leqslant \sigma_{n-1}^2(x^\star) + \sigma_{n-1}^2(x_n)$. When $x_n = \operatorname{argmax}_{x \in \mathcal{X}} \big\{\mu_{n-1}(x) + \phi_{n-1}(x)\big\}$ with the notations and conditions of Lemma A.1 and A.2:

$$\mathbb{P}\Big[M_n \leqslant 2\sqrt{2uy_n} + \sqrt{2u}\Big] \geqslant 1 - e^{-u}.$$

Proof. The process defined by the increments $\Delta M_n - \mathbb{E}\big[\Delta M_n \,|\, \mathcal{F}_{n-1}\big]$ is a centered Gaussian martingale adapted to $\mathcal{F}_n$. Therefore using Lemma A.1,

$$\mathbb{P}\left[M_n \leqslant \sum_{i=1}^{n} \Big(\mu_{n-1}(x^\star) - \mu_{n-1}(x_n)\Big) + \sqrt{2u(y_n+1)} + \sqrt{\frac{u}{2(y_n+1)}} N_n\right] \geqslant 1 - e^{-u}.$$

Algorithm 17: GP-MI (k, η, u)

$\xi_0 \leftarrow 0$
for $n = 1, 2, \dots$ **do**
  Compute μ_n and σ_n^2 (Eq. 2.14, 2.16)
  $\forall x \in \mathcal{X}\ \phi(x) \leftarrow \sqrt{2u}\big(\sigma_{n-1}^2(x) + \xi_{n-1}\big)^{\frac{1}{2}}$,
  $x_{n+1} \leftarrow \operatorname{argmax}_{x \in \mathcal{X}} \big\{\mu_n(x) + \phi(x)\big\}$
  $y_{n+1} \leftarrow$ **Query**(x_{n+1})
  $\xi_{n+1} \leftarrow \xi_{n+1} + \sigma_n^2(x_{n+1})$
end

Now by the selection of x_n, Lemma A.2 gives:

$$\mathbb{P}\left[M_n \leqslant 2\sqrt{2uy_n} + \sqrt{2u}\right] \geqslant 1 - e^{-u}. \qquad \square$$

2 The GP-MI Algorithm and Theoretical Obstacles

Following the query rule from Lemma A.2, we define the Gaussian Process with Mutual Information algorithm (GP-MI), selecting:

$$\begin{aligned} x_{n+1} &\triangleq \operatorname*{argmax}_{x \in \mathcal{X}} \big\{\mu_n(x) + \phi_n(x)\big\}, \\ \text{where } \phi_n(x) &\triangleq \sqrt{2u}\big(\sigma_n^2(x) + \xi_n\big)^{\frac{1}{2}}, \\ \text{and } \xi_n &\triangleq \sum_{i=1}^{n} \sigma_{i-1}^2(x_i). \end{aligned}$$

We have the informational upper bound $\xi_n \leqslant c_\eta \gamma_n$ where $c_\eta \triangleq \frac{2}{\log(1+\eta^{-2})}$ and γ_n the maximal mutual information on f defined in Eq. 2.33, hence the name of the GP-MI algorithm. If the upper bound from Lemma A.3 was valid for the cumulative regret R_n of the GP-MI algorithm, we would have:

$$R_n \leqslant \mathcal{O}\big(\sqrt{\gamma_n}\big),$$

with high probability, that is logarithmic regret for linear kernel and poly-logarithmic regret for squared exponential kernel. Unfortunately, we face two obstacles to apply the previous lemmas. These two difficulties may be bypassed by slight modifications, but getting rid of both obstacles together is not possible for usual Bayesian optimization.

The Supremum is Not Gaussian

First, in the previous section we fixed a point $x^\star \in \mathcal{X}$ and considered $f(x^\star)$ which is distributed as Gaussian, instead of considering $\sup_{x^\star \in \mathcal{X}} f(x^\star)$, which is not Gaussian. It is simple to get rid of this problem by taking union bounds on finite $\mathcal{X}$ at a price $\log|\mathcal{X}|$, or applying chaining and discretization tools from Chapter 3 for continuous $\mathcal{X}$.

The Cumulative Regret is not Adapted to the Filtration

The second obstacle is that R_n is not $\mathcal{F}_{n-1}$-measurable. If we are interested in expected cumulative regret instead of high probability results, the measurability issue can be solved by the following trick. We can decompose the expected cumulative regret as:

$$\begin{aligned}\mathbb{E}\big[R_n\big] &= \mathbb{E}\left[\sum_{i\leqslant n} \sup_{x^\star\in\mathcal{X}} \big(f(x^\star) - f(x_i)\big)\right] \\ &= \sum_{i\leqslant n} \mathbb{E}\left[\mathbb{E}\left[\sup_{x^\star\in\mathcal{X}} f(x^\star) - f(x_i) \,\middle|\, \mathcal{F}_{i-1}\right]\right].\end{aligned}$$

At every iteration i, we introduce a stochastic processes $\widetilde{f}_i$ such that given $\mathcal{F}_{i-1}$, this stochastic process is independent and has the same distribution as f :

$$\mathcal{L}\big(\widetilde{f}_i \,|\, \mathcal{F}_{i-1}\big) \triangleq \mathcal{L}\big(f \,|\, \mathcal{F}_{i-1}\big).$$

For an algorithm that only observes information from $\mathcal{F}_{i-1}$, both stochastic processes f and $\widetilde{f}_i$ are equivalent. Let us define a corresponding regret we denote by *resampling cumulative regret*:

$$\widetilde{R}_n \triangleq \sum_{i\leqslant n} \sup_{x^\star\in\mathcal{X}} \big(\widetilde{f}_i(x^\star) - \widetilde{f}_i(x_i)\big).$$

Thanks to the above decomposition we verify that it is equal to the usual cumulative regret in expectation,

$$\mathbb{E}\big[R_n\big] = \mathbb{E}\big[\widetilde{R}_n\big].$$

However, since we instantiate a new stochastic process at each iteration, the variance of the resampling cumulative regret is typically much smaller. Now, we define the instantaneous regret for the resampled process as:

$$\widetilde{r}_n \triangleq \sup_{x^\star\in\mathcal{X}} \widetilde{f}_n(x^\star) - \widetilde{f}_n(x_n),$$

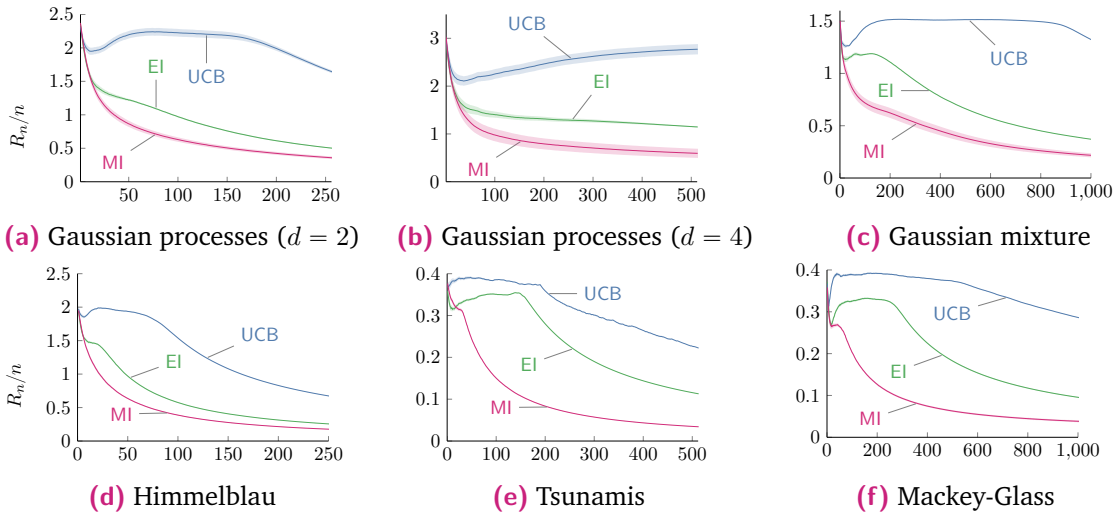
and $\big(\widetilde{\mathcal{F}}_n\big)_{n\geqslant 1}$ the following filtration:

$$\widetilde{\mathcal{F}}_n \triangleq \sigma\Big(x_1, y_1, \widetilde{r}_1, \ldots, x_n, y_n, \widetilde{r}_n\Big).$$

The resampling cumulative regret is $\widetilde{\mathcal{F}}_n$-measurable, and since the additional information in $\widetilde{\mathcal{F}}_n$ compared to $\mathcal{F}_n$ is independent of $\widetilde{f}_{n+1}$, the analysis is not changed. However, we face now the first obstacle—that is $\sup_{x^\star} \widetilde{f}_n(x^\star)$ is not a Gaussian—and the previous trick cannot be applied for the expected resampling cumulative regret since the supremum is moved inside the sum. Another method to bypass this difficulty is to use the natural filtration of R_n, that is:

$$\sigma\Big(f^\star - f(x_1), \cdots, f^\star - f(x_n)\Big),$$

with $f^\star \triangleq \sup_{x^\star} f(x^\star)$. In order to build an algorithm, one has to know $f^\star$ in advance. We do not consider this as a valid approach for global optimization, but we mention that



(a) Gaussian processes ($d = 2$) **(b)** Gaussian processes ($d = 4$) **(c)** Gaussian mixture

(d) Himmelblau **(e)** Tsunamis **(f)** Mackey-Glass

Figure A.1. – Empirical mean and confidence interval of the average regret $\frac{R_n}{n}$ in term of iteration n on real and synthetic tasks for the GP-MI and GP-UCB algorithms and the EI heuristic (lower is better).

this assumption has been used to prove constant cumulative regret in multi-armed bandits (Bubeck et al., 2013).

3 Empirical Assessment

Although the GP-MI algorithm is left unproved, we performed some numerical comparison against two competitors: the GP-UCB algorithm (Srinivas et al., 2012) and the EI algorithm (Jones et al., 1998).

Experimental Protocol

The tasks used for assessment are similar to the ones from Section 3.1.4, and we refer to this section for a detailed description. For all data sets, the algorithms were initialized with a random subset of 10 observations $\{(x_i, y_i)\}_{i \leqslant 10}$. When the prior distribution of the underlying function was not known, the Bayesian inference was made using a squared exponential kernel. We first picked the half of the data set to estimate the hyper-parameters of the kernel via cross validation in this subset. In this way, each algorithm was running with the same prior information. The value of the confidence parameter u for the GP-MI and the GP-UCB algorithms was fixed to $e^{-u} = 10^{-6}$ for all these experimental tasks. The results are provided in Figure A.1. The curves show the evolution of the average regret $n^{-1}R_n$ in term of iteration n. We report the mean value with the confidence interval over a hundred experiments.

Empirical Comparisons

For the most difficult assessments the GP-UCB algorithm performs poorly against the two others, and the GP-MI algorithm always surpasses the EI heuristic. However, for more difficult or large scale optimization problems, the GP-MI algorithm is not guaranteed to converge to the optimum. This may be explained by the fact that theory-driven algorithms, like GP-UCB, are typically over-conservative. Those observations are encouraging to pursue research on algorithm incurring better regrets.

Bibliography

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 2312–2320.

Adler, R. J. and Taylor, J. E. (2009). *Random fields and geometry*. Springer Science & Business Media.

Agarwal, S., Graepel, T., Herbrich, R., Har-Peled, S., and Roth, D. (2005). Generalization bounds for the area under the roc curve. *Journal of Machine Learning Research*, 6(Apr):393–425.

Agrawal, R. (1995). The continuum-armed bandit problem. *SIAM journal on control and optimization*, 33(6):1926–1951.

Audibert, J.-Y., Bubeck, S., and Munos, R. (2010). Best arm identification in multi-armed bandits. *Proceedings of the 23th Conference on Learning Theory (COLT)*, 23:13.

Audibert, J.-Y., Munos, R., and Szepesvári, C. (2009). Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902.

Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *The Journal of Machine Learning Research*, 3:397–422.

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256.

Auer, P. and Ortner, R. (2010). Ucb revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65.

Auer, P., Ortner, R., and Szepesvári, C. (2007). Improved rates for the stochastic continuum-armed bandit problem. In *Proceedings of the 20th Conference on Learning Theory (COLT)*, pages 454–468. Omnipress.

Azimi, J., Fern, A., and Fern, X. (2010). Batch bayesian optimization via simulation matching. In *Advances in Neural Information Processing Systems 24 (NIPS)*, pages 109–117. Curran Associates, Inc.

Bect, J., Bachoc, F., and Ginsbourger, D. (2016). A supermartingale approach to Gaussian process based sequential design of experiments. arXiv:1608.01118.

Bercu, B. and Touati, A. (2008). Exponential inequalities for self-normalized martingales with applications. *The Annals of Applied Probability*, 18(5):1848–1869.

Borell, C. (1975). The Brunn-Minkowski inequality in Gauss space. *Inventiones mathematicae*, 30(2):207–216.

Borgwardt, K. M. and Kriegel, H.-P. (2005). Shortest-path kernels on graphs. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE.

Boucheron, S., Bousquet, O., and Lugosi, G. (2005). Theory of classification: A survey of some recent advances. *ESAIM: probability and statistics*, 9:323–375.

Boucheron, S., Lugosi, G., and Massart, P. (2013). *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press.

Box, G. E. P. and Wilson, K. B. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 13(1):1–45.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Brochu, E., Cora, V. M., and De Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. eprint arXiv:1012.2599, arXiv.org.

Bubeck, S. (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8:231–357.

Bubeck, S. and Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122.

Bubeck, S., Munos, R., and Stoltz, G. (2009). Pure exploration in multi-armed bandits problems. In *Proceedings of the International Conference on Algorithmic Learning (ALT)*, pages 23–37. Springer-Verlag.

Bubeck, S., Munos, R., Stoltz, G., and Szepesvári, C. (2008). Online optimization in X-armed bandits. In *Advances in Neural Information Processing Systems 22 (NIPS)*, pages 201–208. Curran Associates, Inc.

Bubeck, S., Munos, R., Stoltz, G., and Szepesvári, C. (2011). X-armed bandits. *Journal of Machine Learning Research*, 12:1655–1695.

Bubeck, S., Perchet, V., and Rigollet, P. (2013). Bounded regret in stochastic multi-armed bandits. In *COLT*, pages 122–134.

Budal, K. (1977). Theory for absorption of wave power by a system of interacting bodies. *Journal of Ship Research*, 21(4).

Bull, A. D. (2011). Convergence rates of efficient global optimization algorithms. *The Journal of Machine Learning Research*, 12:2879–2904.

Bull, A. D. (2015). Adaptive-treed bandits. *Bernoulli*, 21(4):2289–2307.

Cailliez, F., Bourasseau, A., and Pernot, P. (2014). Calibration of forcefields for molecular simulation: Sequential design of computer experiments for building cost-efficient kriging metamodels. *Journal of computational chemistry*, 35(2):130–149.

Cappé, O., Garivier, A., Maillard, O.-A., Munos, R., and Stoltz, G. (2013). Kullback–leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 41(3):1516–1541.

Child, B. F. M. and Venugopal, V. (2010). Optimal configurations of wave energy device arrays. *Ocean Engineering*, 37(16):1402–1417.

Cirel'son, B. S., Ibragimov, I. A., and Sudakov, V. N. (1976). Norms of Gaussian sample functions. In *Proceedings of the Third Japan—USSR Symposium on Probability Theory*, pages 20–41. Springer.

Clémençon, S. (2011). On U-processes and clustering performance. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 37–45.

Clémençon, S., Lugosi, G., and Vayatis, N. (2008). Ranking and empirical minimization of U-statistics. *The Annals of Statistics*, pages 844–874.

Clémençon, S. and Vayatis, N. (2010). Overlaying classifiers: a practical approach to optimal scoring. *Constructive Approximation*, 32(3):619–648.

Cohn, D., L., A., and Ladner, R. (1994). Improving generalization with active learning. *Mach. Learn.*, 15(2):201–221.

Combes, R. and Proutiere, A. (2014). Unimodal bandits: Regret lower bounds and optimal algorithms. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 521–529.

Contal, E., Buffoni, D., Robicquet, A., and Vayatis, N. (2013). Parallel Gaussian process optimization with upper confidence bound and pure exploration. In *Proceedings of the European Conference on Machine Learning (ECML)*, volume 8188, pages 225–240. Springer Berlin Heidelberg.

Contal, E., Malherbe, C., and Vayatis, N. (2015). Optimization for gaussian processes via chaining. *NIPS Workshop on Bayesian Optimization*.

Contal, E., Perchet, V., and Vayatis, N. (2014). Gaussian process optimization with mutual information. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 253–261. icml.cc / Omnipress.

Côté, F. D., Psaromiligkos, I. N., and Gross, W. J. (2012). A Chernoff-type lower bound for the Gaussian Q-function. arXiv preprint arXiv:1202.6483.

Cox, D. D. and John, S. (1997). SDO: A statistical method for global optimization. *Multidisciplinary design optimization: state of the art*, pages 315–329.

Cristianini, N. and Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

da Sliva Santos, C. H., Goncalves, M. S., and Hernandez-Figueroa, H. E. (2010). Designing novel photonic devices by bio-inspired computing. *Photonics Technology Letters, IEEE*, 22(15):1177–1179.

Dani, V., Hayes, T. P., and Kakade, S. M. (2008). Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Conference on Learning Theory (COLT)*, pages 355–366.

de Freitas, N., Smola, A. J., and Zoghi, M. (2012). Exponential regret bounds for Gaussian process bandits with deterministic observations. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*. icml.cc / Omnipress.

de la Peña, V. H. (1999). A general class of exponential inequalities for martingales and ratios. *The Annals of Probability*, 27(1):537–564.

Desautels, T., Krause, A., and Burdick, J. W. (2012). Parallelizing exploration-exploitation tradeoffs with Gaussian process bandit optimization. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 1191–1198. icml.cc / Omnipress.

Ding, J., Lee, J. R., and Peres, Y. (2011). Cover times, blanket times, and majorizing measures. In *Proceedings of the forty-third annual ACM symposium on Theory of computing (STOC)*, pages 61–70. ACM.

Dudley, R. M. (1967). The sizes of compact subsets of hilbert space and continuity of gaussian processes. *Journal of Functional Analysis*, 1(3):290–330.

Dutykh, D., Poncet, R., and Dias, F. (2011). The VOLNA code for the numerical modelling of tsunami waves: generation, propagation and inundation. *European Journal of Mechanics B/Fluids*, 30:598–615.

Eiben, A. E. and Smith, J. E. (2003). *Introduction to evolutionary computing*. Springer.

Even-Dar, E., Mannor, S., and Mansour, Y. (2002). Pac bounds for multi-armed bandit and markov decision processes. In *Proceedings of the 15th Conference on Computational Learning Theory (COLT)*, pages 255–270. Springer.

Even-Dar, E., Mannor, S., and Mansour, Y. (2006). Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *The Journal of Machine Learning Research*, 7:1079–1105.

Flake, G. W. and Lawrence, S. (2002). Efficient SVM regression training with SMO. *Machine Learning*, 46(1-3):271–290.

Floudas, C. A. and Pardalos, P. M. (2000). *Optimization in Computational Chemistry and Molecular Biology: Local and Global Approaches*. Nonconvex Optimization and Its Applications. Springer.

Gaillard, P. and Gerchinovitz, S. (2015). A chaining algorithm for online nonparametric regression. *Proceedings of the 28th Conference on Learning Theory (COLT)*.

Garivier, A. and Kaufmann, E. (2016). Optimal best arm identification with fixed confidence. In *Proceedings of the 29th Conference on Learning Theory (COLT)*.

Garnier, J. and Kallel, L. (2000). Statistical distribution of the convergence time of evolutionary algorithms for long-path problems. *IEEE Transactions on evolutionary computation*, 4(1):16–30.

Garnier, J. and Kallel, L. (2001). How to detect all maxima of a function. In *Theoretical Aspects of Evolutionary Computing*, pages 343–370. Springer Berlin Heidelberg.

Gärtner, T., Flach, P., and Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. In *Learning Theory and Kernel Machines*, pages 129–143. Springer.

Giné, E. and Nickl, R. (2015). *Mathematical Foundations of Infinite-Dimensional Statistical Models*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Gittins, J. C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 148–177.

Grill, J. B., Valko, M., and Munos, R. (2015). Black-box optimization of noisy functions with unknown smoothness. In *Advances in Neural Information Processing Systems 28 (NIPS)*, pages 667–675. Curran Associates, Inc.

Grünewälder, S., Audibert, J.-Y., Opper, M., and Shawe-Taylor, J. (2010). Regret bounds for Gaussian process bandit problems. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTAT)*, pages 273–280. MIT Press.

Guestrin, C., Krause, A., and Singh, A. (2005). Near-optimal sensor placements in Gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pages 265–272. ACM.

Hanneke, S. (2011). Rates of convergence in active learning. *The Annals of Statistics*, 39(1):333–361.

Hastie, T., Tibshirani, R., and Friedman, J. (2008). *The Elements of Statistical Learning*. Springer.

Hennig, P. and Schuler, C. J. (2012). Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13:1809–1837.

Hoffman, M. D., Shahriari, B., and de Freitas, N. (2014). On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 365–374.

Jamieson, K., Malloy, M., Nowak, R., and Bubeck, S. (2013). lil'UCB: An optimal exploration algorithm for multi-armed bandits. In *Proceedings of the 27th Conference on Learning Theory (COLT)*.

Johnson, D. S. (1973). Approximation algorithms for combinatorial problems. In *Proceedings of the fifth annual ACM symposium on Theory of computing (STOC)*, pages 38–49. ACM.

Johnson, S. G. (2014). The NLopt nonlinear-optimization package. http://ab-initio.mit.edu/nlopt.

Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383.

Jones, D. R., Perttunen, C. D., and Stuckman, B. E. (1993). Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492.

Kaelo, P. and Ali, M. M. (2006). Some variants of the controlled random search algorithm for global optimization. *Journal of optimization theory and applications*, 130(2):253–264.

Kagemoto, H. and Yue, D. (1986). Interactions among multiple three-dimensional bodies in water waves: an exact algebraic method. *Journal of Fluid Mechanics*, 166(1):189–209.

Karnin, Z., Koren, T., and Somekh, O. (2013). Almost optimal exploration in multi-armed bandits. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 1238–1246.

Kaufmann, E., Cappé, O., and Garivier, A. (2012). On bayesian upper confidence bounds for bandit problems. In *International Conference on Artificial Intelligence and Statistics (AISTAT)*, pages 592–600.

Kaufmann, E., Cappé, O., and Garivier, A. (2016). On the complexity of best-arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 17(1):1–42.

Kleinberg, R. (2004). Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems 17 (NIPS)*, pages 697–704. MIT Press.

Kleinberg, R., Slivkins, A., and Upfal, E. (2008). Multi-armed bandits in metric spaces. In *Proceedings of the 40th annual ACM symposium on Theory of computing (STOC)*, pages 681–690.

Ko, C. W., Lee, J., and Queyranne, M. (1995). An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691.

Krause, A. and Guestrin, C. (2005). Near-optimal nonmyopic value of information in graphical models. In *Proceedings of the 21st Conference Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 324–331. AUAI Press.

Kushner, H. J. (1964). A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106.

Lai, T. L. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22.

Ledoux, M. and Talagrand, M. (1991). *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media.

Leslie, C. S., Eskin, E., and Noble, W. S. (2002). The spectrum kernel: A string kernel for SVM protein classification. In *Pacific symposium on biocomputing*, volume 7, pages 566–575.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2002). Text classification using string kernels. *The Journal of Machine Learning Research*, 2:419–444.

Magureanu, S., Combes, R., and Proutiere, A. (2014). Lipschitz bandits: Regret lower bounds and optimal algorithms. In *Proceedings of The 27th Conference on Learning Theory (COLT)*.

Malherbe, C., Contal, E., and Vayatis, N. (2016). A ranking approach to global optimization. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*.

Mannor, S. and Tsitsiklis, J. N. (2004). The sample complexity of exploration in the multi-armed bandit problem. *The Journal of Machine Learning Research*, 5:623–648.

McKay, M. D., Beckman, R. J., and Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61.

Mendelson, S. (2002). Geometric parameters of kernel machines. In *Proceedings of the 15th Conference on Computational Learning Theory (COLT)*, pages 29–43. Springer-Verlag.

Mladineo, R. H. (1986). An algorithm for finding the global maximum of a multimodal, multivariate function. *Mathematical Programming*, 34(2):188–200.

Močkus, J. B. (1974). *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, chapter On bayesian methods for seeking the extremum, pages 400–404. Springer Berlin Heidelberg, Berlin, Heidelberg.

Moles, C. G., Mendes, P., and Banga, J. R. (2003). Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome research*, 13(11):2467–2474.

Munos, R. (2011). Optimistic optimization of deterministic functions without the knowledge of its smoothness. In *Advances in neural information processing systems 25 (NIPS)*.

Myers, R. H. and Montgomery, D. C. (1995). *Response Surface Methodology: Process and Product in Optimization Using Designed Experiments*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition.

Nash, W. J., Sellers, T. L., Talbot, S. R., Cawthorn, A. J., and Ford, W. B. (1994). The population biology of abalone (haliotis species) in tasmania: Blacklip abalone (h. rubra) from the north coast and the island of bass strait. Technical report, Tasmania. Sea Fisheries Division.

Neuhaus, M. and Bunke, H. (2006). Edit distance-based kernel functions for structural pattern classification. *Pattern Recognition*, 39(10):1852–1863.

Papadimitriou, C. H. and Steiglitz, K. (1982). *Combinatorial optimization: algorithms and complexity*. Courier Corporation.

Perchet, V. and Rigollet, P. (2013). The multi-armed bandit problem with covariates. *The Annals of Statistics*, 41(2):693–721.

Press, W. H. (2007). *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press.

Rakhlin, A. and Sridharan, K. (2014). Online nonparametric regression. *Proceedings of the 27th Conference on Learning Theory (COLT)*, 35:1232–1264.

Rasmussen, C. E. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Raz, R. and Safra, S. (1997). A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing (STOC)*, pages 475–484. ACM.

Runarsson, T. P. and Yao, X. (2000). Stochastic ranking for constrained evolutionary optimization. *Evolutionary Computation, IEEE Transactions on*, 4(3):284–294.

Rusmevichientong, P. and Tsitsiklis, J. N. (2010). Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411.

Salomon, A. and Audibert, J.-Y. (2011). Deviations of stochastic bandit regret. In *Proceedings of the International Conference on Algorithmic Learning Theory (ALT)*, pages 159–173. Springer.

Sarkar, D., Contal, E., Vayatis, N., and Dias, F. (2015). A machine learning approach to the analysis of wave energy converters. *Proceedings of the 34th International Conference on Ocean, Offshore and Arctic Engineering (OMAE)*.

Sarkar, D., Contal, E., Vayatis, N., and Dias, F. (2016). Prediction and optimization of wave energy converter arrays using a machine learning approach. *Renewable Energy*, 97:504–517.

Sarkar, D., Renzi, E., and Dias, F. (2014). Wave farm modelling of oscillating wave surge converters. In *Proceedings of the Royal Society of London A*, volume 470, page 20140118. The Royal Society.

Schölkopf, B. and Smola, A. J. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Sebag, M. and Ducoulombier, A. (1998). Extending population-based incremental learning to continuous search spaces. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature (PPSN)*, pages 418–427. Springer Berlin Heidelberg.

Shubert, B. O. (1972). A sequential method seeking the global maximum of a function. *SIAM Journal on Numerical Analysis*, 9(3):379–388.

Simon, M. (1982). Multiple scattering in arrays of axisymmetric wave-energy devices. part 1. a matrix method using a plane-wave approximation. *Journal of Fluid Mechanics*, 120:1–25.

Slivkins, A. (2011). Multi-armed bandits on implicit metric spaces. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 1602–1610. MIT Press.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 2960–2968.

Soare, M. (2015). *Sequential Resource Allocation In Stochastic Linear Bandits*. PhD thesis, Université Lille 1.

Soare, M., Lazaric, A., and Munos, R. (2014). Best-arm identification in linear bandits. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 828–836.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. (2012). Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265.

Stefanakis, T. S., Contal, E., Vayatis, N., Dias, F., and Synolakis, C. E. (2014). Can small islands protect nearby coasts from tsunamis? An active experimental design approach. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 470(2172).

Stefanakis, T. S., Dias, F., and Dutykh, D. (2011). Local run-up amplification by resonant wave interactions. *Physical Review Letters*, 107:124502.

Stefanakis, T. S., Dias, F., Vayatis, N., and Guillas, S. (2012). Long-wave runup on a plane beach behind a conical island. In *Proceedings of the World Conference on Earthquake Engineering*.

Stewart, G. W. (1998). *Matrix Algorithms: Volume 1, Basic Decompositions*. Society for Industrial and Applied Mathematics.

Talagrand, M. (2014). *Upper and Lower Bounds for Stochastic Processes: Modern Methods and Classical Problems*, volume 60. Springer-Verlag Berlin Heidelberg.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294.

Valko, M., Carpentier, A., and Munos, R. (2013). Stochastic simultaneous optimistic optimization. In *Proceedings of the 30st International Conference on Machine Learning (ICML)*.

van der Vaart, A. W. and van Zanten, J. H. (2008). Reproducing kernel Hilbert spaces of Gaussian priors. In *Pushing the limits of contemporary statistics: contributions in honor of Jayanta K. Ghosh*, pages 200–222. Institute of Mathematical Statistics.

van der Vaart, A. W. and van Zanten, J. H. (2009). Adaptive Bayesian estimation using a Gaussian random field with inverse gamma bandwidth. *The Annals of Statistics*, pages 2655–2675.

van der Vaart, A. W. and van Zanten, J. H. (2011). Information rates of nonparametric Gaussian process methods. *The Journal of Machine Learning Research*, 12:2095–2119.

Vazquez, E. and Bect, J. (2010). Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and Inference*, 140(11):3088–3095.

Vishwanathan, S. V. N., Schraudolph, N. N., Kondor, R., and Borgwardt, K. M. (2010). Graph kernels. *The Journal of Machine Learning Research*, 11:1201–1242.

Wang, G. and Shan, S. (2007). Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, 129(4):370–380.

Wang, Z., Shakibi, B., Jin, L., and de Freitas, N. (2014). Bayesian multi-scale optimistic optimization. In *Artificial Intelligence and Statistics (AISTATS)*, pages 1005–1014.

Zabinsky, Z. B. and Smith, R. L. (1992). Pure adaptive search in global optimization. *Mathematical Programming*, 53(1-3):323–338.

Ziemba, W. T. and Vickson, R. G. (2006). *Stochastic optimization models in finance*. World Scientific Singapore.

université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

Titre : Méthodes d'apprentissage statistique pour l'optimisation globale

Mots clefs : optimisation globale, bandits stochastiques, optimisation bayésienne.

Résumé : Cette thèse se consacre à une analyse rigoureuse des algorithmes d'optimisation globale séquentielle. On se place dans un modèle de bandits stochastiques où un agent vise à déterminer l'entrée d'un système optimisant un critère. Cette fonction cible n'est pas connue et l'agent effectue séquentiellement des requêtes pour évaluer sa valeur aux entrées qu'il choisit. Cette fonction peut ne pas être convexe et contenir un grand nombre d'optima locaux. Nous abordons le cas difficile où les évaluations sont coûteuses, ce qui exige de concevoir une sélection rigoureuse des requêtes. Nous considérons deux objectifs, d'une part l'optimisation de la somme des valeurs reçues à chaque itération, d'autre part l'optimisation de la meilleure valeur trouvée jusqu'à présent. Cette thèse s'inscrit dans le cadre de l'optimisation bayésienne lorsque la fonction est une réalisation d'un processus stochastique connu, et introduit également une nouvelle approche d'optimisation par ordonnancement où l'on effectue seulement des comparaisons des valeurs de la fonction. Nous proposons des algorithmes nouveaux et apportons des concepts théoriques pour obtenir des garanties de performance. Nous donnons une stratégie d'optimisation qui s'adapte à des observations reçues par batch et non individuellement. Une étude générique des supremums locaux de processus stochastiques nous permet d'analyser l'optimisation bayésienne sur des espaces de recherche nonparamétriques. Nous montrons également que notre approche s'étend à des processus naturels non gaussiens. Nous établissons des liens entre l'apprentissage actif et l'apprentissage statistique d'ordonnancements et déduisons un algorithme d'optimisation de fonctions potentiellement discontinue.

Title : Statistical Learning Approaches for Global Optimization

Keywords : global optimization, stochastic bandits, Bayesian optimization.

Abstract : This dissertation is dedicated to a rigorous analysis of sequential global optimization algorithms. We consider the stochastic bandit model where an agent aim at finding the input of a given system optimizing the output. The function which links the input to the output is not explicit, the agent requests sequentially an oracle to evaluate the output for any input. This function is not supposed to be convex and may display many local optima. In this work we tackle the challenging case where the evaluations are expensive, which requires to design a careful selection of the input to evaluate. We study two different goals, either to maximize the sum of the rewards received at each iteration, or to maximize the best reward found so far. The present thesis comprises the field of global optimization where the function is a realization from a known stochastic process, and the novel field of optimization by ranking where we only perform function value comparisons. We propose novel algorithms and provide theoretical concepts leading to performance guarantees. We first introduce an optimization strategy for observations received by batch instead of individually. A generic study of local supremum of stochastic processes allows to analyze Bayesian optimization on nonparametric search spaces. In addition, we show that our approach extends to natural non-Gaussian processes. We build connections between active learning and ranking and deduce an optimization algorithm of potentially discontinuous functions.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France